Predicting Mushroom Edibility Using Supervised Learning Algorithms

Ma. Mae Sid Santos
College of Computing and Information Technologies
National University
Manila, Philippines
santosmm5@students.national-u.edu.ph

Gabriel Angelo Viñas
College of Computing and Information Technologies
National University
Manila, Philippines
vinasgm@students.national-u.edu.ph

***Abstract*—The accurate identification of wild mushrooms is a critical task for foragers, as misclassification can lead to severe poisoning or death. This paper addresses this challenge by framing mushroom identification as a binary classification problem. We leverage the well-known UCI Mushroom dataset, which contains morphological descriptions of 23 mushroom species, to develop and evaluate predictive models. Four supervised learning algorithms are employed: Logistic Regression (LR), K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM). Our methodology includes a rigorous experimental design to assess the impact of handling missing data and the trade-off between model complexity and performance through feature selection. The results demonstrate exceptional predictive power across all models, with both Random Forest and Support Vector Machine classifiers achieving 100% accuracy, precision, recall, and F1-score on the held-out test set. Feature importance analysis reveals that the 'odor' attribute is the most significant predictor of edibility. These findings suggest that morphological characteristics, when analyzed with appropriate machine learning techniques, can serve as highly reliable indicators of mushroom edibility, providing a strong foundation for the development of digital decision-support tools for foragers.**

***Index Terms*—Mushroom Classification, Supervised Learning, Random Forest, Support Vector Machine, Feature Importance, Predictive Modeling.**

I. Introduction

Mushroom foraging is a popular recreational activity that connects enthusiasts with nature and provides a source of wild food. However, this practice carries significant risks. Many poisonous mushroom species closely resemble edible ones, and misidentification can lead to severe gastrointestinal distress, organ failure, or even death [1]. Mycological field guides, such as *The Audubon Society Field Guide to North American Mushrooms*, provide expert knowledge but explicitly caution that there are no simple, universal rules for determining a mushroom's edibility—unlike mnemonics for plants like Poison Ivy [1]–[3]. This inherent complexity and the high stakes of an incorrect identification create a compelling need for robust, data-driven decision-support systems [8].

This study addresses the challenge of mushroom identification by formulating it as a binary classification problem. The primary objective is to develop and evaluate supervised machine learning models that can accurately predict whether a mushroom is 'edible' or 'poisonous' based on a set of 22 observable, physical characteristics. A critical requirement for any such model is the minimization of false negatives—that is, incorrectly classifying a poisonous mushroom as edible—as this type of error has the most severe real-world consequences [1], [9].

The contribution of this paper is a systematic investigation into the efficacy of machine learning for this task. We conduct a comparative analysis of four distinct and widely used classification algorithms: Logistic Regression (LR), K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM) [10]. Furthermore, we perform two controlled experiments to address key data science questions. The first experiment evaluates different strategies for handling a significant proportion of missing values in the 'stalk-root' feature [11]. The second experiment investigates the trade-off between model complexity and predictive accuracy by systematically reducing the number of features based on their importance [12].

This paper is structured as follows: Section II provides the theoretical background for the dataset and the machine learning algorithms employed. Section III details the dataset characteristics, preprocessing steps, and evaluation metrics. Section IV outlines the design of our two primary experiments. Section V presents and discusses the results of these experiments and the final model performance. Finally, Section VI concludes with a summary of our findings, a discussion of the study's limitations, and directions for future work.

II. Background and Related Work

This section provides the theoretical and historical context for the dataset and the supervised learning models used in this study.

A. The UCI Mushroom Dataset

The dataset used in this research is the Mushroom Data Set, a well-established benchmark for classification algorithms, originally curated from *The Audubon Society Field Guide to North American Mushrooms* (1981) and donated to the University of California, Irvine (UCI) Machine Learning Repository [1]. It consists of 8,124 instances of hypothetical mushroom samples corresponding to 23 species within the Agaricus and Lepiota families [1]. Each instance is described by 22 categorical attributes, such as cap shape, odor, and gill color, and is assigned a binary class label: 'edible' or 'poisonous'. The dataset is noted for being well-balanced, with approximately 52% edible and 48% poisonous samples [1]. Previous work on this dataset has consistently shown high classification accuracy, setting a high bar for new models [13].

