Machine Learning Models for Mushroom Classification: Edibility

Steven Ray Alvarado
College of Letters and Science
University of California, Davis
Sacramento, United States
stevenrayalvarado@gmail.com

Abstract—**The consumption of wild mushrooms poses inherent risks due to the potential presence of toxic species. Thus, accurate classification of mushroom edibility is crucial for ensuring the safety and efficiency of mushroom foraging. This wert presents a comprehensive investigation into the efficacy of various machine learning models for classifying poisonous mushrooms based on physical attributes.**

With a dataset comprising morphological features of numerous mushroom species, we employ a range of machine learning algorithms including neural networks, decision trees, and random forests to classify mushrooms as poisonous or edible. Performance metrics such as accuracy, precision, recall, and F1 score are recorded to evaluate the models' classification abilities. Furthermore, feature importance analysis is conducted to discern the most influential attributes in the classification process. This aids in understanding the underlying characteristics that distinguish edible from poisonous mushrooms, contributing to both model interpretability and domain knowledge.

I. Introduction

A. Background

Valued for their nutritional and therapeutic benefits, mushrooms have long been a part of the human diet and possess a rich and distinct history as one of the world's staple food sources. One of the key benefits of mushroom foraging is the high nutritional value and unique flavors that these fungi can provide. However, not all mushrooms are deemed safe for human consumption, with certain species capable of causing severe illness or death at lethal doses. Many wild mushrooms have the potential to be poisonous, making proper identification crucial to avoiding potential harm. In the wild, assessing the edibility of a mushroom with the naked eye alone can be a time-consuming and error-prone process. With no simple tests or rules to remember, it takes extensive knowledge and experience to confidently identify species of mushrooms that are safe for ingestion.

B. Problem Statement

Therefore, our problem lies in finding a fast and reliable method for classifying a mushroom's edibility. Machine learning models can be trained to classify mushrooms based on their physical attributes, such as their odor, cap shape, ring type, gill size, etc. Our goal is to train and evaluate several of these classification models for their ability to accurately differentiate between poisonous and edible mushrooms. With this, we hope to optimize the safety and speed of mushroom foraging, where the best models can greatly reduce the risk of error and ultimately automate the identification process.

II. Literature Review

To build a better understanding of how to approach this problem, background research was conducted to review any relevant and potential methods that already exist. Multiple published papers concerning the topic of mushroom identification were reviewed. Firstly, we researched the different methods used for mushroom classification. This included macroscopic, microscopic, and molecular identification techniques, as well as spectral and chromatographic methods [1]. Although each method has its advantages and disadvantages, we learned that macroscopic identification is the type of observation that our mushroom dataset is typically concerned with, where the physical attributes of the mushrooms are strictly examined. Our dataset consists of these naked-eye observations as our variables, such as cap shape, cap color, gill size, veil type, stalk shape, etc.

Some peer-reviewed papers also inspired a plethora of different classification algorithms we can include in our experiments. Many of the studies we researched

compared multiple algorithms and/or models to see which ones were the most effective. In "Classification algorithm for edible mushroom identification," three different models are compared: Decision Tree, Naive Bayes, and Support Vector Machine [2]. This study also used the same data set that we will be using. In another study that we looked into, pre-trained (CNN) models were used, which included VGG16, ResNet50, and Inception V3 [3]. This study differs from ours in that it used images instead of macroscopic observations for the dataset. Based on these related papers, we followed the trend of comparing multiple models and gathering insight on which ones perform optimally. Ultimately, these studies inspired us to explore and compare a large range of models with a greater understanding of their implications in performance.

III. DATASET DESCRIPTION

For this project, we use the "Mushroom" dataset from the UCI Machine Learning wesitory, containing 8,124 instances and 22 categorical features of gilled mushrooms in the Agaricus and Lepiota families. A corresponding dataset can be found on Kaggle with a formatted CSV file of the mushroom samples. Both of these resources were utilized to help form a complete understanding of our model implementation. Imported from the UCI we, included with the dataset is a file named "agaricus-lepiota.names", which is a brief collection of notes provided by the dataset donor. Some of these comments helped us format the data during the exploratory data analysis phase, particularly for missing values found in the stalk root feature.

A. Characteristics

1) Target Features: In Table I, we see that the classification type is binary, where a mushroom is determined to be either poisonous or edible. Edible mushrooms are fungi that humans can safely eat as food. Poisonous mushrooms, on the other hand, are fungi that can be toxic and harmful if ingested, potentially leading to illness or death.

TABLE I
TARGET

Feature	Values
Class	edible, poisonous

2) Cap Features: Moreover, Table II shows the various cap features of mushrooms, where the cap is considered the mushroom's uppermost body structure and often looks like an umbrella. The cap shape is how this upper portion is formed and the cap surface is the type of structure found at the top of the mushroom.

TABLE II
CAP

Feature	Values
Shape	bell, conical, convex, flat, knobbed, sunken
Surface	fibrous, grooves, scaly, smooth
Color	brown, buff, cinnamon, gray, green, pink, purple, red, white, yellow

3) Gill Features: Table III depicts the different gill features in mushrooms, a major characteristic that is located underneath the mushroom cap. The gill attachment is the way the gills grow on the mushroom's stem, the gill spacing is the arrangement of the gills and their distribution, and the gill size is the width of the gills—where "broad" indicates thickness and "narrow" indicates thinness.

