IML PROJECT (GROUP.NO-46)

Yashaswini T (B23CI1039)
Aakarshana A (B23ME1002)
Niveditha V (B23MT1044)
Brundha Devi A (B23CH1015)

MUSHROOM CLASSIFICATION REPORT

Introduction:

This project aims to classify mushrooms as either edible or poisonous using various characteristics of the mushrooms. The dataset consists of several categorical features that describe these characteristics, which are transformed into numerical formats suitable for analysis and modeling.

Dataset Overview:

The dataset was loaded successfully from a CSV file and contains 8124 entries with 23 attributes. The key columns include the target variable class (indicating edibility) and various features such as cap-shape, cap-surface, cap-color, bruises, and more.

Data Preprocessing:

Missing Values

The dataset has been thoroughly checked for missing values, and it was found that all features are complete with **0 missing values** across all columns as well as there are **0 duplicate entries**.

Encoding Categorical Variables

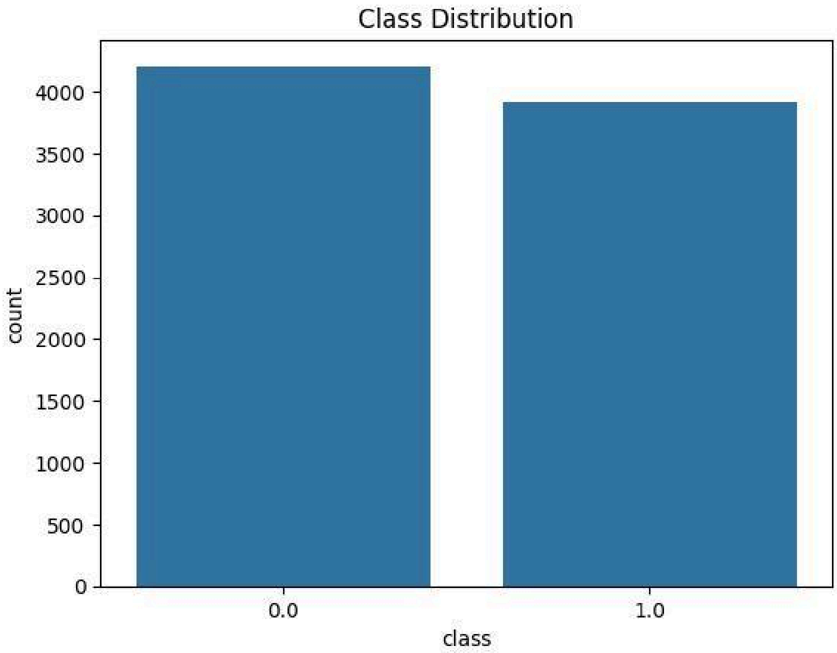
The dataset contains multiple categorical features, including the target variable. We used Ordinal Encoding, which converts each category into a unique numeric value. This encoding allows models to process the data effectively without implying any order among categories.

Class Distribution Analysis

The bar chart represents the distribution of the target variable (class) in the dataset. The target variable has two classes:

Class 0.0: Likely represents one type (e.g., edible mushrooms).

Class 1.0: Likely represents the other type (e.g., poisonous mushrooms).



Exploratory Data Analysis (EDA)

Feature Distribution:

