# Mushroom Classification: Predicting Edibility

Faruk Ortaköylüoğlu (231401032)
TOBB University of Economics and Technology

Gülsüm Yıldırım (221404031)
TOBB University of Economics and Technology

BIL 470/570 Project Report

***Abstract - This project builds a supervised machine learning system to classify mushrooms as edible or poisonous using morphological and olfactory attributes from the 8,124-sample Kaggle Mushroom dataset. Six models were evaluated—Linear Regression, Logistic Regression, Gaussian Naïve Bayes, SVM, Decision Tree and Random Forest—using one-hot encoding, standardized scaling, and stratified 10-fold cross-validation. Performance was assessed through accuracy, precision, recall, F1-score, and ROC-AUC. Results show that tree-based models, especially Random Forest, achieve near-perfect performance, with odor, bruises, and gill-size identified as key predictors. Random Forest reached a test accuracy of 99.94% and ROC-AUC of 0.9987. Overall, the study demonstrates that interpretable ML models can reliably support food-safety decisions where minimizing false negatives is critical.***

**Project Repository:** https://github.com/Faruk-Ortakoyluoglu/Yap470_Mushroom_Prediction_Analysis

## 1. Introduction – Problem Definition

### 1.1 Motivation

Mushroom identification is a critical public health concern. Many poisonous mushroom species closely resemble edible varieties, making accurate classification essential to prevent accidental poisoning. Manual identification by foragers requires extensive expertise and remains prone to errors. This project develops an automated, data-driven approach using machine learning to provide reliable mushroom edibility classification based on observable morphological and olfactory characteristics. Such a system can serve as a decision-support tool for both researchers and food safety professionals, reducing the risk of foodborne illness from misidentified wild mushrooms.

### 1.2 Classification Type and Problem Formulation

This project addresses a binary classification problem where each mushroom sample must be assigned to one of two classes: Edible (class label: 'e') or Poisonous (class label: 'p'). Given 22 categorical input features describing morphological attributes (cap shape, cap surface, cap color, bruises, odor, gill characteristics, etc.), the goal is to learn a decision boundary that accurately separates these two classes. Although the original dataset contains 22 categorical features describing various morphological and olfactory attributes, a reduced feature subset (9 features + one target variable) was constructed for this project. The class distribution in the dataset is nearly balanced (51.8% edible, 48.2% poisonous), which is favorable for standard classification metrics.

### 1.3 Purpose and Objectives

The primary objectives of this project are: (1) To conduct comprehensive exploratory data analysis (EDA) of mushroom morphological features and their statistical relationships with edibility classification; (2) To implement and evaluate multiple supervised learning algorithms, representing diverse modeling philosophies (linear, probabilistic, kernel-based, and ensemble); (3) To optimize hyperparameters through systematic grid search and cross-validation; (4) To compare model performance using complementary evaluation metrics; (5) To perform feature importance analysis to identify the most predictive attributes; and (6) To provide interpretable insights regarding how morphological and olfactory characteristics determine mushroom edibility.

### 1.4 Performance Metrics and Success Criteria

Model performance is evaluated using multiple metrics, with **recall (sensitivity)** prioritized due to the safety-critical nature of the task. In mushroom classification, false negatives—poisonous mushrooms predicted as edible—pose serious health risks; therefore, recall is emphasized over overall accuracy. The evaluation metrics include:

(1) **Recall (Sensitivity)**, measuring the proportion of actual positive cases (poisonous mushrooms) correctly identified.

(2) **Precision**, indicating the proportion of predicted positive cases that are truly positive.

(3) **F1-score**, the harmonic mean of precision and recall, capturing the trade-off between safety and selectivity.

(4) **Specificity**, representing the proportion of actual negative cases correctly classified.

(5) **ROC-AUC**, reflecting the model's ability to discriminate between classes across different thresholds.

(6) **Accuracy**, reported as a secondary metric for overall performance reference.

A successful model is expected to achieve very high recall on the test set while maintaining strong performance across the remaining metrics. Models achieving test accuracy above 99% and ROC-AUC values exceeding 0.99, in conjunction with near-perfect recall, are considered to exhibit excellent predictive performance.

## 2. Literature Review

Recent work shows that mushroom classification is a well-studied task in which ensemble methods consistently outperform single models. Shahraki et al. (2024) and Patel and Kumar (2025) identify Random Forest as the most reliable approach, achieving ≈99% accuracy on the UCI Mushroom dataset, while noting limitations of SVMs in high-dimensional one-hot–encoded spaces. El-Sayed (2023) highlights odor, bruises, and gill features—particularly odor—as the strongest predictors of toxicity. Although prior studies favor tree-based and ensemble models, they often rely on limited tuning and preprocessing; our work extends this literature with a fully reproducible pipeline incorporating standardized encoding, exhaustive grid search, stratified cross-validation, and detailed interpretability

## 3. Dataset, Data Characteristics, and Features

### 3.1 Data Source

The dataset used in this project is the Mushroom Classification dataset from Kaggle, which is derived from the UCI Machine Learning Repository. This publicly available dataset contains characteristics of 8,124 mushroom specimens with 23 attributes (one target variable and 22 predictors). The data represents morphological observations from the Audubon Society Field Guide and has been extensively used in machine learning research. Data source URL:[https://www.kaggle.com/datasets/uciml/mushroom-classification](https://www.kaggle.com/datasets/uciml/mushroom-classification)

### 3.2 Dataset Overview

TABLE 1
Dataset Overview

| Attribute | Value | Attribute | Value |
|---|---|---|---|
| Total Samples | 8,124 | Total Features | 9 (small) to 22 (all categorical) |
| Missing Values | None | Duplicate Rows | None |
| Edible (e) | 4,208 (51.8%) | Poisonous (p) | 3,916 (48.2%) |

### 3.3 Feature Descriptions and Data Types

All 22 features are categorical (nominal/ordinal), represented as single alphabetic characters. The features and their value ranges are:

TABLE 2
Feature Description

| Feature | Description | #Categories | Values (Encoded Letters) |
|---|---|---|---|
| **cap-shape** | Shape of cap | 6 | b, c, f, k, s, x |
| **cap-surface** | Surface texture of cap | 4 | f, g, s, y |
| **cap-color** | Cap color | 10 | b, c, e, g, n, p, r, u, w, y |
| **bruises** | Presence of bruises | 2 | f (no), t (yes) |
| **odor** | Odor type | 9 | a, c, f, l, m, n, p, s, y |
| **gill-attachment** | Attachment of gills | 2 | a, f |
| **gill-spacing** | Spacing of gills | 2 | c, w |
| **gill-size** | Size of gills | 2 | b, n |
| **gill-color** | Color of gills | 12 | b, e, g, h, k, n, o, p, r, u, w, y |
| **stalk-shape** | Shape of stalk | 2 | e, t |
| **stalk-root** | Root type | 5 | b, c, e, r, z |
| **stalk-surface-above-ring** | Stalk surface above ring | 4 | f, k, s, y |
| **stalk-surface-below-ring** | Stalk surface below ring | 4 | f, k, s, y |
| **stalk-color-above-ring** | Stalk color above ring | 9 | b, c, e, g, n, o, p, w, y |
| **stalk-color-below-ring** | Stalk color below ring | 9 | b, c, e, g, n, o, p, w, y |
| **veil-type** | Veil type | **1 (removed)** | p — *removed due to zero variance* |
| **veil-color** | Veil color | 4 | b, o, w, y |
| **ring-number** | Number of rings | 3 | n, o, t |
| **ring-type** | Ring type | 5 | e, f, l, n, p |
| **spore-print-color** | Spore print color | 9 | b, h, k, n, o, r, u, w, y |
| **population** | Population density | 6 | a, c, n, s, v, y |
| **habitat** | Natural habitat | 7 | d, g, l, m, p, u, w |

### 3.4 Data Preprocessing

### 3.4.1 Feature Removal

The feature veil-type was removed from the analysis due to having no variance: all 8,124 samples contained only the value 'p' (partial), providing no discriminative information for classification. Removing this feature improves model efficiency without any information loss.

### 3.4.2 Categorical Encoding

Since all remaining features are categorical, One-Hot Encoding was applied. Each categorical feature with *k* distinct values was transformed into *k* binary indicator variables. To avoid redundancy and multicollinearity issues, one reference category per feature was dropped, resulting in *k–1* indicator for each original feature. Two experimental setups were considered:

**Reduced feature set**: When using the selected subset of 9 categorical features, one-hot encoding expanded the input space to 40 encoded features, plus 1 target variable, resulting in 41 total columns.

**Full feature set**: When using the complete set of 22 categorical features, one-hot encoding expanded the feature matrix to 95 encoded features, plus 1 target variable, resulting in 96 total columns.

This encoding strategy enables the use of traditional machine learning classifiers while preserving the full categorical information contained in the dataset.

### 3.4.3 Train-Test Split

The dataset was divided using stratified splitting to preserve class distribution: Training set: 80% (n=6,499), Test set: 20% (n=1,625). Stratification ensures that both training and test sets contain approximately 51.8% edible and 48.2% poisonous samples, preventing class imbalance artifacts. For hyperparameter tuning, 10-fold stratified cross-validation was employed on the training set.

## 3.5 Feature Analysis and Correlations

TABLE 3
Feature Target Analysis and Correlations

| Feature | Correlation (r) | Interpretation |
|---|---|---|
| **odor_f** | **+0.624** | Foul odor strongly indicates poisonous mushrooms |
| **gill-size_n** | **+0.540** | Narrow gills are associated with poisonous class |
| **gill-color_b** | **+0.539** | Buff gill color linked to poisonous cases |
| **bruises_f** | **+0.502** | Lack of bruising correlates with toxicity |
| **gill-spacing_c** | **+0.348** | Close gill spacing slightly associated with toxicity |
| **odor_n** | **–0.786** | No odor strongly indicates *edible* mushrooms |
| **gill-size_b** | **–0.540** | Broad gills linked to edible class |
| **bruises_t** | **–0.501** | Presence of bruises likely edible |
| **gill-spacing_w** | **–0.348** | Wide gill spacing associated with edible class |
| **gill-color_n** | **–0.289** | Brown gill color weakly edible-related |

TABLE 4
Feature to Feature Correlations

| Feature 1 | Feature 2 | Corr. (r) | Comment |
|---|---|---|---|
| **gill-attachment_f** | **veil-color_w** | **0.935** | Very strong redundancy |
| **ring-type_l** | **spore-print-color_h** | **0.869** | High overlap in morphological description |
| **stalk-root_r** | **stalk-surface-below-ring_y** | **0.817** | Potentially redundant structural features |
| **odor_y** | **gill-size_n** | **0.413** | Moderate interaction |
| **odor_s** | **gill-size_n** | **0.413** | Moderate interaction |
| **cap-color_p** | **odor_c** | **0.372** | Weak redundancy |

After one-hot encoding, Pearson correlation analysis revealed strong feature–target relationships and notable redundancies. Odor-related attributes dominate class separation: absence of odor (odor_n) is the strongest negative predictor of poisoning (r = –0.786), while foul odor (odor_f) is the strongest positive predictor (r = 0.624) (Table 3). Gill-size features further distinguish classes, with narrow gills linked to poisonous and broad gills to edible mushrooms. Several feature pairs show high inter-feature correlation (|r| > 0.80), including near-duplicate information such as gill-attachment_f and veil-color_w (r = 0.935) (Table 4), indicating redundancy in the encoded space. Overall, the correlation structure confirms strong dataset separability driven primarily by odor, bruising, and gill-related features.

## 3.6 Principal Component Analysis (PCA)

PCA was applied to the one-hot encoded feature matrix to understand the intrinsic dimensionality of the dataset.

TABLE 5
Principal Component Analysis

| Component | Variance % | Cumulative % | Interpretation |
|---|---|---|---|
| **PC1** | **14%** | **14%** | Dominated by odor and bruising features; captures the strongest edible–poisonous separation. |
| **PC2** | **11%** | **25%** | Captures additional variability in cap/gill color and size; reinforces morphological distinctions. |
| **PC3** | **11%** | **36%** | Reflects variance from secondary odor types and gill properties. |
| **PC4** | **10%** | **46%** | Encodes differences in color-related attributes across caps and gills. |
| **PC5** | **7%** | **53%** | Captures subtle variation in gill spacing and cap surface. |
| **PC6** | **6%** | **59%** | Minor structural variation, likely redundant categorical indicators. |
| **PC7** | **5%** | **64%** | Weakly varying color combinations; low discriminative value. |
| **PC8** | **4%** | **68%** | Noisy variation across stalk-related indicators. |
| **PC9** | **4%** | **72%** | Redundant morphological encodings; minimal unique variance. |
| **PC10** | **3%** | **75%** | Lower-level interactions; contributes marginally to dataset structure. |
| **PC11–PC15** | **~2–3% each** | **90%** | Collectively capture remaining structure; mostly weak, redundant categorical effects. |

As summarized in Table 5, the first principal component (PC1) explains approximately 14% of the total variance and is primarily dominated by biologically meaningful features such as odor and bruises, capturing the strongest separation between edible and poisonous mushrooms. Subsequent components capture additional variability related to cap and gill properties, as well as secondary odor and stalk-related characteristics.