TABLE III
GILL

Feature	Values
Attachment	attached, descending, free, notched
Spacing	close, crowded, distant
Size	broad, narrow
Color	black, brown, buff, chocolate, gray, green, orange, pink, purple, red, white, yellow

4) Stalk Features: Table IV illustrates a mushroom's possible stalk features. The stalk shape is the shape of the mushroom's stem that holds the cap above the ground. Some mushrooms possess stalks with a thicker base ("enlarged") and others possess stalks that start to thin from cap to base ("tapering"). Additionally, the stalk root is the lower part of the stem that attaches to or makes contact with a substrate. The stalk surface is the structure above or below the mushroom's ring, and the stalk color is the color of the stalk above or below the mushroom's ring.
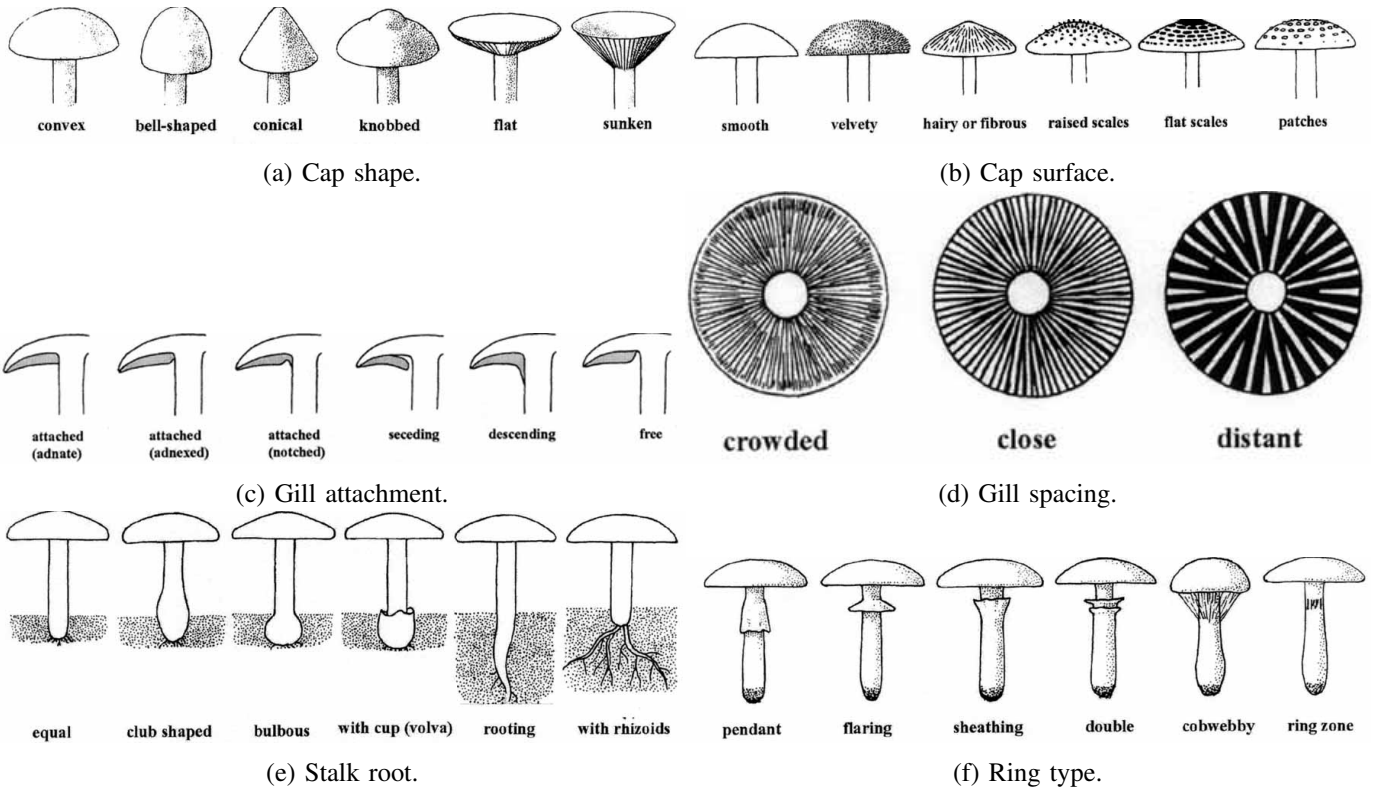
(a) Cap shape.

(b) Cap surface.

(c) Gill attachment.

(d) Gill spacing.

(e) Stalk root.

(f) Ring type.

Fig. 1. Some mushroom features included in the dataset. [2]

TABLE IV
STALK

Feature	Values
Shape	enlarging, tapering
Root	bulbous, club, cup, equal, rhizomorphs, rooted, missing
Surface above/below ring	fibrous, scaly, silky, smooth
Color above/below ring	brown, buff, cinnamon, gray, orange, pink, red, white, yellow

TABLE V
VEIL

Feature	Values
Type	partial, universal
Color	brown, orange, white, yellow

TABLE VI
RING

Feature	Values
Number	none, one, two
Type	cobwebby, evanescent, flaring, large, none, pendant, sheathing, zone

5) Veil Features: Furthermore, Table V portrays the veil features of the mushroom, where the veil type is the kind of veil membrane the mushroom has, and the veil color is simply the color of the mushroom's veil.