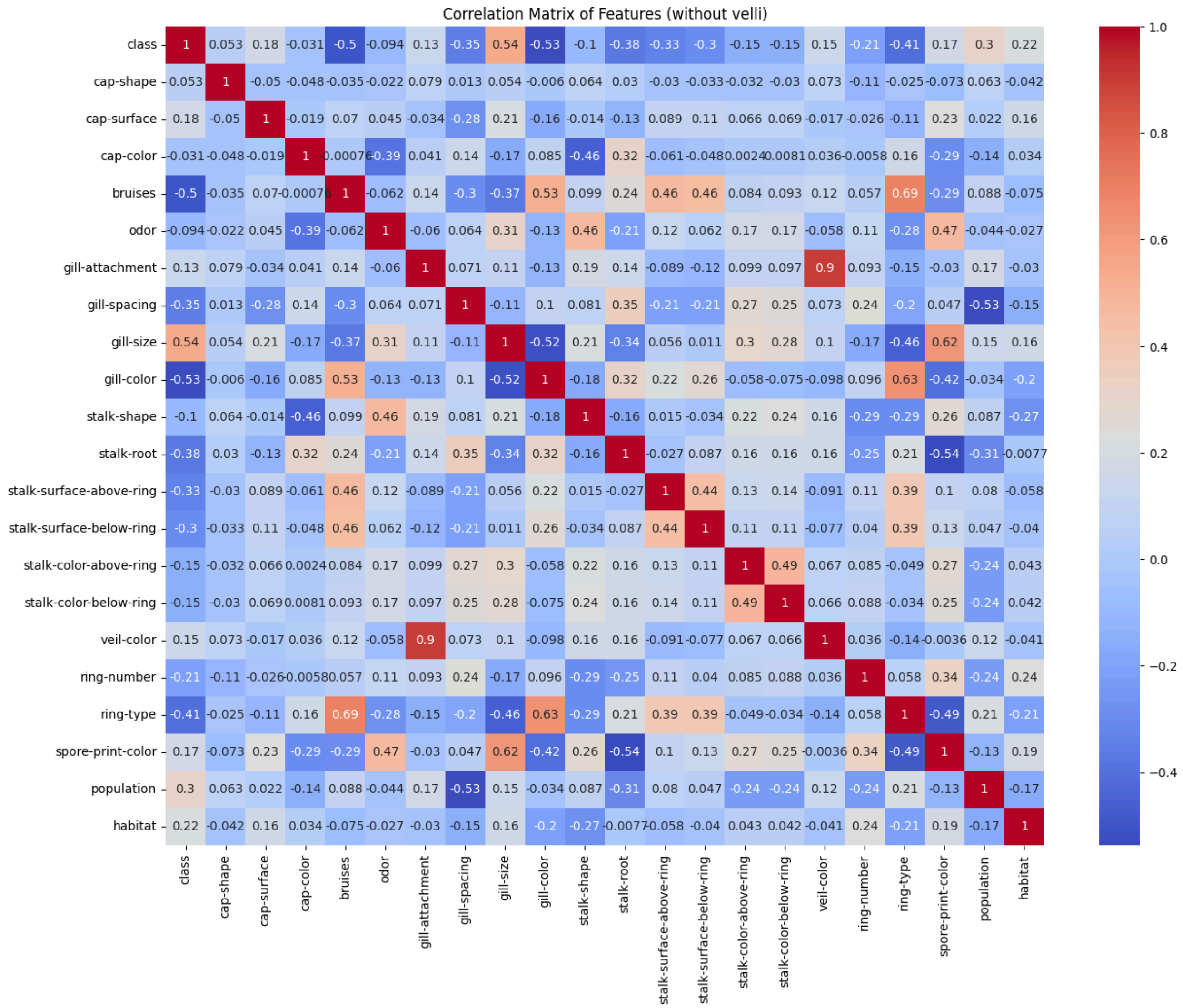
The numeric columns in the dataset were visualized using histograms to understand the distribution of each feature:

- Some features, such as gill-size and ring-type, show a bimodal distribution, indicating that mushrooms tend to have certain dominant categories. Other features, such as veil-type and stalk-color, are more uniformly distributed, suggesting a wider range of variation across mushroom species.

Correlation Analysis:

A heat map of the correlation matrix for numeric features was generated to find a relationships between variables:

- Features such as gill-attachment and veil-color have a high positive correlation, as expected due to their similarity in representing color characteristics.
- Most other features show low or near-zero correlations, suggesting that each feature may independently contribute to classification.



Data Splitting:

- Features (X): All columns except the target variable (class) were used as features, totaling 22 features.
- Target (y): The first column (class) was designated as the target variable.

We used 70% of the data for training set and 30% into the test set with a random state of 42.

Dimensionality Reduction Using PCA (with Outliers):

PCA Implementation

- **Number of Components**: We adjusted the number of components to **N** =9

Interpretation of Results

- The explained variance ratios indicate how much variance is captured by the chosen components. Collectively, these components explained about **80.09%** of the total variance.

Outlier Removal:

- We utilized the Interquartile Range (IQR) method to identify and remove outliers from the dataset. This step is crucial as outliers can skew the results of PCA, leading to misleading interpretations of the variance and structure of the data.

