King Saud University
College of Computer and Information Sciences
Department of Information Technology

IT326: Data Mining
1st Semester 1447 H

Mushroom Edibility Mining

Phase # 3

Section #	NAME	ID
56546	*Rana Almutairi*	*445200294*
	Haifa Bin Salmah	*445202292*
	Lama Alghamdi	*445200358*
	Aliyah Alharbi	*445201302*

Supervised By: Dr. *Lama Alsudais*

Problem:

The goal of this project is to apply data mining techniques—specifically classification and clustering—to a Mushroom dataset in order to distinguish edible mushrooms from poisonous ones and to explore meaningful grouping patterns within the data. This problem is important because correct identification directly impacts safety; misclassification can lead to severe health risks. By analyzing mushroom attributes, the project aims to build predictive and descriptive models to uncover insights that support reliable and interpretable decision-making.

Data Mining Task:

Classification Task

The classification task focuses on predicting whether a mushroom is edible or poisonous based on its characteristics. This frames the problem as a supervised learning task, where the goal is to learn patterns from labeled examples and use them to classify new, unseen mushrooms.

Clustering Task

The clustering task aims to group mushrooms into meaningful clusters based on their features without using the class label. This turns the problem into an unsupervised learning task, where the objective is to discover natural structure and similarities among mushroom samples.

Data:

The dataset used is the Mushroom Classification Dataset (UCI/Kaggle), provided in CSV format. It contains 8124 mushroom samples described by 23 categorical attributes**,** covering characteristics such as cap shape, odor, gill size, stalk color, habitat, and more. The target variable is class, which has two values:

- **e** = edible (4208 samples)
- **p** = poisonous (3916 samples)

The dataset is fairly balanced and fully categorical, making it suitable for both classification and clustering. Raw inspection showed that every column is of type "object," and the dataset includes no numerical features prior to preprocessing.

Data preprocessing:

Several preprocessing steps were applied to prepare the dataset for data mining:

1) Handling Missing Values

The attribute *stalk-root* contained missing entries represented with "?". These values were replaced with the most frequent category (mode) to keep the dataset complete and consistent.

2) Treating Outliers

Each categorical feature was examined for rare values appearing in less than 1% of the data. Instead of removing them, all outlier categories were replaced with a unified label ("Other") to reduce noise and improve model stability.

3) Encoding Categorical Features

Because all attributes were categorical, the following encoding steps were applied:

- **Label Encoding** for the target variable class
- **One-Hot Encoding** for all other attributes
  This expanded the dataset to approximately 103 binary feature columns.

4) Feature Selection

A Chi-Square test was used to identify statistically significant features related to the class label. After that, a correlation analysis was performed to remove highly correlated one-hot encoded features (correlation > 0.75), reducing redundancy.

Data Mining Technique:

Classification Technique (Decision Tree – Gini & Entropy):

we applied Decision Tree classification as a supervised learning technique to predict whether each mushroom is edible or poisonous based on its encoded features. Two splitting criteria were used to examine how the model behaves under different measures:

- **Gini Index**
- **Entropy (Information Gain)**

To evaluate the model under different data conditions, we tested three train–test split ratios:

- **60/40**
- **70/30**
- **80/20**

Stratification (**stratify=y**) was used in each split to maintain the original class balance, and the tree depth was limited using **max_depth = 3** to keep the model simple and prevent overfitting.

For every split and every splitting criterion, we trained a Decision Tree model and generated the following:

- The **accuracy score**, showing how many samples were classified correctly
- The **confusion matrix**, showing correct vs. incorrect predictions
- A **Decision Tree plot**, visualizing how the model makes decisions based on the input features

This setup allows us to compare how different splitting criteria and different train–test partitions affect the structure, behavior, and performance of the classification model.