6) Ring Features: Table VI outlines the potential ring features a mushroom can have, where the rings are tissue-like structures that surround the stalk of certain mushrooms. The ring number is the number of rings a mushroom has on the stalk, and the ring type refers to the type of structure that the ring possesses.

7) Miscellaneous Features: Finally, Table VII describes numerous characteristics that do not fall under a grouping of related features. We refer to these as miscellaneous features. The odor of a mushroom is the aroma that it emits, ranging from foul or pungent to earthy and pleasant. The spore print color is the color of the spore deposit left by a mushroom when it discharges mature spores. The population feature describes the distribution

of mushrooms within a particular area, where "several" means that there are multiple individual species, "solitary" means that are only one species, "scattered" means that mushrooms are scattered throughout the area, and "clustered" means that mushrooms in the area grow in groups or colonies.

TABLE VII
MISCELLANEOUS

Feature	Value
Bruises	yes, no
Odor	almond, anise, creosote, fishy, foul, musty, none, pungent, spicy
Spore print color	black, brown, buff, chocolate, green, orange, purple, white, yellow
Population	abundant, clustered, numerous, scattered, several, solitary
Habitat	grasses, leaves, meadows, paths, urban, waste, woods

IV. EXPLORATORY DATA ANALYSIS

A. Observations

For dataset analysis and understanding, we imported our data as a Pandas dataframe in a collaborative Jupyter notebook and dissected it with several functions to help illustrate any significant details. The original data values were all in character format, such as the categories for the "class" feature, where each entry contained either "p" for poisonous or "e" for edible. This feature was determined to be the classifier, where the target variable is "poisonous". Regardless, it was difficult to interpret the dataset with character values alone, so we reformatted them by mapping each character to its full string name. This helped us view the dataset from a more verbose and clarified perspective.

Overall, the dataset consists of 8,124 rows and 23 columns. As shown in Fig. 2, the target class has a roughly balanced distribution, where 51.8% of the mushrooms are edible and 48.2% are poisonous. It is important to note that every column is categorical (i.e., in character format), where techniques such as label encoding and one-hot encoding must be applied to transform the data into a usable, numerical format. From this, we gathered statistics such as percentage breakdowns of each column to influence what variables we should keep or remove.

B. Feature Selection

In Fig. 3a, we discovered that the mushroom dataset has a large majority of samples that have free gill attachments (97.42%). The veil type feature in Fig. 3f was the largest indicator of this, where 100% of the samples were all one type ("partial"). This trend can also be observed in Fig. 3h, where 92.17% of mushrooms have one ring. The veil color, shown in Fig. 3g, also had a similar case, where 97.54% of the mushrooms have white veil colors and 2.46% make up the others. Ultimately, the skewed nature of these distributions signifies variables that do not provide enough unique information that can effectively train our model.

As depicted in Fig. 3b, we also observed that 56.72% of the mushrooms have tapering stalk shapes while 43.28% have enlarging stalk shapes. With this relatively even distribution, we can also see that there is an insignificant difference between the poisonous and edible mushrooms for each value. There were also missing data values that we had to consider, as seen in Fig. 3c. 30.53% of the mushrooms had an unknown stalk root, which indicated inconclusive data that we decided to ignore during model training. Additionally, for Fig. 3d and Fig. 3e, we noticed that the stalk surface and the stalk color below the ring had closely similar distributions to their above-the-ring counterparts. As a result, such features were eliminated from the dataset to ensure that our model utilizes the most useful information.

V. PROPOSED METHODOLOGY

TABLE VIII
CLASSIFIERS

Model
Logistic Regression
Perceptron Learning
Gaussian Naive Bayes
Multinomial Naive Bayes
Categorical Naive Bayes
Dummy Classifier
Decision Tree
Random Forest

A. Classification Models

Given the variety of different models that can be applied to this dataset, we decided to train with the classifiers listed in Table VIII and compile performance
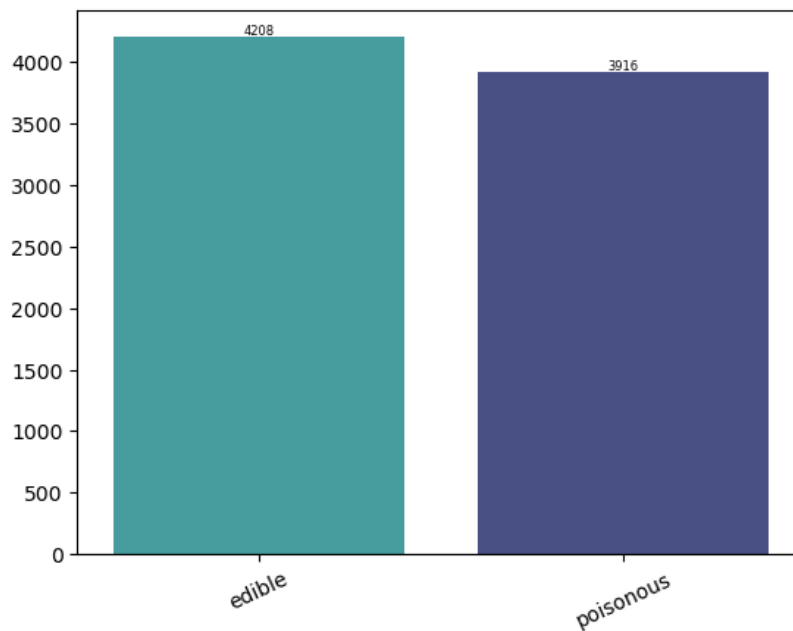
Fig. 2. Count plot of edible and poisonous mushrooms.

metrics for each model to help evaluate their robustness. These models were discovered through background research and imported through the `scikit-learn` library, which also contained several other classifiers that we could have included in our experiments. However, to stay within the scope of what was taught in class, we limited training on select models that were covered in the lectures.