PCA Implementation without outlier:

Explained Variance Ratio:

The explained variance ratios for the Nine principal components were approximately **38.37%**, **52.58%**, **60.52%** etc. Collectively, these components explained about **90.57%** of the total variance.

CONCLUSION:

Dimensionality Reduction Efficiency:

- **Clean Data**: The PCA results indicated that a significant portion of the variance could be captured in fewer dimensions (nine components), which is advantageous for visualization and further analysis.
- **Data with Outliers**: The dimensionality reduction might be less efficient if outliers skewed the data, potentially requiring more components to capture the same amount of variance.

LDA Implementation with outliers:

"The LDA process enables the model to identify the optimal linear combinations of features that best separate the different classes in the dataset."

- The accuracy of the LDA model with outliers is **94.79%**

Interpretation of the Confusion Matrix:

- **True Positives (TP)**: 1225 - Correctly predicted mushrooms of the positive class.
- **True Negatives (TN)**: 1085 - Correctly predicted mushrooms of the negative class.
- **False Positives (FP)**: 47 - Incorrectly predicted mushrooms of the positive class (Type I error).
- **False Negatives (FN)**: 81 - Incorrectly predicted mushrooms of the negative class (Type II error).

Outlier Removal:

- We utilized theInterquartile Range (IQR) method to identify and remove outliers from the dataset. This step is crucial as outliers can skew the results of LDA, leading to misleading interpretations of the variance and structure of the data.

Linear Discriminant Analysis (LDA) Results without outlier:

The accuracy of the LDA model on the test data was calculated as **99.07% .**The result suggests that LDA effectively captured the underlying patterns in the data without the interference of outliers.

Interpretation of the Confusion Matrix:

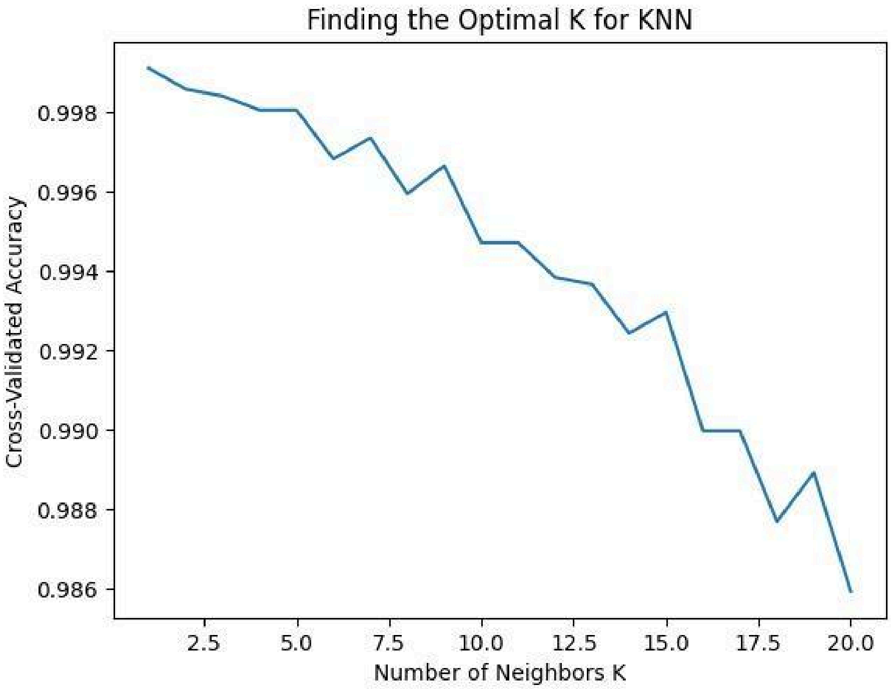
- **True Positives (TP)**: 428 instances were correctly classified as belonging to the positive class.
- **True Negatives (TN)**: 642 instances were correctly classified as belonging to the negative class.
- **False Positives (FP)**: 0 instances were incorrectly classified as positive when they were actually negative.
- **False Negatives (FN)**: 10 instances were incorrectly classified as negative when they were actually positive.