Clustering Technique (K-means)

In this phase, we applied **K-means clustering** as an unsupervised learning technique to group mushrooms based on similarities in their encoded characteristics. Because clustering does not use class labels, we removed the target variable and worked only with the feature set.
We evaluated the model using three different values of K, selected based on the outcomes of the Elbow and Silhouette analyses:

- **K = 8**
- **K = 9**
- **K = 10**

For each K value, we trained a K-means model and generated:

• Cluster assignments for all samples
• The **Silhouette score**, which measures cluster separation and cohesion
• The **Within-Cluster Sum of Squares (WCSS)**, which measures cluster compactness
• **PCA visualizations**, showing how clusters appear when projected into 2D space

This process helps us understand how different numbers of clusters affect the structure, separation, and compactness of the dataset. Comparing K = 8, K = 9, and K = 10 allows us to determine which configuration produces the most meaningful clustering pattern.

Python packages

For the classification task, the project relies on several key components from scikit-learn. The train–test splits are created using ***train_test_split*** from ***sklearn.model_selection***, while the decision-tree models (Gini and Entropy) are built using ***DecisionTreeClassifier*** from ***sklearn.tree***. Model performance is measured with ***accuracy_score*** and ***confusion_matrix*** from ***sklearn.metrics***, and the tree structure is visualized using ***plot_tree***, also from ***sklearn.tree***.

For the clustering task, **KMeans** from *sklearn.cluster* was used to form groups of mushrooms based on feature similarity. Cluster quality was evaluated using both **silhouette_score** and **silhouette_samples** from *sklearn.metrics* to measure separation and cohesion. The features were standardized using **StandardScaler** from *sklearn.preprocessing* before applying K-means to ensure fair distance calculations.
To visualize the clusters, dimensionality was reduced to two components using **PCA** from *sklearn.decomposition*, and **ConvexHull** from *scipy.spatial* was applied to outline the boundaries of each cluster in the 2D space for clearer interpretation.

Evaluation and Comparison:

Classification (Gini vs. Entropy)

Decision Tree classification was evaluated using three train–test splits (60/40, 70/30, 80/20) and two attribute selection measures (Gini index and Entropy). For each combination, accuracy and the confusion matrix were reported, and the corresponding tree was visualized.
Across all splits, the Gini-based tree consistently achieved higher accuracy than the Entropy-based tree. Approximate test accuracies were:

Gini:

	80 % training set 20% testing set:	70 % training set 30% testing set:	60 % training set 40% testing set:
Accuracy	**97.97%**	**98.07%**	**98.15%**

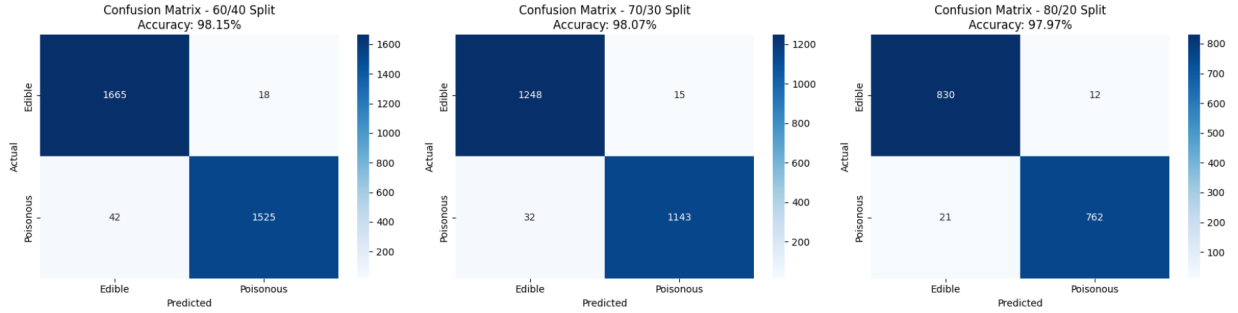
Entropy:

	80 % training set 20% testing set:	70 % training set 30% testing set:	60 % training set 40% testing set:
Accuracy	**96.68%**	**96.92%**	**96.92%**

