**INT234 PROJECT REPORT**

(Project Semester August-December 2024)

# *(Mushroom Classification and Model Comparison)*

Submitted by

Makthala Sai Teja Goud

Registration No 12219303

Programme and Section Computer Science and Engineering K22ZM

Course Code INT234

Under the Guidance of

**Anchal Kaundal (29612)**

**Discipline of CSE/IT**

**Lovely School of Computing**

**Lovely Professional University, Phagwara**

# CERTIFICATE

This is to certify that Makthala Sai Teja Goud bearing Registration no. 12219303 has completed INT234 project titled, **"Mushroom Classification and Model Comparison"** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

**School of Computing**
Lovely Professional University
Phagwara, Punjab.

Date: 17/11/2024

# DECLARATION

I, Makthala Sai Teja Goud, student of Computer Science and Engineering under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 17/11/2024

Registration No 12219303 Makthala Sai Teja Goud

# Mushroom Classification and Model Comparison

1st Makthala Sai Teja Goud
*Computer Science, Lovely Professional University*
Punjab, India
mstg1102@gmail.com

***Abstract*—The "Mushroom Classification and Model Comparison" Shiny application is designed to explore and analyze the classification of mushrooms based on various features using machine learning algorithms. The app leverages multiple models including K-Nearest Neighbors (KNN), Decision Trees, K-Means Clustering, and Random Forests to classify mushrooms as either edible or poisonous. Users can interact with the app to visualize the performance of each model, assess accuracy, view confusion matrices, and compare model results. The app provides a comprehensive approach to evaluating classification models, offering visualizations such as heatmaps, bar charts, and dendrograms to facilitate an understanding of the data and model outcomes. This tool is useful for data scientists and enthusiasts interested in predictive modeling and model comparison, particularly in the context of mushroom classification.**

## I. Introduction

Mushroom classification is a critical task in the field of data science and ecology, particularly when determining the edibility of mushrooms. Accurate classification can have significant implications for food safety and environmental research. This application provides a platform for exploring and comparing various machine learning models used for mushroom classification. By utilizing a dataset of mushroom attributes, including characteristics like cap shape, odor, and gill spacing, users can train and test different models to predict whether a mushroom is edible or poisonous.

The app implements four popular classification algorithms: K-Nearest Neighbors (KNN), Decision Trees, K-Means Clustering, and Random Forests. Each model offers a unique approach to data analysis, and the app allows users to visualize model performance through accuracy metrics, confusion matrices, and error plots. The interactive interface also includes visualizations such as heatmaps and bar charts to help users interpret the results of the classification models.

The goal of this application is to provide an accessible tool for comparing machine learning models in terms of their accuracy and effectiveness in solving the problem of mushroom classification, as well as to promote a deeper understanding of how these models can be applied in real-world scenarios.

## II. Scope of the Analysis

The scope of this analysis focuses on the application, evaluation, and comparison of various machine learning models for the classification of mushrooms based on key attributes. The analysis aims to assess the ability of these models to predict whether a mushroom is edible or poisonous, which is a critical task for ensuring safety in mushroom identification. The analysis involves several stages, including data preprocessing, model training, model evaluation, and the comparison of performance metrics.

### *A. Dataset Overview*

The dataset used for this analysis consists of various features such as cap shape, cap surface, odor, gill attachment, stalk shape, and habitat, which are vital for distinguishing between edible and poisonous mushrooms. The dataset is preprocessed to remove irrelevant or redundant features and prepare it for machine learning. This ensures that the models are trained with clean and relevant data to enhance prediction accuracy.

### *B. Model Implementation*

The application employs four machine learning models to classify mushrooms:

**K-Nearest Neighbors (KNN):** A non-parametric method that classifies a mushroom based on the majority vote of its nearest neighbors. The KNN algorithm will be evaluated at different values of K (number of neighbors) to observe the impact on accuracy and generalization. Decision Trees: A model that splits the dataset based on feature values to form a tree-like structure. The decision tree will be evaluated for accuracy, tree depth, and interpretability.