CONCLUSION :

Accuracy: The accuracy of the LDA model without outliers is slightly higher at **99.07%**, compared to **94.74%** when outliers are included. This difference, though seemingly small, indicates that outliers can affect overall classification performance.

K-Nearest Neighbors (KNN) Cross-Validation for Optimal K-Selection:

- **Accuracy Trend**: The cross-validation accuracy decreases as K increases from 1 to 20. The model performs best at smaller values of K, particularly around K=1, where the accuracy is close to 100%.
- **Decreasing Accuracy**: The higher K values smooth out decision boundaries, leading to increased bias and reduced ability to capture complex patterns, especially for imbalanced datasets or ones with distinct clusters.



K-Nearest Neighbors (KNN) Model on original data:

Accuracy:

- The KNN model achieved an accuracy of **99.917%** on the test set. This high accuracy indicates that the model is effective at classifying the data correctly in most cases.

- **Precision**: Both classes (0 and 1) have a precision of approximately **1.00**, meaning that almost all predictions for each class are correct.
- **Recall**: Both classes also achieved a recall close to **1.00**, meaning nearly all actual instances of each class were identified correctly.
- **F1-Score**: For both classes, the F1-score is also close to **1.00**, confirming a balanced performance in terms of precision and recall.

Logistic Regression on original data:

Accuracy: The logistic regression model achieved an accuracy of **94.67%** on the test set.

The logistic regression model shows robust performance even in the presence of outliers, with an accuracy of 94.67%. The KNN model further verifies these results, suggesting a reliable classification framework for the mushroom dataset.

SVM Model on original data:

The SVM model demonstrated exceptional effectiveness with a accuracy of **99.01% - with kernel linear,100% with kernel poly ,100% with kernel rbf**.

Decision Tree Classifier using Gini - Index :

The Decision Tree Classifier using the Gini index criterion performed effectively in classifying mushrooms, achieving competitive accuracy on both the training (95.74%) and test sets (96.14%). The high accuracy scores indicate that the model generalizes well, maintaining a robust performance on unseen data.

Decision Tree Classifier Report (Criterion: Entropy):

The Decision Tree Classifier using the entropy criterion performed effectively in classifying mushrooms, achieving competitive accuracy on both the training (95.65%) and test sets (95.85%). The decision tree model with entropy as the criterion and a maximum depth of 3 performs effectively, maintaining high accuracy and interpretability.

Classification Accuracies Across Different Methods:

Consider (n=13) (90% explained variance)

Methods of classification	**PCA with outliers**	**PCA without outliers**	**LDA with outliers**	**LDA without outliers**
KNN	1.00	1.00	0.9327	0.9962
Logistic regression	0.9136	1.00	0.9474	0.9935
SVM	0.9975	1.00	0.9495	1.00
● **Linear**	0.920	1.00	0.9483	1.00
● **Poly**	0.9991	1.00	0.9011	1.00
● **Rbf**	0.9975	1.00	0.9495	1.00
Decision Tree	0.9188	0.9889	0.9327	1.00

Consider (n=9) (80% explained variance)

Methods of classification	**PCA with outliers**	**PCA without outliers**	**LDA with outliers**	**LDA without outliers**
KNN	1.00	1.00	0.9327	0.9962
Logistic regression	0.9061	0.9843	0.9474	0.9935
SVM	0.994	1.00	0.9495	1.00
● **Linear**	0.9138	0.9888	0.9483	1.00
● **Poly**	0.9860	0.9990	0.9011	1.00
● **Rbf**	0.994	1.00	0.9495	1.00
Decision Tree	0.9188	0.9889	0.9327	1.00

CONCLUSION:

- KNN and SVM (RBF kernel) performed consistently well across both scenarios (with and without outliers) and both PCA dimensions (n=9 and n=13), achieving nearly perfect accuracy (1.00).
- Logistic Regression also performed well, particularly when outliers were removed.
- SVM with Polynomial and Linear kernels showed variability, with lower accuracy for some configurations (e.g., SVM Poly with LDA with outliers).
- Decision Tree performed relatively lower than KNN and SVM but still achieved high accuracy, especially with outliers removed.

Best Method: KNN and SVM (RBF kernel) are the best choices for the mushroom classification task, as they consistently achieved the highest accuracy across different dimensionality reductions techniques and outlier scenarios.