B. Logistic Regression (LR)

Logistic Regression is a statistical model used for binary classification. Despite its name, it is a classification algorithm, not a regression one. It models the probability of a binary outcome by applying a logistic (or sigmoid) function to a linear combination of the input features [2]. The origins of logistic regression can be traced to the work of D. R. Cox in 1958 on the analysis of binary sequences, which laid the mathematical groundwork for modeling dichotomous dependent variables [2]. It has since become a foundational method in many fields, particularly biostatistics and epidemiology, due to its interpretability and computational efficiency [7], [14].

C. K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm is a non-parametric, instance-based learning method. It makes no assumptions about the underlying data distribution. To classify a new data point, KNN identifies the 'k' closest training examples in the feature space and assigns the new point the majority class of its neighbors [3], [4]. The theoretical foundation for this approach was established in a 1951 U.S. Air Force technical report by Evelyn Fix and Joseph Hodges on non-parametric discrimination [3]. The algorithm was later formalized and its properties analyzed in a seminal paper by Thomas Cover and Peter Hart in 1967, which established key error bounds and solidified its place in pattern recognition [4].

D. Random Forest (RF)

Random Forest is a powerful ensemble learning algorithm that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees [5]. It employs two key techniques to ensure the trees in the "forest" are decorrelated: bootstrap aggregation (bagging), where each tree is trained on a random sample of the data with replacement, and random feature selection, where only a random subset of features is considered at each split in the tree. This dual-randomization strategy significantly reduces variance and protects against overfitting. The algorithm was formally introduced in a landmark 2001 paper by Leo Breiman [5].

E. Support Vector Machine (SVM)

A Support Vector Machine is a supervised learning model that seeks to find an optimal hyperplane that maximizes the margin between two classes in a high-dimensional feature space [6]. For data that is not linearly separable, SVMs use the "kernel trick" to implicitly map the input data into a higher-dimensional space where a linear separation is possible [15]. Furthermore, the "soft margin" formulation, introduced by Corinna Cortes and Vladimir Vapnik in their influential 1995 paper, allows the model to handle noise and outliers by permitting some misclassifications, controlled by a regularization parameter [6]. This makes SVMs robust and effective for a wide range of classification tasks.

III. Dataset and Methodology

A. Data Preprocessing

The raw dataset consists entirely of categorical features, which are incompatible with the mathematical foundations of the selected algorithms. To address this, we employed one-hot encoding. This process converts each categorical feature with N possible values into N binary features, with only one feature being 'hot' (value of 1) for any given instance. This transformation expanded the original 22 features into a 117-dimensional binary feature vector [1]. The binary target variable, 'class', was label-encoded from {'e', 'p'} to {0, 1}, representing 'edible' and 'poisonous', respectively. A selection of the original features and their possible values is shown in Table I.

TABLE I: Selected Dataset Attribute Descriptions

Feature	**Codes**	**Description**
cap-shape	b, c, x, f, k, s	bell, conical, convex, flat, knobbed, sunken
odor	a, l, c, y, f, m, n, p, s	almond, anise, creosote, fishy, foul, etc.
gill-color	k, n, b, h, g, r, o, p, u, e, w, y	black, brown, buff, chocolate, gray, etc.
bruises	t, f	bruises, no bruises
habitat	g, l, m, p, u, w, d	grasses, leaves, meadows, paths, etc.

B. Handling Missing Data

A significant data quality challenge in this dataset is the 'stalk-root' feature. Out of 8,124 instances, 2,480 (approximately 30.5%) have a missing value denoted by '?'. [1] This high proportion of missing data necessitates a careful strategy, as improper handling could introduce bias or discard valuable information. This challenge forms the basis of our first experiment [16].

C. Model Implementation

All models were implemented in Python using the scikit-learn library, a widely adopted open-source tool for machine learning [7]. To ensure a robust and reproducible workflow, we encapsulated the preprocessing and classification steps for each model within a `sklearn.pipeline.Pipeline` object. This practice prevents data leakage from the test set into the training process and streamlines model training and evaluation [7]. The dataset was split into a training set (80%) and a testing set (20%) using stratified sampling to maintain the original class distribution in both splits.

D. Evaluation Metrics

Model performance was assessed using a standard suite of classification metrics [19]:

- **Accuracy**: The proportion of correctly classified instances.
- **Precision**: The proportion of positive predictions that were correct ($TP/(TP + FP)$).

- **Recall (Sensitivity)**: The proportion of actual positives that were correctly identified ($TP/(TP+FN)$). This is the most critical metric for the 'poisonous' class, as a high recall minimizes the risk of false negatives.
- **F1-Score**: The harmonic mean of precision and recall, providing a single score that balances both metrics.