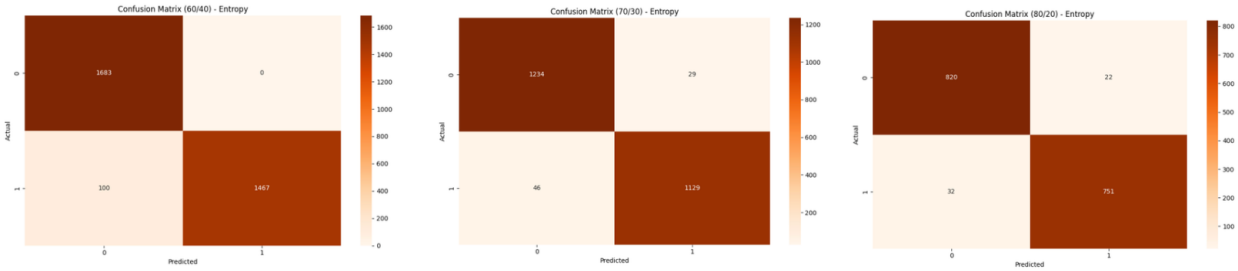
Interpretation:

- Both criteria produced very few misclassifications across all splits.
- **Gini** consistently showed **slightly fewer errors** than **Entropy** for the same split.
- **Gini** performs better because it creates **simpler and more stable splits**, while **Entropy** is slightly **more sensitive** to small variations in the data.
- The **60/40 split** achieved the best performance overall because it provides a strong amount of training data while still keeping a large testing set.
- Overall, **Gini is the best-performing attribute selection measure** across all partitions.
- **Entropy** is still competitive but remains **consistently lower** in accuracy compared to Gini.

confusion matrix for Gini :



confusion matrix for Entropy :



Clustering (K-Means: K = 8, 9, 10)

We computed the Silhouette score and WSS for each value of K.
The results obtained from our experiment are summarized below:

Metric	K = 8	K = 9	K = 10
Average Silhouette Width	0.1783	0.1891	0.1623
WSS	480,611.0	393,866.2	386,373.5

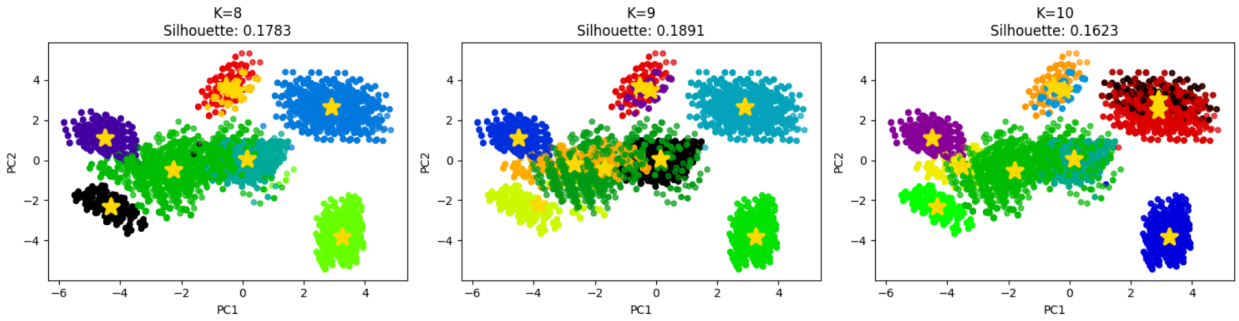
Interpretation:

• Silhouette Score:
Higher values indicate better cluster separation.
K = 9 achieved the highest Silhouette score (0.1891), meaning it provides the clearest and most well-separated clusters.

• WSS:
Lower WSS indicates more compact clusters.
Although K = 10 has the lowest WSS, its lower Silhouette score suggests over-clustering, where adding more clusters weakens overall cluster quality.
K = 8 performs reasonably well but does not match the separation quality of K = 9.