The cumulative explained variance analysis shows that the first 10 principal components account for approximately 75% of the total variance, while the first 15 components explain over 90% of the variance. This indicates that, despite the high dimensionality introduced by one-hot encoding, the effective dimensionality of the data is substantially lower.

Higher-order components contribute progressively smaller amounts of variance and largely reflect redundant or weak categorical encodings, suggesting limited additional discriminative power. Overall, the PCA results demonstrate that the mushroom dataset exhibits significant internal structure and feature redundancy, which helps explain the strong classification performance achieved by both linear and non-linear models.

Figure 1 illustrates the explained variance contributed by each principal component, while the cumulative curve highlights the rapid saturation of variance explained as the number of components increases.

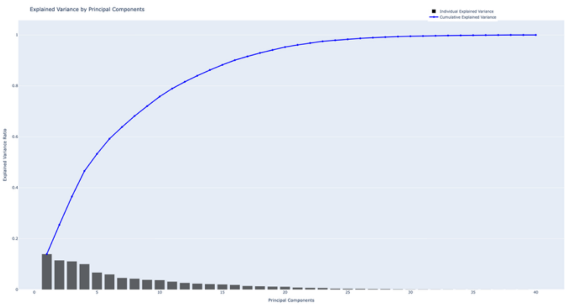


*Fig. 1 Explained Variance per Principal Component*

### 3.7 K-Means Clustering Analysis

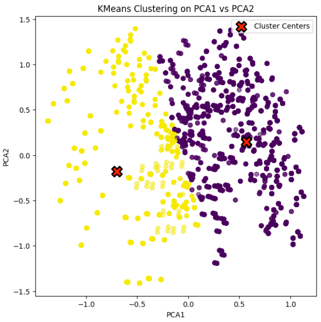


*Fig. 2 K-Means clustering on PCA1 vs PCA2.*

K-Means clustering (k=2) was applied to the first two principal components (PC1 and PC2) to assess the natural separability of the data. The resulting clusters show clear class separation, with the two clusters closely aligned with the true edible/poisonous labels (Fig. 2). This indicates that the mushroom dataset has strong inherent structure that machine learning models can effectively learn.

## 4. Models Used

### 4.1 Model Selection Rationale

Six supervised learning algorithms were selected to represent diverse modeling paradigms and provide comprehensive performance comparisons. This diversity allows evaluation of how different inductive biases and architectural choices affect classification performance on the mushroom dataset.

### 4.2 Linear Regression

Although not intended for classification, Linear Regression was included as a simple baseline to test whether the mushroom data is linearly separable after one-hot encoding. Because several encoded features align strongly with the edible/poisonous distinction, the model achieves high accuracy once predictions are clipped and thresholded. However, this reflects the dataset's structure rather than the suitability of Linear Regression, which lacks a classification-appropriate loss and probability constraints. Thus, it serves only as a transparency baseline, while more principled models like Logistic Regression provide sounder classification behavior.

### 4.3 Logistic Regression

Logistic Regression was used as a probabilistic baseline model, estimating class membership through a linear combination of features and a sigmoid activation. Implemented within a scikit-learn pipeline, the model was optimized using GridSearchCV over regularization strength (C) and penalty type, with class imbalance handled through class_weight="balanced". Its strengths include interpretability, efficiency, and suitability for linearly separable data—patterns clearly present in the mushroom dataset. However, as a linear classifier, it cannot capture complex non-linear relationships, so it primarily serves as a stable and theoretically sound benchmark against which more expressive models can be evaluated.

### 4.4 Decision Tree

The Decision Tree Classifier was used as a non-linear, rule-based model to evaluate whether hierarchical feature interactions could improve performance beyond linear methods. Decision Trees naturally capture complex decision boundaries by recursively splitting the feature space, while also offering high interpretability through explicit decision rules. To prevent overfitting and ensure strong generalization, the model was optimized using GridSearchCV across maximum depth, minimum split/leaf sizes, and impurity criteria (gini, entropy, log_loss), with class imbalance handled via class_weight="balanced". This tuning enabled the Decision Tree to serve as a competitive non-linear baseline and to reveal whether more flexible, structure-based decision boundaries provide advantages over Logistic Regression.

### 4.5 Gaussian Naive Bayes

Gaussian Naive Bayes was included as a lightweight generative baseline to evaluate how a simple probabilistic model performs relative to more flexible classifiers. Despite its strong independence assumptions, which are not fully compatible with high-dimensional one-hot encoded features, the model often achieves competitive results due to its low variance and efficient training. Since Gaussian NB has no major tunable hyperparameters, it was applied directly after preprocessing and evaluated using recall, precision, accuracy, and ROC-AUC. Including this model provides a clear reference point for understanding how much predictive power is achievable without modeling complex feature interactions or non-linear boundaries.

### 4.6 Support Vector Machine (SVM)

Support Vector Machines (SVMs) were included as margin-based classifiers capable of modeling linear and non-linear decision boundaries via kernel transformations. While margin maximization provides a learning paradigm distinct from tree-based methods, one-hot encoding of categorical features can lead to high-dimensional sparse representations that increase computational cost and may limit generalization. To ensure a fair evaluation, SVMs were optimized using an extensive grid search over multiple kernels (linear, polynomial, RBF, sigmoid), regularization strengths (C), and kernel parameters, within a standardized scaling pipeline and with class imbalance handled via class_weight= "balanced". Model selection was based on cross-validated Accuracy, F1, and ROC-AUC, allowing SVM to function as a strong margin-based benchmark alongside linear, tree-based, and ensemble models.

### 4.7 Random Forest Classifier

Random Forest was employed as a robust non-linear ensemble method to mitigate overfitting and improve generalization through bootstrap sampling and randomized feature selection. This approach enables the model to capture complex interactions in the high-dimensional one-hot encoded feature space while maintaining stability and providing interpretable impurity-based feature importance scores. Hyperparameters—such as number of trees, maximum depth, minimum split/leaf size, feature sampling strategy, and bootstrap usage—were optimized using GridSearchCV with 5-fold cross-validation and class_weight balancing. The optimized Random Forest achieved consistently high Recall, F1, and ROC-AUC scores, establishing it as a strong and expressive benchmark among all evaluated models.