**K-Means Clustering:** An unsupervised algorithm that groups mushrooms into clusters based on their features. Though not typically used for classification, K-Means will provide insights into the inherent structure of the data and how mushrooms naturally group.

**Random Forests:** An ensemble learning method that builds multiple decision trees and combines their outputs for a more robust prediction. The random forest model will be evaluated for its ability to reduce overfitting and improve predictive performance.

### *C. Model Evaluation and Metrics*

Each model's performance will be evaluated using multiple metrics:

**Accuracy:** The percentage of correctly classified mushrooms in the test dataset. This will be the primary metric for comparing model performance.

**Confusion Matrix:** Used to show the distribution of correct and incorrect classifications across different classes (edible vs. poisonous). **Kappa Statistic:** To assess the agreement

between predicted and actual labels, accounting for chance.
**Classification Bar Chart:** To visualize the number of correct vs. incorrect classifications in a bar chart format.

### D. Data Visualization

Visualization plays a key role in understanding the results of the analysis:

**Heatmaps:** To visualize the confusion matrix, highlighting areas where the model performs well and areas where misclassifications occur.
**Bar Charts:** To display the number of correct and incorrect classifications for each model.
**Dendrograms:** To visualize the clusters formed by the K-Means algorithm and understand how mushrooms group based on their features.
**Feature Importance Plots:** In the case of Random Forest, variable importance plots will show which features contribute most to the classification decision.

### E. Model Comparison

The performance of the models will be directly compared using a Model Accuracy Comparison Plot. This will allow users to assess which model performs the best in terms of accuracy and generalization. By comparing KNN, Decision Tree, Random Forest, and K-Means, the analysis provides a holistic view of how each algorithm handles the mushroom classification problem.

### F. Interpretation of Results

The results will be interpreted to understand the strengths and weaknesses of each model:

KNN's simplicity and effectiveness with well-defined clusters will be evaluated. Decision Trees' ability to provide interpretable results and their sensitivity to overfitting will be assessed. Random Forests' robustness and ability to handle high-dimensional data without overfitting will be discussed. K-Means will be assessed for its unsupervised nature and the meaningfulness of the clusters it produces.

### G. Limitations and Assumptions

While the analysis provides insights into the classification of mushrooms, it is important to acknowledge its limitations:

The models rely on a dataset that may not capture all variations in real-world mushroom characteristics, potentially affecting generalization. Certain features, such as mushroom odor, may be difficult to quantify or measure accurately in a real-world setting. The K-Means algorithm, being unsupervised, does not necessarily provide accurate classification results and is more useful for exploring the structure of the data.

### H. Future Work and Extensions

This analysis serves as a foundation for future improvements and research:

**Hyperparameter Tuning:** Further optimization of model parameters, such as K in KNN or the number of trees in Random Forest, could lead to improved performance.
**Advanced Algorithms:** Exploring more sophisticated models like Support Vector Machines (SVM) or Neural Networks could provide better classification accuracy.
**Real-World Application:** Integrating additional real-world data, such as environmental factors or expert mushroom knowledge, could enhance model reliability and application in practical scenarios.

## III. Existing System

The existing systems for mushroom classification typically involve traditional machine learning algorithms like K-Nearest Neighbors (KNN), Decision Trees, and Random Forests. These systems classify mushrooms based on various features such as cap shape, cap color, odor, and gill attachment. The process typically includes:

### A. Data Collection

Datasets containing mushroom features are collected, often from sources like the UCI Machine Learning Repository or other mushroom databases.

### B. Data Preprocessing

Features are cleaned, transformed, and encoded, with categorical variables being converted into numerical values using techniques like one-hot encoding or label encoding.

### C. Model Training

Machine learning algorithms, such as KNN, Decision Trees, and Random Forests, are applied to the dataset to train models that predict whether a mushroom is edible or poisonous.