• Overall Insight:
When considering both metrics together, K = 9 offers the strongest balance between compactness and separation, and PCA visualizations also shows clearer cluster boundaries at K = 9 compared to K = 8 and K = 10.

Clustering Visualizations:

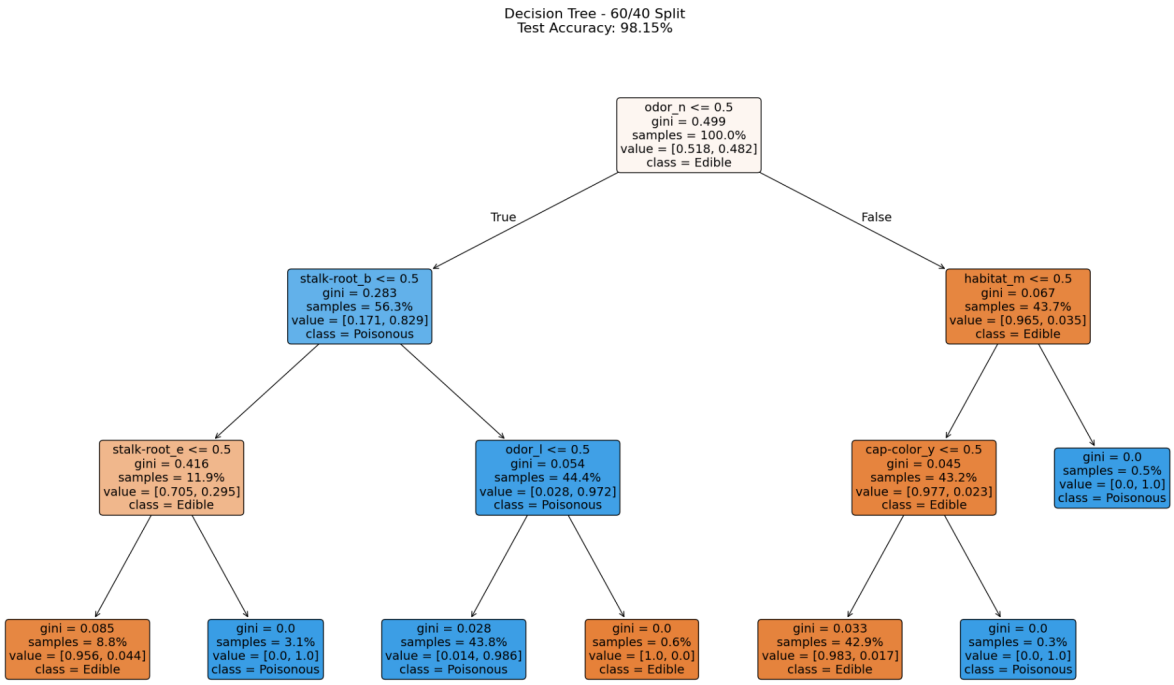


Findings and Discussion:

Findings and Discussion Related to Classification:

The classification results showed that both Decision Tree models—Gini and Entropy—performed consistently well across all train–test splits. Accuracy remained above 96% in all cases, with **Gini slightly outperforming Entropy** due to producing simpler and more stable splits. The confusion matrices confirmed that misclassifications were very limited, demonstrating that the encoded mushroom features are highly separable. These findings align with the research paper's conclusion that tree-based models achieve strong performance on the mushroom dataset and are effective in classification tasks involving fully categorical attributes.

The final Decision Tree model achieved an **excellent** Test Accuracy of **98.15%.** The tree below visualizes the logic provided, revealing the most critical features for distinguishing between edible and poisonous mushrooms.



Key Findings

- **Primary Rule:** The most significant feature is **odor**. The tree shows that if a mushroom has **no odor(n)** and certain stalk-root conditions, it is **overwhelmingly likely to be poisonous** (98.6%). Conversely, other odor types (like foul f) lead to edible classifications.