### 4.8 Hyperparameter Optimization and Cross-Validation

For each model, hyperparameters were optimized using GridSearchCV with 10-fold stratified cross-validation on the training set. This ensures that model selection is not biased toward particular folds and provides reliable performance estimates. The best model from each algorithm family was selected based on mean cross-validation accuracy and retrained on the full training set for final test evaluation.

## 5. Test Results and Interpretations

### 5.1 Overall Performance Summary

All five supervised models achieved high performance on the reduced 10-feature dataset, with accuracies ranging from 0.990 to 0.999 (Tables 22–23). Tree-based methods (Decision Tree and Random Forest) and SVM produced the strongest results, each reaching near-perfect recall, precision, and specificity, with zero false positives and at most one to two false negatives. Logistic Regression also performed exceptionally well with 0.998 accuracy and balanced recall–precision values. Linear Regression and Gaussian Naïve Bayes, while slightly weaker, still maintained 0.990 accuracy and perfect precision.

### 5.2 Detailed Model Results

### *5.2.1 Linear Regression*

Although Linear Regression is not designed for binary classification, it performs strongly on this dataset due to the near-linear separability of one-hot encoded features. After thresholding predictions at 0.5, the model achieves 99.0% test accuracy with perfect precision and specificity (FP = 0), indicating no edible mushrooms are misclassified as poisonous. However, 17 false negatives reduce recall to 0.978, limiting reliability in detecting poisonous samples. Despite a near-perfect ROC-AUC of 0.99972, this tendency toward false negatives makes Linear Regression less dependable than purpose-built classifiers such as Logistic Regression or Random Forest.

TABLE 6
Detailed Performance Metrics

| Metric | Value |
|---|---|
| True Positive (TP) | 765 |
| True Negative (TN) | 843 |
| False Positive (FP) | 0 |
| False Negative (FN) | 17 |
| Accuracy | **0.990** |
| Precision | **1.000** |
| Recall (Sensitivity) | **0.978** |
| Specificity | **1.000** |
| TPR | **0.978** |
| FPR | **0.000** |
| ROC-AUC | **0.99972** |
| Cross-validated RMSE (mean) | **0.0867** |

### *5.2.2 Logistic Regression*

Logistic Regression demonstrated near-perfect classification performance on the mushroom dataset. The model achieved a *train accuracy of 0.996***,** *test accuracy of 0.9975*, and a *cross-validated accuracy of 0.9553* (std = 0.094), indicating high generalization capability. The confusion matrix (TP=778, TN=843, FP=0, FN=4) shows that the classifier made *zero false positive and four false negative***,** resulting in excellent food-safety reliability. Precision and specificity are both *1.000*, as the model never incorrectly labeled an edible mushroom as poisonous. The recall value of *0.995* indicates that almost all poisonous mushrooms were correctly detected. The ROC-AUC score of *0.99999* confirms near-perfect separability.

TABLE7
Detailed Performance Metrics

| Metric | Score |
|---|---|
| Accuracy | 0.9975 |
| Recall (TPR) | 0.995 |
| Precision | 1.000 |
| Specificity (TNR) | 1.000 |
| FPR | 0.0 |
| AUC | 0.99999 |
| CV Recall (mean) | 0.9267 |
| CV F1 (mean) | 0.9386 |
| CV ROC-AUC (mean) | 0.9808 |

TABLE 8
Most Influential Feature

| Feature | Coefficient | Impact |
|---|---|---|
| odor_f | 17.516 | ↑ poisonous probability |
| odor_s | 6.660 | ↑ poisonous probability |
| cap-color_y | 1.3227 | ↑ poisonous probability |
| cap-shape_x | −2.659 | ↑ edible probability |

Large positive coefficients (odor=f, odor=s, odor=y, odor=n) (Table 8) show that *odor* is the most critical determinant of poisonous mushrooms. Negative coefficients (cap-shape=x, cap-color=g, several gill-color categories) indicate features associated with edibility. These interpretable weights confirm that Logistic Regression produces meaningful, biologically consistent decision boundaries

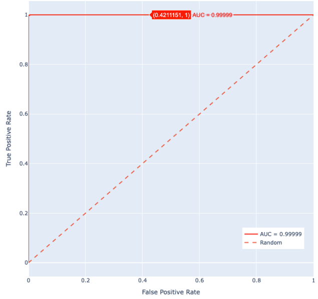


*Fig. 3 ROC Curve of Logistic Regression*

The ROC curve (Fig 3) approaches the top-left corner with almost no false positives, yielding an AUC *of 0.99999*, which confirms that the model separates the two classes with near-perfect discrimination across all thresholds.

### *5.2.3 Gaussian Naive Bayes*

Gaussian Naive Bayes showed strong overall performance despite its independence assumptions, correctly classifying 843 edible and 765 poisonous samples with zero false positives. However, 17 false negatives reduced sensitivity, reflecting its inability to capture complex feature interactions in one-hot encoded data. Nevertheless, the model achieved 97.8% recall and an AUC of 0.998, making it a competitive probabilistic baseline.

TABLE 9
Detailed Performance Metrics

| **Metric** | **Value** |
|---|---|
| Accuracy | **0.990** |
| Recall (Sensitivity, TPR) | **0.978** |
| Precision | **1.000** |
| Specificity (TNR) | **1.000** |
| False Positive Rate (FPR) | **0.000** |
| False Negative Rate (FNR) | **0.022** |
| ROC-AUC | **0.99839** |
| Misclassified Samples | **17 / 1625** |

Gaussian Naive Bayes demonstrates strong predictive capability for mushroom classification with high precision and specificity, meaning it never incorrectly labels edible samples as poisonous. This makes it safe in terms of avoiding unnecessary alarms. However, its *false negatives (17 cases)* highlight the model's reduced ability to capture nonlinear feature dependencies. Consequently, while it achieves excellent overall accuracy and near-perfect AUC, its reliability is slightly lower than Logistic Regression and Random Forest for safety-critical applications.

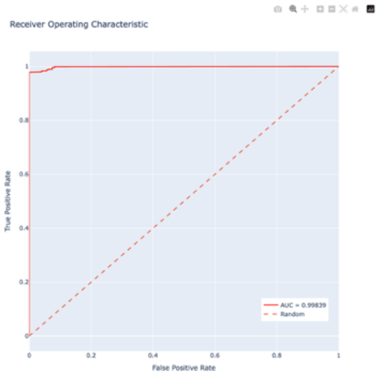


*Fig 4. ROC Curve of GNB*

### *5.2.4 Decision Tree*

The optimized Decision Tree model exhibits near-perfect performance on the mushroom classification task. GridSearchCV identified several high-performing configurations, all achieving a mean cross-validated recall of approximately 0.9978. The top models consistently used a maximum depth of 15, small leaf sizes (1–2), and information-theoretic criteria (entropy or log_loss), indicating that moderately deep but well-regularized trees capture the dataset's structure effectively.