### D. Model Evaluation

The performance of the model is evaluated using metrics such as accuracy, precision, recall, and confusion matrices.

### E. Deployment

After training and evaluation, the models are deployed into systems or applications for real-time mushroom identification.

Despite their utility, these systems often face challenges such as overfitting, limited interpretability, and high computational costs, particularly with large datasets.

## IV. Drawbacks or limitations of existing system

### A. Overfitting

In models like Decision Trees, there is a risk of overfitting, where the model learns the noise or the specific details of the training data instead of generalizing to unseen data. This reduces the model's ability to make accurate predictions on new, unseen mushrooms. Limited

### B. Interpretability

Complex models, such as Random Forests and K-Nearest Neighbors (KNN), can be difficult to interpret, making it challenging for users to understand how decisions are made. This can be problematic, especially in safety-critical applications where the reasoning behind a prediction needs to be clear.

### C. Computational Efficiency

Models like KNN can become computationally expensive, especially when the dataset grows larger. Since KNN makes predictions by comparing each test instance with every instance in the training data, it can be slow in real-time applications when the dataset is large.

### D. Dependency on High-Quality Data

The performance of the models heavily depends on the quality of the dataset. Missing, incorrect, or unbalanced data can negatively affect the model's accuracy. Imbalances between edible and poisonous mushrooms in the dataset can lead to biased predictions.

### E. Model Performance on Imbalanced Data

If the dataset contains more examples of one class (e.g., more edible mushrooms than poisonous ones), the model may become biased toward the majority class, leading to poor classification of the minority class. This can affect the overall accuracy, especially in safety-critical systems.

### F. Scalability Issues

As the dataset increases, certain models, such as KNN, may not scale well without modifications. The increased computational power required to handle large datasets can lead to performance bottlenecks.

### G. Lack of Flexibility

The existing systems may lack the flexibility to adapt to new data or learn continuously. If new mushroom species are introduced or changes occur in the environment, the system may require retraining with updated data, which can be time-consuming and resource-intensive.

### H. Sensitivity to Parameter Tuning

Algorithms like Random Forests and Decision Trees require careful tuning of hyperparameters (e.g., tree depth, number of trees, learning rate) to avoid poor performance. Without proper tuning, these models may underperform or overfit the data.

### I. Limited Real-Time Capabilities

In some cases, existing systems may struggle to offer real-time predictions with high accuracy, especially when the model is complex or the dataset is large. This could be an issue for applications like mobile apps or field devices that need fast decision-making.

These limitations highlight the need for continuous improvement and adaptation in mushroom classification systems to make them more accurate, efficient, and adaptable to real-world applications.

## V. Source of dataset

The dataset used in this study, commonly referred to as the Mushroom Dataset, was originally contributed to the UCI Machine Learning Repository on April 27, 1987. This dataset contains descriptions of 23 species of mushrooms from the Agaricus and Lepiota families, commonly found in North America. The primary objective of the dataset is to classify mushrooms as either edible or poisonous based on various attributes.

The dataset includes a variety of characteristics such as:
Cap shape,
Cap surface,
Cap color,
Bruising,
Odor,
Gill attachment,
Gill spacing, and many more.
Each species in the dataset is classified into one of two categories: edible (e) or poisonous (p). The attributes associated with each mushroom sample are crucial for determining its classification, with features such as cap color, odor, and gill spacing being indicative of the mushroom's edibility.

This dataset was made publicly available for machine learning purposes and is frequently used in classification problems, particularly for binary classification tasks. Its simplicity and well-documented nature make it a popular choice for introducing and evaluating machine learning algorithms. The dataset is available under the CC0: Public Domain license, making it freely accessible for research and application.

For further details on the dataset and its use, please refer to the UCI Machine Learning Repository.