1) Logistic Regression: Logistic Regression is a linear model commonly used for binary classification problems. It estimates the probability that a given input belongs to a particular category. This fits within our dataset because we are faced with the binary task of determining edibility, where 0 is encoded as edible and 1 is encoded as poisonous.

2) Perceptron Learning: Perceptron Learning is a type of linear classifier and is one of the simplest artificial neural networks. It is suitable for binary classification problems, particularly for linearly separable data.

3) Naive Bayes: Naive Bayes classifiers apply Bayes' theorem with strong (naive) independence assumptions between features. There are several types depending on the distribution of the data.

a) Gaussian: Gaussian Naive Bayes is a variant of the Naive Bayes classifier that assumes the features follow a Gaussian (normal) distribution. It's fast, simple, and performs well with smaller datasets.

b) Multinomial: Multinomial Naive Bayes is used for classification with discrete features and is effective for high-dimensional data.

c) Categorical: Categorical Naive Bayes is a variant that is highly suitable for problems where features are categorical rather than continuous. This is the type of classifier that fits most with the nature of our dataset. However, like every Naive Bayes variant, it assumes that features are independent, which may not be realistic for all datasets.

Although the Gaussian and Multinomial variants do not particularly follow the distribution of our dataset, it is important to include them in our model evaluations for comparison purposes.

4) Dummy Classifier: The Dummy Classifier is a simple baseline classifier that makes predictions based on simple rules, such as predicting the most frequent class or making random predictions. Since it does not provide meaningful predictions, it is only used to compare the performance of the other classifiers.

5) Decision Tree: The Decision Tree Classifier uses a tree-like model of decisions and their possible consequences. It splits the data into subsets based on the value of input features and can handle both numerical and categorical data.
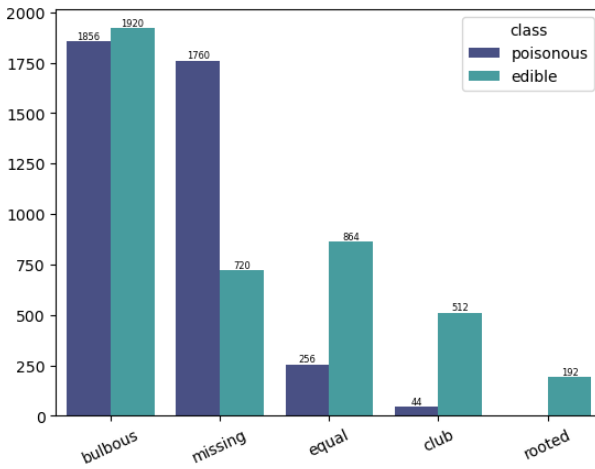
6) Random Forest: The Random Forest Classifier is an ensemble learning method that constructs multiple decision trees during training. By averaging multiple
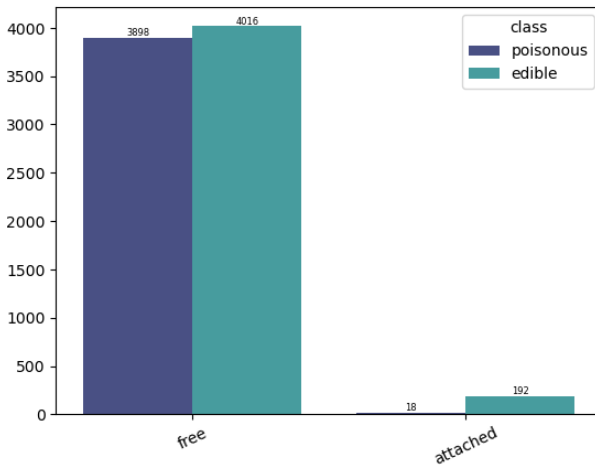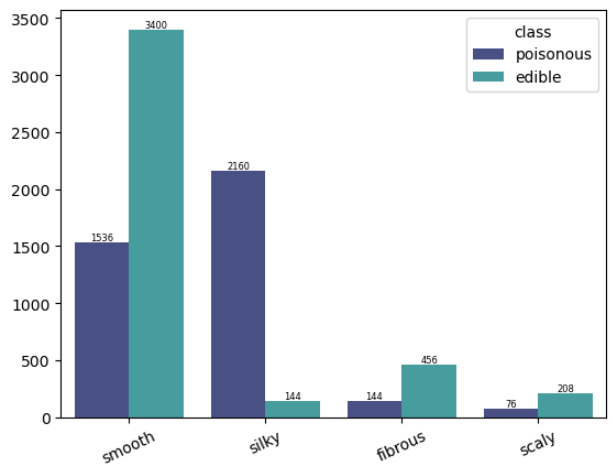
(a) Gill attachment.
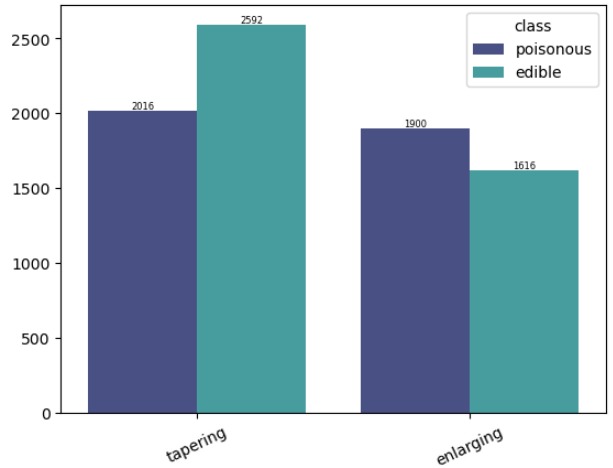
(b) Stalk shape.