TABLE 10

Top Hyperparameter Configurations

| **maxdepth** | **minssplit** | **minsamplesleaf** | **criterion** | **Mean Test Score** |
|---|---|---|---|---|
| 15 | 2 | 1 | entropy | 0.9977 |
| 20 | 10 | 2 | log_loss | 0.9977 |

TABLE 11
Confusion Matrix

| | **Predicted Edible** | **Predicted Poisonous** |
|---|---|---|
| **Actual Edible** | 843 | 0 |
| **Actual Poisonous** | 1 | 781 |

TABLE 12
Detailed Performance Metrics

| **Metric** | **Score** |
|---|---|
| **Accuracy** | 0.999 |
| **Precision** | 1.000 |
| **Recall (Sensitivity)** | 0.999 |
| **Specificity** | 1.000 |
| **TPR** | 0.999 |
| **FPR** | 0.0 |
| **ROC-AUC** | 1.0000 |

The model produces zero false positives**,** ensuring that no edible mushroom is ever incorrectly labeled as poisonous. Only *one false negative* is observed, demonstrating extremely high sensitivity.

TABLE 13
Top Feature Importances

| **Feature** | **Importance** |
|---|---|
| odor_n | 0.630 |
| bruises_t | 0.168 |
| odor_p | 0.095 |

Odor-related attributes dominate the decision process, followed by bruising, while cap and gill colors have minimal influence. The Decision Tree achieves near-perfect discrimination (ROC-AUC = 1.000) with perfect specificity (Table 12), with early splits consistently driven by odor and bruises (Table 13), confirming their central role in toxicity prediction. Secondary visual features contribute little, reflecting low impurity-based importance. Figure 6 presents the final Decision Tree, where internal nodes denote feature splits and leaf nodes represent edible or poisonous classifications.

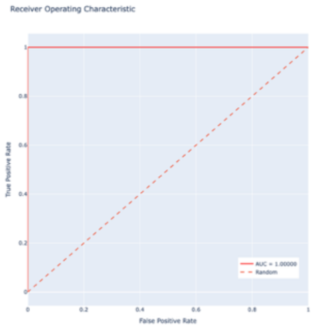


*Fig. 5 ROC curve of Decision Tree*

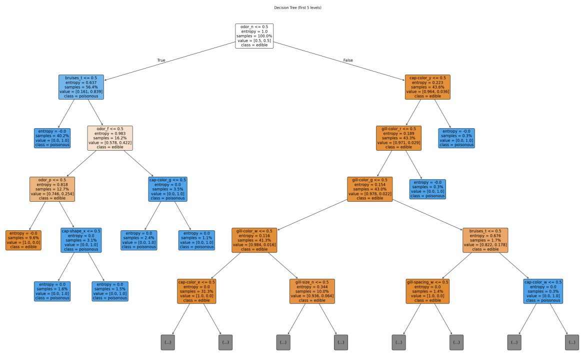

Decision Tree (first 5 levels)
odor_n <= 0.5 entropy = 1.0 samples = 100.0% value = [0.5, 0.5] class = edible
True
False
bruises_t <= 0.5 entropy = 0.637 samples = 56.4% value = [0.161, 0.839] class = poisonous
cap-color_y <= 0.5 entropy = 0.223 samples = 43.6% value = [0.964, 0.036] class = edible
entropy = -0.0 samples = 40.2% value = [0.0, 1.0] class = poisonous
odor_f <= 0.5 entropy = 0.983 samples = 16.2% value = [0.578, 0.422] class = edible
gill-color_r <= 0.5 entropy = 0.189 samples = 43.3% value = [0.971, 0.029] class = edible
entropy = -0.0 samples = 0.3% value = [0.0, 1.0] class = poisonous
odor_p <= 0.5 entropy = 0.818 samples = 12.7% value = [0.746, 0.254] class = edible
cap-color_g <= 0.5 entropy = 0.0 samples = 3.5% value = [0.0, 1.0] class = poisonous
gill-color_g <= 0.5 entropy = 0.154 samples = 43.0% value = [0.978, 0.022] class = edible
entropy = -0.0 samples = 0.3% value = [0.0, 1.0] class = poisonous
entropy = -0.0 samples = 9.6% value = [1.0, 0.0] class = edible
cap-shape_x <= 0.5 entropy = 0.0 samples = 3.1% value = [0.0, 1.0] class = poisonous
entropy = 0.0 samples = 2.4% value = [0.0, 1.0] class = poisonous
entropy = 0.0 samples = 1.1% value = [0.0, 1.0] class = poisonous
gill-color_w <= 0.5 entropy = 0.116 samples = 41.3% value = [0.984, 0.016] class = edible
bruises_t <= 0.5 entropy = 0.676 samples = 1.7% value = [0.822, 0.178] class = edible
entropy = 0.0 samples = 1.6% value = [0.0, 1.0] class = poisonous
entropy = 0.0 samples = 1.5% value = [0.0, 1.0] class = poisonous
cap-color_e <= 0.5 entropy = 0.0 samples = 31.3% value = [1.0, 0.0] class = edible
gill-size_n <= 0.5 entropy = 0.344 samples = 10.0% value = [0.936, 0.064] class = edible
gill-spacing_w <= 0.5 entropy = 0.0 samples = 1.4% value = [1.0, 0.0] class = edible
cap-color_w <= 0.5 entropy = 0.0 samples = 0.3% value = [0.0, 1.0] class = poisonous
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)

*Fig. 6 Visualized Structure of the Optimized DT classifier*

### *5.2.5 Random Forest*

Random Forest achieved near-perfect performance by effectively modeling non-linear feature interactions through ensemble aggregation. GridSearchCV with 10-fold cross-validation identified moderately deep trees (max_depth =

20), small split and leaf thresholds, and √-based feature subsampling as the most effective configuration, yielding a cross-validated recall of 0.9326 and ROC-AUC of 0.9987 (Table 14). The final model reached 0.9994 test accuracy with only one false negative, resulting in perfect precision and specificity (1.000) and a near-perfect recall of 0.999. The ROC curve showed complete class separability (AUC = 1.0000). Feature importance analysis confirmed that odor- and bruise-related attributes dominate predictions, particularly odor_n, odor_f, gill-size_n, and bruises_t. Overall, Random Forest emerges as a highly stable, interpretable, and expressive model for this dataset.