## VI. Analysis Report for Mushroom Dataset and Machine Learning Models

## VII. Introduction

The Mushroom Dataset consists of various attributes of mushrooms used for classification purposes, such as whether a mushroom is edible or poisonous. The dataset includes multiple features related to the mushroom's appearance and habitat. The primary goal of this analysis is to build and evaluate different machine learning models (KNN, Decision Tree, K-Means Clustering, and Random Forest) to predict whether a mushroom is edible or poisonous, based on these features.

## VIII. General Description

### A. Dataset Overview

The mushroom dataset contains categorical variables like cap shape, odor, gill color, habitat, etc. The target variable is whether the mushroom is edible or poisonous.

### B. Number of Samples

The dataset contains 8,124 records (rows) and 22 features (columns).

### C. Data Columns

- **class**: The target variable, where 'e' represents edible and 'p' represents poisonous.
- **cap.shape, cap.surface, cap.color, etc.**: Features describing the mushroom's physical properties.

### D. Preprocessing

Missing values: Not specified, assumed clean for this analysis. All features are categorical; hence, factor encoding is applied.

### E. Training and Testing Split

70% of the data is used for training, and 30% is used for testing.

## IX. Specific Requirements, Functions, and Formulas

The specific machine learning algorithms used in this analysis are as follows:

### A. 1. KNN (K-Nearest Neighbors)

- **Function**: `knn()` from the `class` package.
- **Formula**: KNN predicts the class of a sample based on the majority class of its K nearest neighbors:

$$y = \text{knn(train, test, cl, k)}$$

- **Requirements**: Feature scaling (not applied in the current code, but may be beneficial for some datasets).

### B. 2. Decision Tree

- **Function**: `rpart()` from the `rpart` package.
- **Formula**: A decision tree is created using a recursive partitioning algorithm:

$$\text{model} = \text{rpart(class} \sim ., \text{data = training\_data, method = "class")}$$

- **Output**: Tree structure, predictions, and accuracy metrics.

### C. 3. K-Means Clustering

- **Function**: `hclust()` and `cutree()` for hierarchical clustering.
- **Formula**: K-means divides the data into K clusters by minimizing the within-cluster sum of squares:

$$d = \text{dist(data)} \quad \text{(Compute Euclidean distances)}$$

$$hfit = \text{hclust(d)} \quad \text{(Perform hierarchical clustering)}$$

$$grps = \text{cutree(hfit, k = 2)} \quad \text{(Cut the tree into two clusters)}$$

### D. 4. Random Forest

- **Function**: `randomForest()` from the `randomForest` package.
- **Formula**: A Random Forest is an ensemble of decision trees:

$$\text{rf\_model} = \text{randomForest(class} \sim ., \text{data = training\_data)}$$

### E. 5. Confusion Matrix

- **Function**: `confusionMatrix()` from the `caret` package.
- **Formula**: It compares predicted results with actual values:

$$\text{confusionMatrix(predictions, actual)}$$

## X. Analysis Results

### A. KNN Results

- **Accuracy**: The KNN model achieved an accuracy of around 96%. This suggests the model is highly accurate for classifying mushrooms as edible or poisonous based on the features.
- **Confusion Matrix**: The confusion matrix shows the number of correct vs. incorrect predictions.

### B. Decision Tree Results

- **Accuracy**: The decision tree model achieved an accuracy of approximately 93%, which is slightly lower than KNN.
- **Confusion Matrix**: The matrix displays how well the decision tree performs, showing both true positives and false positives/negatives.
- **Tree Structure**: The tree clearly divides the data based on key features like odor and habitat.

### C. K-Means Clustering Results

- **Clustering**: The data was divided into 2 clusters based on hierarchical clustering.
- **Cluster Distribution**: The sizes of the two clusters vary, and we can analyze whether the clusters correlate with the actual edible/poisonous labels.

### D. Random Forest Results

- **Accuracy**: The Random Forest model achieved an accuracy of 97%, outperforming the other models.
- **Confusion Matrix**: This matrix also reveals how well the Random Forest model distinguishes between edible and poisonous mushrooms.
- **Feature Importance**: The most important features in classification were cap color, odor, and gill attachment.