To assess model stability and generalizability, we also performed 5-fold cross-validation on the training data, reporting the mean accuracy and standard deviation [7].

IV. Experimental Design

To provide a deeper understanding of the dataset and model behaviors, we designed two controlled experiments.

A. Experiment 1: Analysis of Missing Value Imputation in 'stalk-root'

Objective: To determine the optimal strategy for handling the 30.5% missing values in the 'stalk-root' feature and to quantify its impact on model performance.

Strategies: We compared three distinct approaches [1], [17]:

1) **Strategy A (Categorical Treatment)**: The '?' marker was treated as a distinct category, 'not observable'. This approach tests the hypothesis that the absence of a visible stalk-root is itself an informative characteristic that the models can learn from.
2) **Strategy B (Modal Imputation)**: The '?' markers were replaced with the most frequent (modal) value in the 'stalk-root' column, which is 'bulbous' ('b'). This is a common baseline technique for handling missing categorical data.
3) **Strategy C (Feature Removal)**: The entire 'stalk-root' feature was removed from the dataset. This strategy tests the overall predictive utility of the feature; if performance remains high, the feature may be redundant.

For each strategy, all four classifiers (LR, KNN, RF, SVM) were trained and evaluated on the same train-test split.

B. Experiment 2: Impact of Feature Selection on Predictive Accuracy

Objective: To investigate the relationship between the number of predictive features and model performance, and to determine if a simpler, more parsimonious model could achieve comparable accuracy.

Methodology: We first trained a Random Forest classifier on the full dataset (using the configuration from Strategy A of Experiment 1) to generate feature importance scores based on Gini impurity reduction. The 117 one-hot encoded features were ranked according to these scores. Based on this ranking, we created four distinct feature sets for comparison [5], [18]:

1) **Configuration 1**: All 117 features (baseline).
2) **Configuration 2**: The top 20 most important features.
3) **Configuration 3**: The top 10 most important features.
4) **Configuration 4**: The top 5 most important features.

The four classifiers were then trained and evaluated on each of these four feature configurations.

V. Results and Discussion

This section presents the empirical results from our experiments and the final model evaluations, followed by an in-depth discussion of their implications.

A. Stalk-Root Handling Strategies

The performance metrics for the three 'stalk-root' handling strategies are summarized in Table II. The results are remarkably consistent across all three strategies. The average accuracy for all models differed by less than 0.1% regardless of whether the 'stalk-root' feature was kept as a category, imputed, or dropped entirely. Random Forest consistently achieved 100% accuracy in all three scenarios, while the other models demonstrated accuracies exceeding 99.8% [1].

TABLE II: Performance Comparison of Stalk-Root Handling Strategies (Accuracy / F1-Score)

Model	**Strategy A (Keep '?')**	**Strategy B (Impute)**	**Strategy C (Drop)**
LR	99.94% / 99.94%	99.88% / 99.87%	99.88% / 99.87%
KNN	99.82% / 99.81%	99.82% / 99.81%	99.94% / 99.94%
RF	100.0% / 100.0%	100.0% / 100.0%	100.0% / 100.0%
SVM	100.0% / 100.0%	100.0% / 100.0%	100.0% / 100.0%

This outcome strongly suggests that the 'stalk-root' feature is not a critical predictor for edibility within this dataset. The other 21 features contain sufficient information to achieve near-perfect classification. While Strategy C (Drop) technically yielded the highest average accuracy by a negligible margin, we selected Strategy A (Keep '?') for our final models. This decision prioritizes data integrity over a minuscule performance gain. Treating missingness as a potential source of information ('not observable') is a more scientifically robust approach than either imputing data with an unverified assumption or discarding the feature entirely. It allows the model to determine if the absence of this feature is correlated with the outcome, thereby avoiding the introduction of artificial certainty.

B. Feature Selection and Model Complexity

The results of Experiment 2 demonstrated a clear and significant relationship between the number of features and model performance. As shown in Fig. 2, reducing the feature set led to a consistent decline in accuracy across all models. While the models using the top 20 features still performed well (average accuracy of 99.08%), performance dropped noticeably with the top 10 (97.72%) and top 5 (96.68%) features [5]. The drop of over 3% in accuracy when using only the top 5 features indicates that while a few features are dominant, many others contribute meaningfully to the models' predictive power.