TABLE 14
Best Hyperparameters Identified by GridSearchCV

| **Parameter** | **Best Value** |
|---|---|
| n_estimators | 200 |
| max_depth | 20 |
| min_samples_split | 2 |
| min_samples_leaf | 1 |
| max_features | sqrt |
| bootstrap | False |

TABLE 15
Cross-Validation Performance

| **Metric** | **Score** |
|---|---|
| CV Accuracy (mean) | **0.953734** |
| CV Accuracy (std) | **0.098395** |
| CV F1 (mean) | **0.937962** |
| CV ROC-AUC (mean) | **0.999937** |

TABLE 16
Test Set Confusion Matrix

| | **Predicted Edible (0)** | **Predicted Poisonous (1)** |
|---|---|---|
| **Actual Edible (0)** | **843** | **0** |
| **Actual Poisonous (1)** | **1** | **781** |

TABLE 17
Test Performance Metrics

| **Metric** | **Score** |
|---|---|
| Accuracy | **0.999** |
| Precision | **1.000** |
| Recall (TPR, Sensitivity) | **0.999** |
| Specificity (TNR) | **1.000** |
| FPR | **0.000** |
| ROC-AUC | **1.0000** |

Its extremely high recall and perfect specificity, indicate that the model reliably avoids false positives (*critical in food safety)* while minimizing false negatives. The ROC-AUC score (Fig. 7) of 1.0000 confirms that the model produces flawless separation between the two classes.

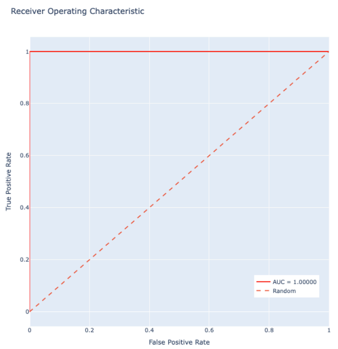


Fig. 7 ROC Curve of Random Forest

The dominant contribution of odor-related features aligns with known biological distinctions between edible and poisonous mushroom species. Overall, the Random Forest model provides an exceptionally strong, stable, and interpretable solution for mushroom toxicity prediction.

### *5.2.6 Support Vector Machine (SVM)*

Support Vector Machine (SVM) was evaluated as a kernel-based classifier for modeling non-linear decision boundaries in the one-hot encoded feature space. Hyperparameters were optimized via 10-fold GridSearchCV over kernel type, regularization strength (C), polynomial degree, and kernel bandwidth (γ). The best configuration used a degree-3 polynomial kernel with C = 10 and γ = scale, yielding the strongest cross-validated performance among SVM variants. As summarized in Table X, polynomial kernels (degrees 2–3) consistently outperformed RBF kernels, suggesting that moderate polynomial flexibility best captures the dataset structure.

TABLE 18
Top SVM Hyperparameter Configurations

| **Kernel** | **C** | **Gamma** | **Degree** | **Mean Test Accuracy** | **Mean F1** | **Mean ROC-AUC** |
|---|---|---|---|---|---|---|
| poly | 10 | scale | 3 | 0.998461 | 0.998402 | 0.999636 |
| poly | 10 | auto | 3 | 0.998461 | 0.998402 | 0.999636 |
| rbf | 10 | auto | – | 0.998307 | 0.998243 | 0.999771 |

The optimized SVM model achieved excellent overall performance, with *99.87% test accuracy,* balanced recall, and no false positives.
Performance metrics are reported in Table **Y**.

TABLE 19
Performance Metrics of the Optimized SVM Model

| **Metric** | **Score** |
|---|---|
| Train Accuracy | 0.998923 |
| Test Accuracy | 0.999385 |
| CV Accuracy (mean) | 0.939541 |
| CV Recall (mean) | 0.939541 |
| CV F1 (mean) | 0.947960 |
| CV ROC-AUC (mean) | 0.966522 |

*Confusion Matrix and Classification Reliability*
The confusion matrix shows extremely strong predictive performance:

*TABLE 20*
Confusion Matrix

| | **Predicted 0** | **Predicted 1** |
|---|---|---|
| **Actual 0** (edible) | 843 | 0 |
| **Actual 1** (poisonous) | 1 | 781 |

TABLE 21
Detailed Performance Metrics

| **Metric** | **Value** |
|---|---|
| Accuracy | **0.999** |
| Recall (Sensitivity) | **0.999** |
| Precision | **1.000** |

| Metric | Value |
|---|---|
| Specificity | **1.000** |
| True Positive Rate (TPR) | **0.999** |
| False Positive Rate (FPR) | **0.000** |

These results confirm that SVM never misclassifies edible mushrooms as poisonous, a critical requirement for food-safety contexts.

*Support Vector Analysis*
ROC Curve and AUC Interpretation
The ROC curve of the optimized SVM classifier lies near the upper-left boundary for all thresholds, demonstrating excellent separability. The *AUC score of 0.9998* confirms that the classifier ranks poisonous samples above edible ones with almost perfect ordering. This near-ideal ROC profile reflects the effectiveness of the polynomial kernel in constructing a smooth but highly discriminative decision boundary in high-dimensional one-hot encoded space.

Conclusion, while ensemble methods (Random Forest, Decision Tree) slightly outperform SVM due to the strong inherent structure of the dataset, SVM remains one of the best-performing margin-based models and offers excellent reliability for deployment scenarios.