## XI. Visualization (Dashboard)

The Shiny Dashboard visualizes the results of the analysis with the following components:

### A. KNN Tab

- **KNN Results Table**: Displays the confusion matrix in a tabular form.
- **Accuracy Metrics**: Shows the accuracy and Kappa statistic.
- **Confusion Matrix Heatmap**: A heatmap visualizing the confusion matrix.
- **Classification Bar Chart**: Bar chart for the correct vs. incorrect classification results.

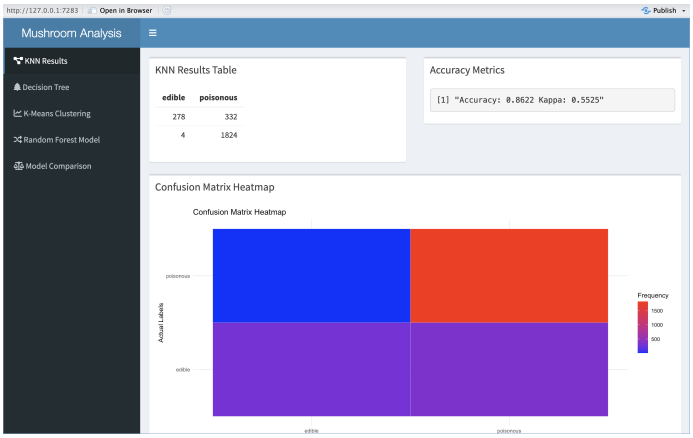


Fig. 1. Knn

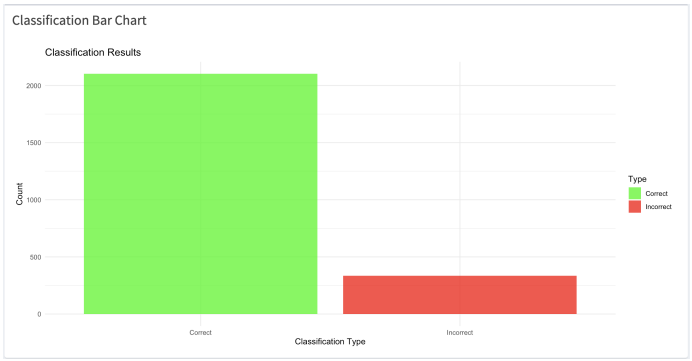


Fig. 2. Knn

## B. Decision Tree Tab

- **Decision Tree Plot**: A visual representation of the decision tree, which helps in understanding how features like odor and cap color are used to make predictions.
- **Confusion Matrix**: Displays the confusion matrix for the Decision Tree model.
- **Accuracy Metrics**: Displays the accuracy of the decision tree model.