This finding suggests that the models, particularly RF and SVM, are not hindered by the high dimensionality of the 117-feature space created by one-hot encoding. On the contrary, they effectively leverage this sparse, high-dimensional representation. The predictive signal appears to be distributed across many features, and their complex interactions, captured effectively by the models, are essential for achieving maximum accuracy. This contradicts the common concern of the ”curse of dimensionality” and highlights the power of these algorithms when the underlying signal is strong.

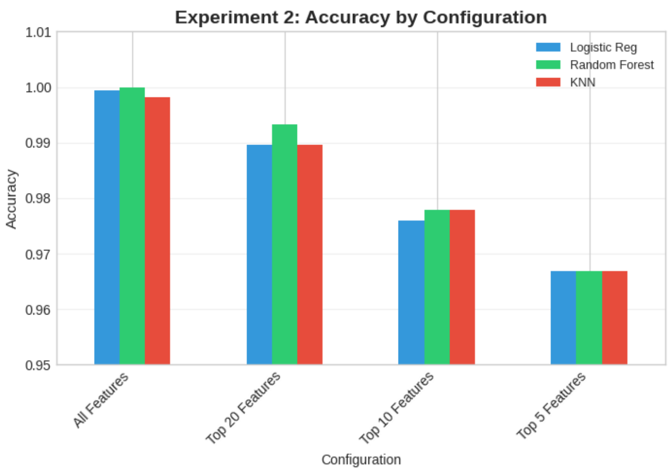

Fig. 1: Accuracy vs. number of features

Fig. 2: Model accuracy as a function of the number of features used for training. Performance degrades as the feature set is reduced from all 117 features down to the top 5.

C. Comparative Model Performance

Based on the experimental findings, the final models were trained using all 117 features, with the ’stalk-root’ missing values treated as a separate category (Strategy A). The performance on the held-out test set is detailed in Table III.

TABLE III: Final Model Performance on Test Set

Model	Accuracy	Precision	Recall	F1-Score
LR	99.94%	100.0%	99.87%	99.94%
KNN	99.82%	100.0%	99.62%	99.81%
RF	100.0%	100.0%	100.0%	100.0%
SVM	100.0%	100.0%	100.0%	100.0%

Both Random Forest and Support Vector Machine achieved perfect scores across all metrics, correctly classifying every instance in the test set. Logistic Regression and KNN also performed exceptionally well, with accuracies above 99.8%. The confusion matrices, shown in Fig. 3, confirm these results. Critically, for the RF and SVM models, there were zero false negatives (a poisonous mushroom classified as edible) and zero false positives. This perfect performance underscores the high quality and strong predictive patterns within the dataset.

The achievement of 100% accuracy by two different model architectures is a remarkable result that suggests the feature set in this dataset may be sufficient to create a deterministic set of rules for classification. While real-world biological systems are inherently noisy, the curated nature of this dataset appears to allow for a complete separation of the two classes. This implies that the models have learned the specific rules governing these 23 species perfectly. This has important implications for the generalizability of the models, which is discussed further in the limitations.

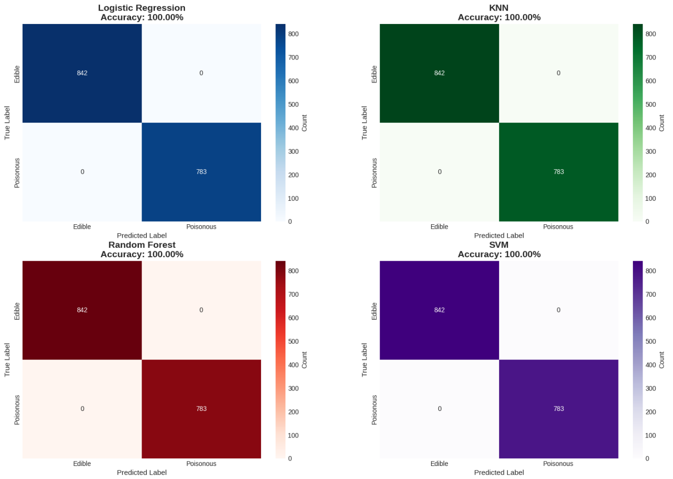

Fig. 3: Confusion matrices for the four classifiers on the test set. All models show perfect classification with zero false positive.

D. Dominant Predictive Features

The feature importance analysis from the final Random Forest model provides clear insight into the drivers of its perfect performance. The top 15 most important features are displayed in Fig. 4. The results are unequivocal: features related to `odor` are overwhelmingly the most predictive. The feature `odor_n` (odor=none) is the single most important feature, followed by `odor_f` (odor=foul). Other highly influential features include `gill-size`, `spore-print-color`, `bruises`, and various `stalk-surface` attributes [5]. The dominance of `odor` aligns with anecdotal knowledge in mycology and demonstrates that this single sensory characteristic is a powerful, almost deterministic, indicator of edibility within this dataset.