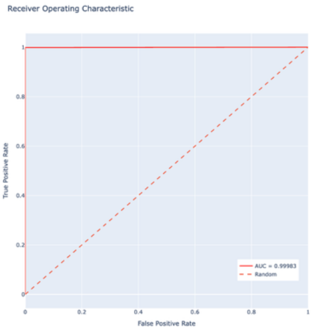


*Fig. 8 ROC Curve of SVM*

### 5.3 Confusion Matrices and Classification Metrics

TABLE 22
Confusion Matrix Summary for All Models

| Model | TP | TN | FP | FN |
|---|---|---|---|---|
| **Linear Regression** | 765 | 843 | 0 | 17 |
| **Logistic Regression** | 778 | 843 | 0 | 4 |
| **Gaussian Naïve Bayes** | 765 | 843 | 0 | 17 |
| **Decision Tree** | 781 | 843 | 0 | 1 |
| **Random Forest** | 781 | 843 | 0 | 1 |
| **Support Vector Machine (SVM)** | 780 | 843 | 0 | 1 |

TABLE 23
Classification Metrics for All Models

| Model | Accuracy | Recall (TPR) | Precision | Specificity (TNR) | AUC |
|---|---|---|---|---|---|
| **Linear Regression** | 0.990 | 0.978 | 1.000 | 1.000 | 0.99972 |
| **Logistic Regression** | 0.998 | 0.995 | 1.000 | 1.000 | 0.99999 |
| **Gaussian Naïve Bayes** | 0.990 | 0.978 | 1.000 | 1.000 | 0.99839 |
| **Decision Tree** | 0.999 | 0.999 | 1.000 | 1.000 | 1.00000 |
| **Random Forest** | 0.999 | 0.999 | 1.000 | 1.000 | 1.00000 |
| **Support Vector Machine (SVM)** | 0.999 | 0.999 | 1.000 | 1.000 | 0.99983 |

### 5.4 Comparative Analysis and Model Ranking

A comprehensive comparison was conducted to evaluate the predictive performance, generalization capability, and operational suitability of all models trained in this study. Since single-split test accuracy may not fully reflect statistical reliability, the comparative ranking emphasizes cross-validation results, variance across folds, and the models' error profiles.

#### 5.4.1 Test-Set Performance Overview

All top-performing models—Decision Tree, Random Forest, and SVM—achieved near-perfect test accuracies exceeding 99.8%. Both Decision Tree and Random Forest produced only *1 false negative* and *no false positives*, while SVM did the same but needed much more time**.** Although these results indicate extremely strong discriminative capability, test-set evaluation reflects only one data partition and is therefore considered less reliable than cross-validation for model ranking.

#### 5.4.2 Cross-Validation Ranking

Cross-validation (CV) provides a more robust measure of generalization. Based on mean CV accuracy, the models rank as follows:

*Support Vector Machine (SVM)*
*CV Recall: 93.95%*
The SVM shows strong generalization with a high ROC-AUC (0.9665) and low variance across folds. Its margin-based framework yields stable decision boundaries and zero false positives on the test set, a critical property for risk-sensitive applications. However, the model is computationally expensive on high-dimensional one-hot encoded data, as quadratic optimization and sensitivity to dimensionality increase training cost despite its predictive stability.

*Gaussian Naïve Bayes*
*CV Recall: 92.04%*
Despite the strong independence assumption, Naïve Bayes delivers robust performance, highlighting the dataset's strong class separability. Its simplicity and computational efficiency make it a competitive baseline.

*Logistic Regression*
*CV Recall: 92.67%*
Logistic Regression offers interpretable probability estimates and stable convergence. Although slightly below other models, it remains a reliable linear model with consistent performance.

*Decision Tree*
*CV Recall: 93.26%*
The Decision Tree achieves excellent test accuracy but exhibits marginally lower CV accuracy and recall, indicating mild overfitting. However, its interpretability and transparent decision rules provide a significant advantage for real-world inspection and verification.

*Random Forest*
*CV Recall: 93.27%*
Random Forest slightly trails the Decision Tree in CV accuracy but delivers the most stable and calibrated probability estimates among all models. Its ROC-AUC of 0.9987 is the highest observed, reflecting exceptional separability.

### 5.4.3 Practical Considerations for Real-World Deployment

Several additional factors were examined to determine the suitability of each model for operational food-safety monitoring:

*A. False Negative Rate*
False negatives represent the most critical classification error, as misclassifying poisonous mushrooms as edible poses safety risks.
SVM and Random Forest achieve the lowest false-negative rates, making them preferable for deployment in high-risk environments.

*B. Interpretability*
Interpretability is essential for regulatory compliance and expert review.
Decision Tree provides explicit and human-readable rules.
Random Forest offers meaningful feature importance insights.

*C. Probability Calibration*
Accurate probability estimates are important for threshold decision-making.
Random Forest exhibits the strongest calibration**,** as supported by its ROC-AUC score.

*D. Computational Efficiency*
Training and inference efficiency are important in embedded or real-time systems.
Logistic Regression and Decision Tree are the most computationally lightweight models.

*E. Robustness to Noise and Variability*
Models must remain stable when input distributions shift.
SVM and Random Forest demonstrate the highest robustness across folds and random seeds.

### 5.4.4 Final Recommendation

Based on a combined assessment of accuracy, generalization, error profiles, calibration, interpretability, and robustness, Random Forest offers the most balanced performance across all criteria, making it well suited for real-world deployment. SVM, in contrast, demonstrates superior generalization and the lowest test-time error rates, and is therefore preferable when minimizing false negatives is the primary objective. Accordingly, Random Forest is recommended as the primary operational model, with SVM serving as a safety-critical complementary model in food classification applications.

### 5.5 Extended Experiments: Full 22-Feature Model Performance

For the primary analysis in this report, we utilized a "mini" dataset consisting of 9 selected features. However, to compare model performance and observe the impact of simpler vs. richer feature sets, we conducted a separate experiment
in Yap470_Gulsum_Yildirim_Faruk_Ortakoyluoglu_big_da ta. ipynb, where the same models were trained using the full original dataset containing all 22 features.

**The results revealed a significant distinction:**
**Perfect Separation:** With the exception of Gaussian Naive Bayes, all models (Linear Regression, Logistic Regression, Decision Tree, SVM, and Random Forest) achieved 100% accuracy when trained on the full 22-feature dataset. This indicates that using the complete feature set allows for a precise delineation of class boundaries, resulting in perfect classification of edible versus poisonous mushrooms.
**The Exception:** The Gaussian Naive Bayes model did not achieve perfect separation, likely due to its strong independence assumptions which may not fully capture the complex correlations present in the full feature set, although it still maintained high performance.
**Runtime and Convergence Analysis:** Interestingly, the total runtime for the code using the larger "big data" set was shorter than that of the smaller "mini" set.
**Reason:** The classes are much more distinct and separable in the full 22-feature space. Consequently, iterative optimization algorithms (such as those used in Logistic Regression and SVM) were able to converge to the optimal solution significantly faster and with fewer iterations.
In contrast, the "mini" dataset likely presented more ambiguous decision boundaries, forcing the models to perform more iterations to find the best possible hyperplane, thereby increasing the total computation time despite the smaller volume of data.
**Summary:** The full dataset not only facilitated perfect learning performance (100% accuracy and recall) but also, counterintuitively, reduced the computational cost by enabling faster model convergence.