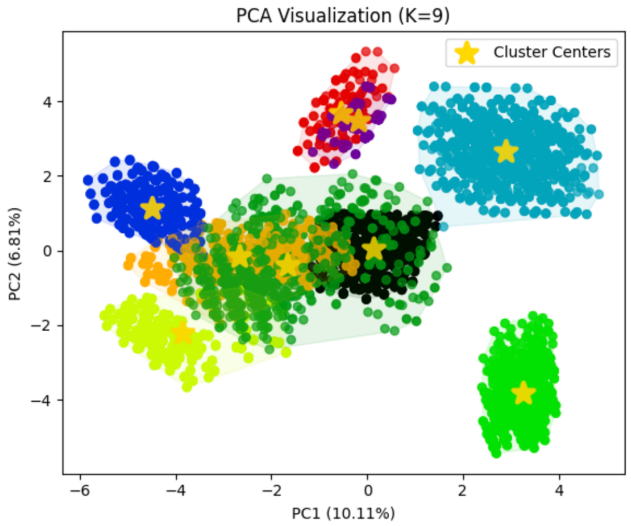
- **Secondary Confirming Features:** The stalk-root feature provides the **initial split**, but odor immediately refines the decision. Further down the tree, features like habitat and cap-color are used to confirm the classification or handle edge cases.

- **Model Interpretability:** The tree is highly interpretable. For example, a simple rule derived from the tree could be: "If the mushroom has no odor (n) and its stalk root is not bulbous (b), it is poisonous." This aligns with known mycological knowledge, where **certain odors** are **strong indicators of toxicity**.

Findings and Discussion Related to Clustering:

For clustering, evaluating **K = 8, 9, and 10** revealed clear differences in cluster structure and quality. **K = 9 produced the highest Silhouette score (0.1891)**, indicating the strongest separation between clusters, while the PCA visualizations showed that clusters at K = 9 were more distinct and balanced than those at K = 8 or K = 10. Although K = 10 had slightly lower WSS, its reduced Silhouette score suggested over-clustering. These results demonstrate that the dataset contains meaningful internal groupings that can be captured effectively when K is chosen appropriately.

The following plot visualizes the final k=9 clusters:



Interpreting the Clusters:

- **Cluster Separation:** The clusters are reasonably well-separated in the 2D PCA space, indicating that the clustering algorithm found distinct groups within the data. This suggests there are several "types" of mushrooms based on their characteristics.

- **Linking Clustering to Classification:** We can infer that these clusters likely correspond to groups of mushrooms with similar physical traits. By examining the original features of the mushrooms in each cluster (e.g., odor, cap-color, habitat), we would likely find that certain clusters are predominantly edible or poisonous. For instance, the cluster containing the mushrooms with odor_f (foul) would almost certainly be a distinct, poisonous cluster.

Conclusion:

Comparing the two data-mining techniques, classification provides the most direct and accurate solution to the edible/poisonous prediction task, achieving high performance with minimal error. Clustering, on the other hand, offers exploratory insights by revealing the natural structure of the dataset but does not replace supervised prediction. Together, the results confirm that the mushroom dataset is highly informative, easy to separate using decision rules, and capable of forming coherent clusters in an unsupervised setting.

Overall, the **Decision Tree with Gini** emerges as the strongest classification model due to its slightly higher accuracy and more stable splits, while **K = 9** delivers the most meaningful clustering structure based on silhouette and PCA analyses. These findings align with the referenced research paper, which emphasizes the strong separability of mushroom attributes and the effectiveness of classification techniques on this dataset.

Although the paper does not explore clustering, the attribute patterns it highlights help explain why our unsupervised clustering results—particularly at K = 9—form clear and consistent groupings.

References:

[1] R. N. S. A. Sallam, E. A. Shehab, and S. A. Shehab, "The classification of mushroom using ML," *Kafrelsheikh Journal of Information Sciences*, vol. 4, no. 2, pp. 1–7, Nov. 2023.

[2] Kaggle, "Mushroom Classification Dataset."
Available: https://www.kaggle.com/uciml/mushroom-classification