(c) Stalk root.

(d) Stalk surface below ring.

trees, it reduces overfitting and improves accuracy. It is suitable for a variety of classification problems and handles missing values well.

B. Preprocessing

To prepare the model for training, data preprocessing was conducted to allow the machine learning algorithms to effectively utilize the input information. The data was split into their corresponding train and test sets, with X_train and X_test for the dependent variables and y_train and y_test for the target class. Label encoding and one-hot encoding were used for the y and X variables, respectively. With only two distinct values between the classes, the use of label encoding for the target feature is suitable for defining an intrinsic order within a few categories, where 0 stands for edible and 1 stands for poisonous. In this case, more meaning is assigned to the "poisonous" class, which corresponds

with our model's purpose in determining whether or not a mushroom is dangerous to eat. Moreover, since we are dealing with categorical data that is nominal (lacks inherent order) for the dependent variables, one-hot encoding transforms this information into a numerical format that can be used in model training.

C. Metrics

With that, several performance metrics are measured for each model that grant us insight into how well the classifier performs on our dataset. To do this, a variety of scores for each model are stored in a metrics dataframe for us to review. Many of these scores were calculated with the help of the `scikit-learn` module.
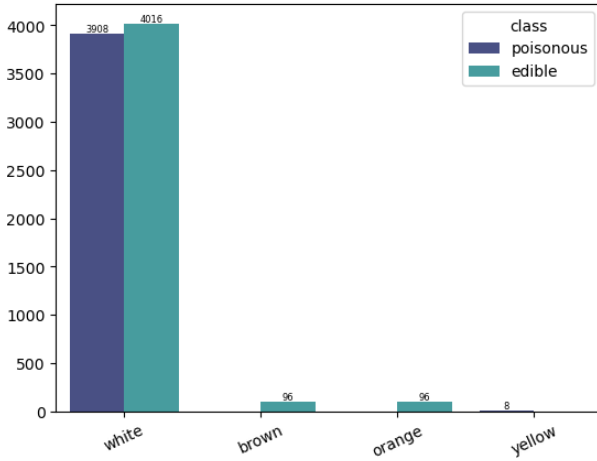
1) Accuracy: Classification accuracy is a commonly used metric that measures the proportion of correct predictions made by a model out of the total predictions made. It is defined as the ratio of the number of
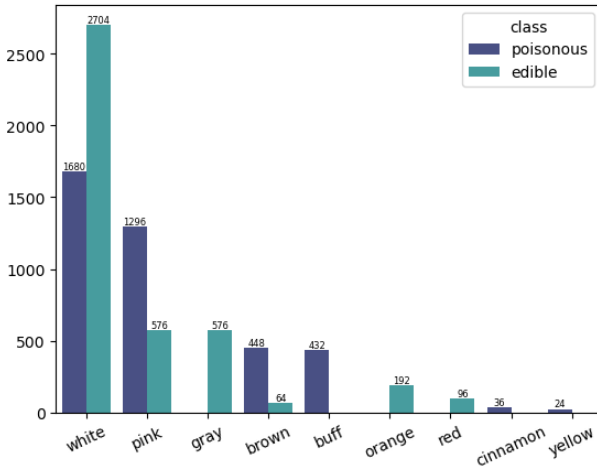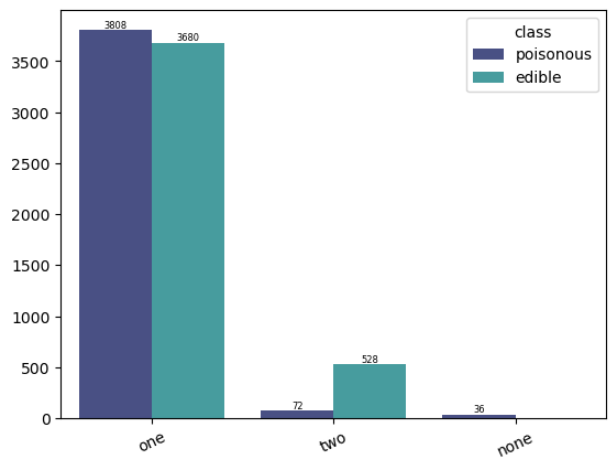
(e) Stalk color below ring.
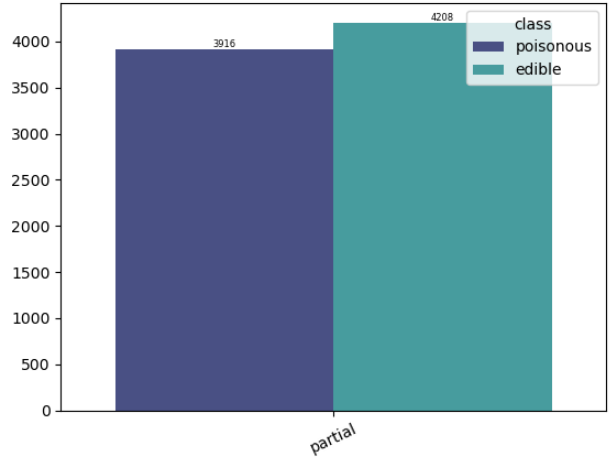
(f) Veil type.