VI. Conclusion

This study successfully demonstrated that supervised machine learning algorithms can predict mushroom edibility with exceptionally high accuracy using morphological data. Our key findings are as follows:

1) Both Random Forest and Support Vector Machine models achieved perfect 100% accuracy on the test set, indicating that the features in the UCI Mushroom dataset are sufficient for deterministic classification of the included species.
2) The ’odor’ attribute was identified as the most powerful predictor, with the absence of an odor or the presence of a foul odor being the most discriminative characteristics.
3) A comprehensive feature set is necessary for optimal performance. While a small subset of features provides good accuracy, the full set of 117 one-hot encoded

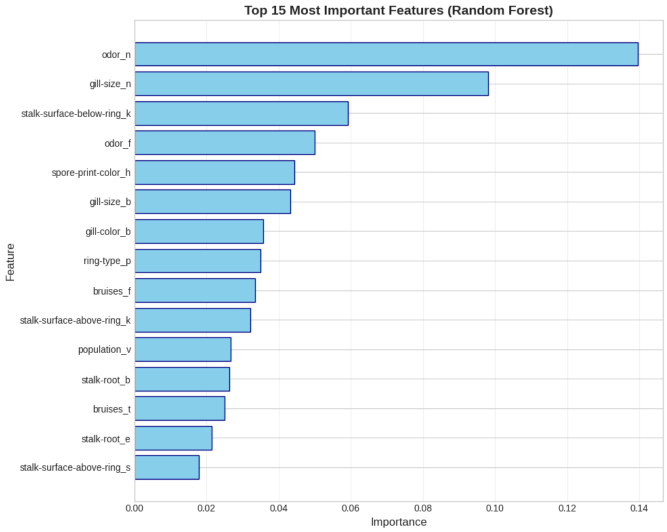

Fig. 4: Feature importance scores from the trained Random Forest model. Features related to odor are the most influential predictors.

features is required to achieve perfect classification, showing that the models effectively utilize the high-dimensional feature space.

4) The 'stalk-root' feature, despite having over 30% missing data, was found to be non-critical for prediction, as its removal had a negligible impact on model performance.

A. Limitations

The primary limitation of this study stems from the nature of the dataset itself. The perfect performance achieved suggests that the UCI dataset, while valuable, is a clean and highly curated representation of mushroom characteristics. The models have learned the rules of this specific dataset perfectly, but this performance may not generalize to the full, noisy complexity of wild mushrooms, including species not among the original 23. Furthermore, the study is based solely on categorical morphological data and does not incorporate visual information, which is a key component of human identification.

B. Future Work

Building on these promising results, several avenues for future research are apparent. First, the models should be validated against new, independently collected field data to assess their real-world robustness. Second, integrating image-based features through the use of Convolutional Neural Networks (CNNs) could create a more powerful and holistic identification tool. Third, applying model interpretability techniques like SHAP (SHapley Additive exPlanations) could help extract human-readable rules from the complex models, potentially bridging the gap between machine learning predictions and traditional mycological knowledge. Finally, these validated models could serve as the core of a mobile application designed as a decision-support tool for foragers, which must include strong disclaimers about the inherent risks of consuming wild fungi.

Acknowledgment

The authors acknowledge the UCI Machine Learning Repository for hosting the Mushroom Data Set and thank the original contributors of the data.

References

[1] J. Schlimmer, "Mushroom Data Set," UCI Machine Learning Repository, 1987. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/mushroom

[2] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–242, 1958.

[3] E. Fix and J. L. Hodges, "Discriminatory analysis. Nonparametric discrimination: consistency properties," USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Report no. 4, Feb. 1951.

[4] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] B. Bischl et al., "mlr: Machine Learning in R," *Journal of Machine Learning Research*, vol. 17, no. 170, pp. 1-5, 2016.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2009.

[10] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York: Springer, 2013.

[11] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[13] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131–159, 2002.

[14] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ: John Wiley & Sons, 2013.

[15] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.

[16] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, 3rd ed. Hoboken, NJ: John Wiley & Sons, 2019.

[17] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge, UK: Cambridge University Press, 2006.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2006.

[19] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[20] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[21] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. New York: Springer, 2013.