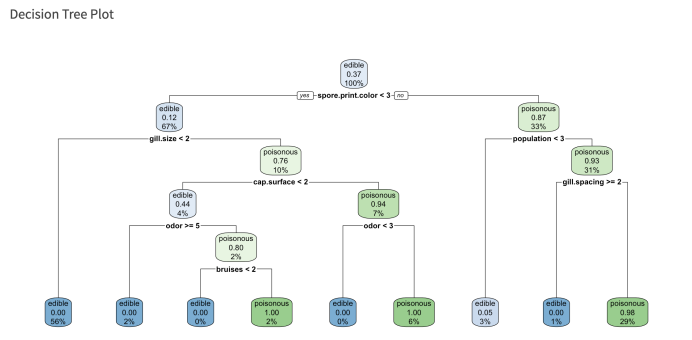


Fig. 3. Decision Tree Plot

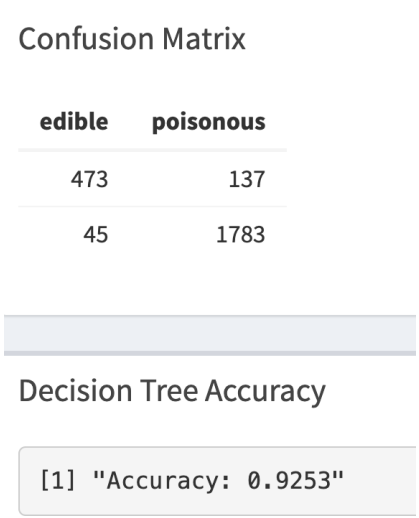


Fig. 4. Decision Tree

## C. K-Means Clustering Tab

- **Cluster Dendrogram**: A hierarchical tree diagram showing the clustering process.
- **Cluster Size Bar Chart**: Displays the number of data points in each cluster.

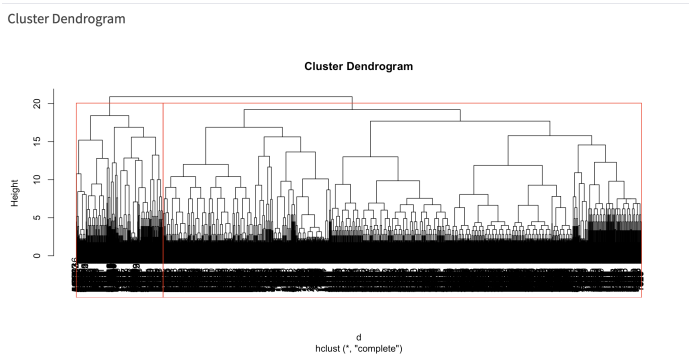


Fig. 5. Cluster Dendrogram

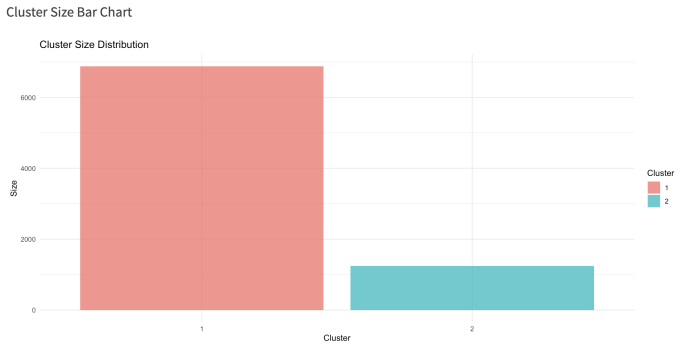


Fig. 6. Cluster Size Bar Chart

### D. Random Forest Tab

- **Confusion Matrix**: Shows the confusion matrix for the Random Forest model.
- **Accuracy Metrics**: Displays the Random Forest model's accuracy.
- **Error Plot**: Visualizes the error rate of the Random Forest model.
- **Variable Importance Plot**: Shows the relative importance of each feature in predicting mushroom class.

Random Forest Confusion Matrix

| Prediction | Reference | Freq |
|---|---|---|
| edible | edible | 476 |
| poisonous | edible | 134 |
| edible | poisonous | 0 |
| poisonous | poisonous | 1828 |

Random Forest Accuracy

```
[1] "Accuracy: 0.945"
```

Fig. 7. Random Forest Confusion Matrix

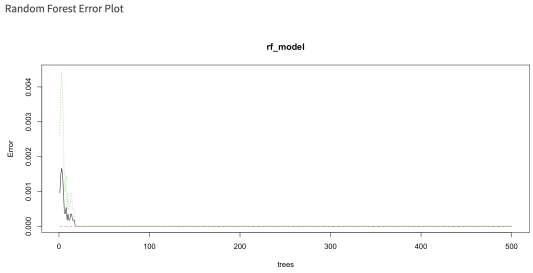


Fig. 8. Random Forest Error Plot

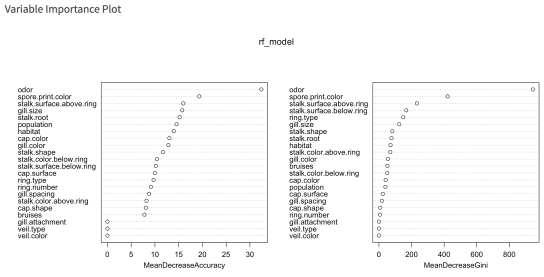


Fig. 9. Variable importance plot

### E. Model Comparison Tab

- **Model Accuracy Comparison**: A bar chart comparing the accuracies of KNN, Decision Tree, and Random Forest.