(g) Veil color.

(h) Ring number.

Fig. 3. Count plots of insignificant features that were dropped from training.

correctly classified instances (both true positives and true negatives) to the total number of instances. In our source code, we iterate through ground truth values and compare them with generated predictions to produce an accuracy score.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

Accuracy can be misleading in cases of imbalanced datasets, where the number of instances in one class significantly outweighs the number of instances in another class. In our case, the number of instances in each of our target classes has a nearly even split. Overall, accuracy is a basic yet important metric that provides a direct measure of a machine learning model's performance by indicating how often the model's predictions are correct.

2) Precision: Precision is defined as the ratio of true positive predictions (correctly predicted positive cases) to the total number of positive predictions made by the model (both true positives and false positives). In other words, it is a measure of the accuracy of the positive predictions that are made by a model.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

A high precision means that when the model predicts a positive class, it is very likely to be correct. Precision is particularly useful in scenarios where the cost of false positives is high. For instance, in the case of our classification problem, it can be a matter of life or death if a poisonous mushroom is accidentally ingested.

3) Recall: Also known as sensitivity or true positive rate, recall is a measure of a model's ability to correctly

	Model	Accuracy (train)	Accuracy (test)
0	LogisticRegression	1.000000	1.000000
1	Perceptron	1.000000	1.000000
5	RandomForestClassifier	1.000000	1.000000
7	DecisionTreeClassifier	1.000000	1.000000
2	GaussianNB	0.990153	0.984737
3	MultinomialNB	0.972591	0.972920
4	CategoricalNB	0.960939	0.958641
6	DummyClassifier	0.519941	0.512063

(a) Accuracy.

	Model	Precision (train)	Precision (test)
0	LogisticRegression	1.000000	1.000000
1	Perceptron	1.000000	1.000000
5	RandomForestClassifier	1.000000	1.000000
6	DummyClassifier	1.000000	1.000000
7	DecisionTreeClassifier	1.000000	1.000000
3	MultinomialNB	0.998915	0.996815
4	CategoricalNB	0.992305	0.993471
2	GaussianNB	0.980866	0.969667

(b) Precision.

	Model	Recall (train)	Recall (test)
0	LogisticRegression	1.000000	1.000000
1	Perceptron	1.000000	1.000000
2	GaussianNB	0.998974	1.000000
5	RandomForestClassifier	1.000000	1.000000
7	DecisionTreeClassifier	1.000000	1.000000
3	MultinomialNB	0.943932	0.947528
4	CategoricalNB	0.925812	0.921292
6	DummyClassifier	0.000000	0.000000

(c) Recall.

	Model	F1 score (train)	F1 score (test)
0	LogisticRegression	1.000000	1.000000
1	Perceptron	1.000000	1.000000
5	RandomForestClassifier	1.000000	1.000000
7	DecisionTreeClassifier	1.000000	1.000000
2	GaussianNB	0.989837	0.984600
3	MultinomialNB	0.970645	0.971547
4	CategoricalNB	0.957906	0.956021
6	DummyClassifier	0.000000	0.000000

(d) F1 score.

Fig. 4. Tables of performance metrics sorted in ascending order according to the test measure.

identify all relevant instances within a dataset. Specifically, it quantifies the proportion of actual positive cases that are correctly identified by the model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

A higher value indicates that the model is more effective at identifying positive instances. Recall is particularly important in situations where missing positive cases is more critical than incorrectly identifying negatives.

4) F1 score: The F1 score is a widely known metric used in machine learning for evaluating the performance of classification models, especially in binary classification tasks. The F1 score is the harmonic mean of precision and recall and balances both metrics, making it useful for evaluating the overall performance of a classification model.

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

VI. MODEL EVALUATION

A. Experimental Results

Evaluating machine learning models is a critical step in ensuring their effectiveness, reliability, and validity. According to Fig. 4, several of our models were reported to have 100% accuracy, precision, recall, and F1 scores by our metrics dataframe. While this may initially indicate a perfect performance, it raises the question of whether there was potential overfitting with the data. To further explore this, we took measures from both training and testing sets and included them in our tables. A great difference in performance between the training and testing sets possibly indicates an overfit. However, since the two score columns tend to closely or completely match each other, this does not seem to be the case.