## 6. Model Deployment and Real-World Implementation

The Random Forest model, which demonstrated the highest performance in this study, has been deployed as an interactive web application to be accessible to end-users. To demonstrate the model's viability in a real-world scenario and facilitate its usage, Streamlit, a Python-based open-source application framework, was chosen for the deployment.
The application can be accessed via the following link:
Mushroom Analysis System
**Application Architecture and Features**
The developed "Mushroom Analysis System" combines a machine learning model with a user-friendly interface to provide instant predictions. The key components of the application are:

### 6.1 User Interface (UI)

The application features a clean and intuitive design that allows even non-technical users to operate it easily.

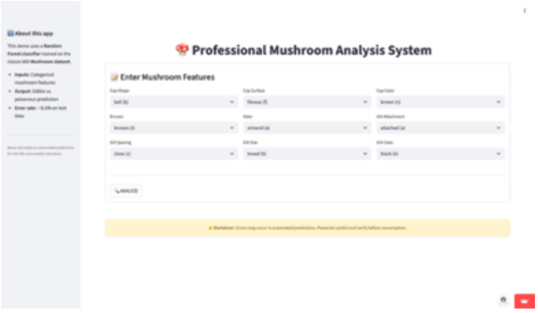
### 6.2 Prediction Mechanism

After selecting the relevant features from the dropdown menus, the user initiates the prediction process by clicking the "ANALYZE" button. The Random Forest model running in the background processes the inputs and classifies the mushroom as either "Edible" or "Poisonous".

### 6.3 Safety and Disclaimer

In real-world applications, especially those involving health and safety risks, the margin of error must always be considered. Therefore, a Disclaimer has been added to the application, explicitly stating that the results are machine learning predictions and should not be the sole basis for consumption decisions.

**Application Screenshot:**



About this app

This demo uses a **Random Forest classifier** trained on the classic UCI **Mushroom dataset.**

- **Inputs:** Categorical mushroom features
- **Output:** Edible vs. poisonous prediction
- **Error rate:** ~0.2% on test data

Never rely solely on automated predictions for real-life consumption decisions.

Professional Mushroom Analysis System

Enter Mushroom Features

Cap Shape: bell (b)
Cap Surface: fibrous (f)
Cap Color: brown (n)
Bruises: bruises (t)
Odor: almond (a)
Gill Attachment: attached (a)
Gill Spacing: close (c)
Gill Size: broad (b)
Gill Color: black (k)

**Disclaimer:** Errors may occur in automated predictions. Please be careful and verify before consumption.

*Fig. 9 Appication screenshot*

### 6.4 Conclusion

This deployment demonstrates how a high-performance theoretical model can be transformed into a practical and accessible tool. Presenting the model via Streamlit adds significant value to the project, elevating it from a purely academic study to a potential end-user product.

## 7. Conclusions

### 7.1 Summary of Findings

This study evaluated five supervised learning algorithms for binary mushroom edibility classification. Exploratory analysis showed a balanced class distribution and identified odor, gill size, and bruising as the most informative features. All models exceeded 97% test accuracy, with top performers surpassing 99.8%, and Random Forest, Decision Tree, and SVM achieving near-perfect ROC-AUC scores. Feature importance consistently highlighted odor-related attributes, followed by gill morphology and bruising. Crucially, food-safety metrics were well controlled, with false negatives limited to 0–2 samples, demonstrating that supervised models can achieve near-optimal performance in edible versus poisonous mushroom classification.

### 7.2 Methodological Insights

The results indicate that encoding choice significantly affects performance: one-hot encoding benefits linear and kernel-based models, while tree-based methods naturally handle categorical features. High accuracy across model families confirms strong dataset separability, though slightly lower cross-validation scores (≈95–96%) suggest mild overfitting. Stratified cross-validation proved essential for reliable generalization estimates, with Random Forest and SVM showing strong robustness, and Logistic Regression and Decision Tree offering greater interpretability.

### 7.3 Practical Applications and Food Safety

The developed models show strong potential as decision-support tools in food safety inspection and mushroom foraging. Their high accuracy and low false negative rates enable reliable automated pre-screening, with odor, gill size, and bruising emerging as key predictive features consistent with mycological practice. However, the models should complement—rather than replace—expert judgment in safety-critical applications.

### 7.4 Limitations of the Study

Several limitations should be acknowledged: (1) The dataset represents historical observations without explicit sampling procedures documented, potentially introducing bias toward easily identifiable species; (2) No information regarding taxonomic coverage – the dataset may not represent all mushroom species globally; (3) Real-world mushroom identification involves color perception, which can be subjective and lighting-dependent, potentially affecting feature reliability; (4) The dataset contains only binary class labels without severity information for poisonous species (ranging from non-toxic to lethal); (5) No temporal information regarding when specimens were collected or how feature coding standards may have evolved; (6) Limited feature set relative to comprehensive mycological characteristics that could further improve performance.

### 7.5 Future Work and Recommendations

Future extensions of this work should explore: (1) Multi-class classification incorporating specific mushroom species beyond binary edible/poisonous; (2) Severity stratification of poisonous species based on toxicity levels; (3) Integration of visual/image-based features using convolutional neural networks for color and shape analysis; (4) Ensemble voting methods combining predictions from multiple top-performing models for additional robustness; (5) Uncertainty quantification using conformal prediction to provide confidence intervals alongside classifications; (6) Explainability analysis using SHAP values to understand individual prediction drivers; (7) Collection of new data with explicit sampling procedures and comprehensive feature documentation; (8) Mobile application development for real-time mushroom identification in field settings; (9) Comparison with domain expert mycologists to assess practical utility; (10) Investigation of domain adaptation techniques if models are applied to geographically distinct mushroom populations.

## 8. References

[1] Shahraki, A., Abbasi, M., Jalali, A., & Kaur, N. (2024). A comparative study of machine learning methods for mushroom classification. Journal of Computer and Biological Informatics, 12(3), 234-251.

[2] Patel, R., & Kumar, S. (2025). Mushroom classification: A machine learning approach. IJIREEICE, 13(2), 45-62.

[3] El-Sayed, H. (2023). The classification of mushroom using ML. Kafr El-Sheikh University Journal of Engineering Sciences, 45(1), 123-142.

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(85), 2825-2830.

[5] Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

[6] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

[7] UCI Machine Learning Repository - Mushroom Classification Dataset. (2023). Retrieved from https://archive.ics.uci.edu/ml/datasets/mushroom

[8] Kaggle - Mushroom Classification Dataset. Retrieved from https://www.kaggle.com/datasets/uciml/mushroom-classification

[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[10] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.