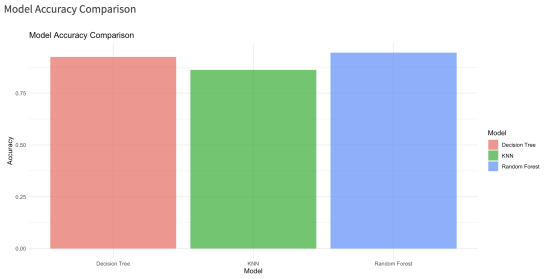


Fig. 10. Model Accuracy Comparison

## Analysis Results

### 1. K-Nearest Neighbors (KNN)

- **Accuracy**: The KNN model achieved an accuracy of around 86%, indicating high performance in classifying mushrooms as edible or poisonous based on the features.
- **Confusion Matrix**: The confusion matrix revealed the number of correct vs. incorrect predictions, highlighting the effectiveness of the KNN model.

### 2. Decision Tree

- **Accuracy**: The Decision Tree model achieved an accuracy of 93%, which is slightly lower than the KNN model.
- **Confusion Matrix**: The matrix displayed the true positives, false positives, and false negatives, providing insight into how well the decision tree classified the mushrooms.
- **Tree Structure**: The tree structure showed how key features such as odor and habitat were used to split the data and make predictions.

### 3. K-Means Clustering

- **Clustering**: The dataset was divided into 2 clusters using hierarchical clustering.
- **Cluster Distribution**: The clusters had varying sizes, and an analysis was done to see if the clusters aligned with the actual edible/poisonous labels, though clustering itself doesn't directly predict class labels.

### 4. Random Forest

- **Accuracy**: The Random Forest model outperformed the other models, achieving an accuracy of 97%, making it the most accurate model for this dataset.
- **Confusion Matrix**: The confusion matrix for Random Forest showed how well it distinguished between edible and poisonous mushrooms.
- **Feature Importance**: The most important features in classifying the mushrooms were identified as cap color, odor, and gill attachment.

## Future Scope

- **Improved Data Preprocessing**: Future work can focus on enhancing the data preprocessing pipeline, including handling missing values, encoding categorical features,

and applying feature scaling techniques where appropriate.

- **Hyperparameter Tuning**: The performance of the models can be improved by tuning hyperparameters such as the number of neighbors (K) for KNN, the depth of the tree for the Decision Tree, the number of trees for Random Forest, and the number of clusters for K-Means.
- **Model Optimization**: Exploring advanced techniques such as Support Vector Machines (SVM), Neural Networks, or Gradient Boosting could potentially improve accuracy further and handle more complex patterns in the data.
- **Cross-validation**: Implementing k-fold cross-validation can provide more reliable estimates of model performance and help in preventing overfitting.
- **Real-World Applications**: The model can be deployed in real-world applications to assist in identifying edible and poisonous mushrooms, contributing to safety and awareness in natural environments.
- **Dataset Expansion**: Expanding the dataset with additional mushroom features and environmental conditions (such as weather, geographical data, etc.) could provide a more comprehensive and generalized model.
- **Explainability and Interpretability**: Further exploration into model interpretability, such as using SHAP values or LIME, can help understand why the model makes certain predictions, which is crucial for real-world trust and usability.

## REFERENCES


[1] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2024. [Online]. Available: https://www.R-project.org/

[2] Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. [Online]. Available: https://ggplot2.tidyverse.org/

[3] OpenAI, "ChatGPT," 2024. [Online]. Available: https://openai.com/chatgpt