The top 5 models that performed well include Logistic Regression, Perceptron, Gaussian Naive Bayes, Decision Tree, and Random Forest. Since Logistic Regression primarily deals with binary classification tasks, it turned out to be a great model for our mushroom problem. Perceptron Learning was also suitable with its linear approach. The Decision Tree and Random Forest classifiers exhibited top scores as well. Although Decision Trees can be more susceptible to overfitting, Random Forest acts as an ensemble of decision trees that merges them to produce more accurate and stable predictions.

On the other hand, Gaussian Naive Bayes outperformed both the Multinomial and Categorical variants on every level. Theoretically, Categorical Naive Bayes
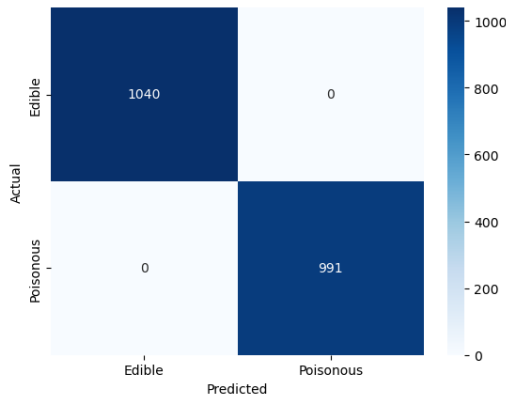
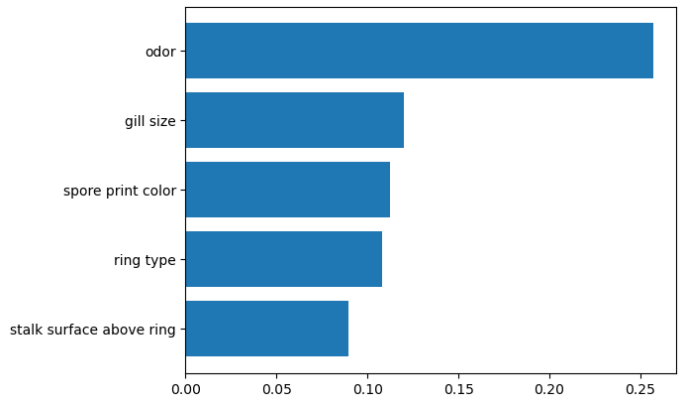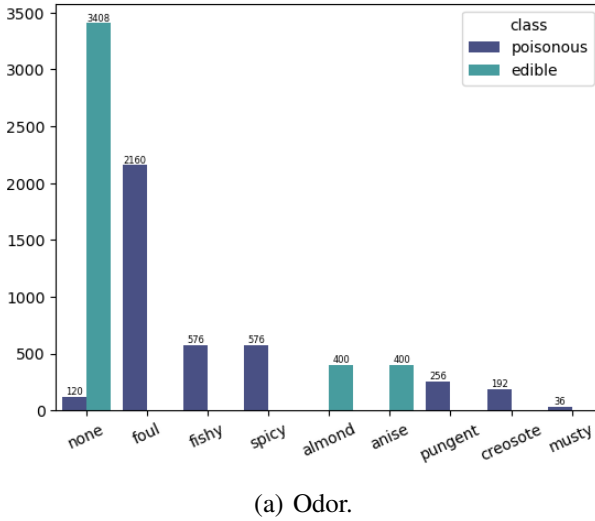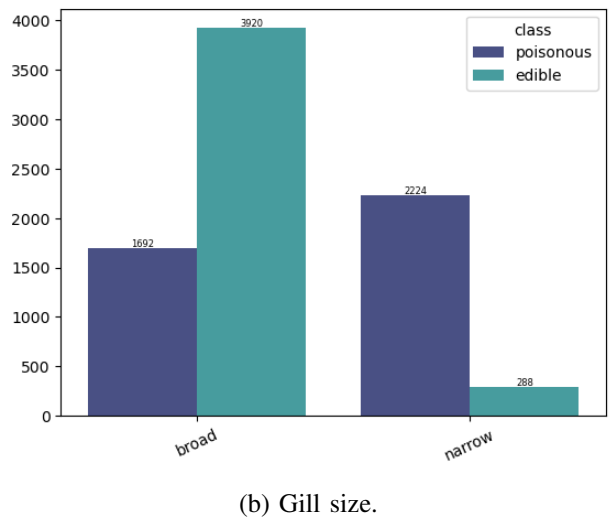Fig. 5. Confusion matrix for Random Forest.

Fig. 6. Random Forest's most important features.

(a) Odor.

(b) Gill size.

should have performed better due to the nature of our dataset. However, since these features were transformed through encoding, we ended up benefiting the performance of Gaussian Naive Bayes while hindering the performance of the Categorical variant. Even after explicitly converting the feature data types during data formatting, it seems that the Categorical variant was not insensitive to feature encoding.

As for the overall performance of the Naive Bayes models (with scores below 100%), it is important to remember that these algorithms operate with the assumption of independent features, which is not strictly the case for our variables. For instance, mushrooms with certain cap colors might be more prone to bruising, or bruising may change the apparent color of the cap. The type of habitat where a mushroom grows (woods, grass, etc.) can also be related to the population size. For example, mushrooms that grow in woods might be found in smaller populations compared to those growing

in open fields.

Finally, the Dummy Classifier serves as a reference point to understand if a more sophisticated model is actually learning meaningful patterns from the data, or if it is just performing at the level of a naive strategy. As we can see from the accuracy table in Fig. 4a, the Dummy Classifier correctly classifies a poisonous mushroom approximately 50% of the time.

B. Confusion Matrix

To help identify any patterns of misclassification, a confusion matrix was used to provide a visual summary of the predictions compared to the ground truths. With a 100% accuracy rate, the confusion matrix for the top models (Linear Regression, Perceptron, Decision Tree, and Random Forest) is shown in Fig. 5. For further exploration into our experimental results and their implications as a whole, we decided to focus on the Random Forest model.
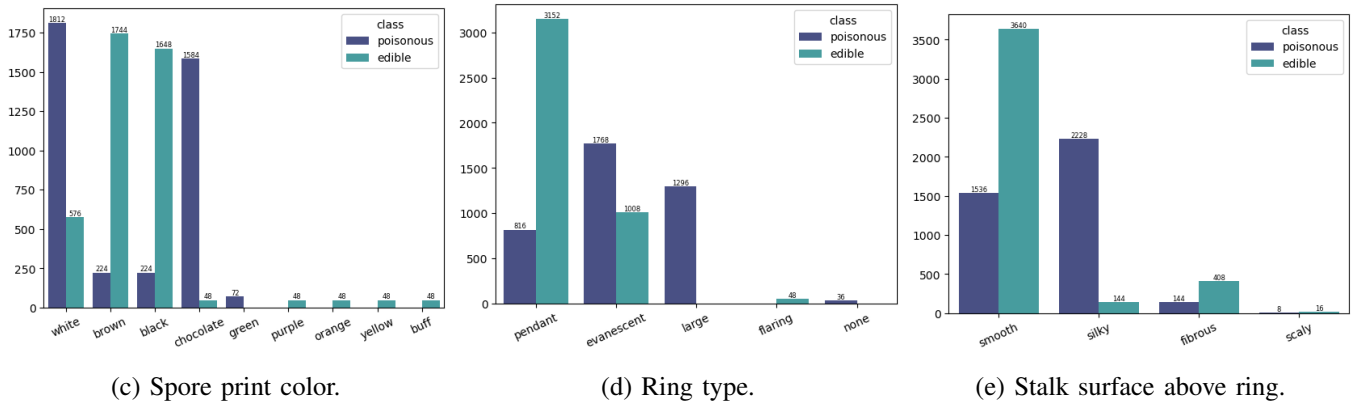
(c) Spore print color. (d) Ring type. (e) Stalk surface above ring.

Fig. 7. Count plots of the top features in descending order.

C. Feature Importance

Included in the `scikit-learn` library's Random Forest Classifier is gini importance, a measure used to evaluate the significance of features in decision tree models. Also known as Mean Decrease in Impurity (MDI), it helps in understanding which features contribute the most to the prediction accuracy of the model. With Fig. 6, we can see the Random Forest model's report of the top 5 most important features: odor, gill size, spore print color, ring type, and stalk surface above the ring.

VII. DISCUSSION AND CONCLUSION

Based on the insights gained from the model training and evaluation, certain features stand out as strong indicators of whether a mushroom is edible or poisonous. The most important feature determined by the Random Forest model is odor. According to Fig. 7a, mushrooms without any discernible odor are more likely to be edible, while those with a foul smell are more likely to be poisonous. The width of the gills also plays a significant role. In Fig. 7b, Mushrooms with narrow gills tend to be more poisonous, while those with broad gills are predominantly edible. However, it's worth noting that even broad-gilled mushrooms may carry some risk of toxicity. For Fig. 7c, mushrooms leaving white or chocolate-colored spore prints are more likely to be poisonous, whereas those leaving brown or black prints are generally safer to consume. Show in Fig. 7d, mushrooms with a pendant ring are more likely to be edible, while those with an evanescent (disappearing) or large ring are more likely to be poisonous. Finally, in Fig. 7e, mushrooms with a silky stalk surface above the ring are typically poisonous, whereas those with a smooth surface are more likely to be edible.

In this project, we explored the application of machine learning models for classifying the edibility of wild mushrooms. Among the classifiers tested, decision trees and random forests emerged as the most effective models due to their high accuracy, robustness, and interpretability. Overall, this study demonstrates that machine learning can be a powerful tool in mycology for identifying potentially dangerous mushrooms, highlighting the importance of leveraging computational methods for enhancing food safety practices. The findings of this report pave the way for developing reliable, automated systems for mushroom edibility classification, contributing to both scientific research and practical applications in the field. Future work could involve expanding the dataset, exploring additional features, and integrating advanced techniques like deep learning to further enhance classification performance.

References

[1] Z. Yan, H. Liu, J. Li, and Y. Wang, "Application of identification and evaluation techniques for edible mushrooms: A review," *Critical Reviews in Analytical Chemistry*, vol. 53, no. 3, pp. 634–654, 2021.

[2] G. Lincoff and National Audubon Society, *The National Audubon Society Field Guide to Mushrooms*. New York: Knopf: Distributed by Random House, 1981.

[3] A. Wibowo, Y. Rahayu, A. Riyanto, and T. Hidayatulloh, "Classification algorithm for edible mushroom identification," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018, pp. 250–253.

[4] N. Zahan, M. Z. Hasan, M. A. Malek, and S. S. Reya, "A deep learning-based approach for edible, inedible and poisonous mushroom classification," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, 2021, pp. 440–444.