Inspiring Excellence

CSE422 CSE422: Artificial Intelligence
Project Name: Mushroom Edibility Prediction

Submitted By:
Group: 8

ID	Name
22201884	M.A.Zubaer
22201391	Fatema Siddika Tanha

Section: 16

Table of contents

Introduction

This project is all about a mushroom dataset for classifying mushrooms that includes specific morphological and ecological features. Each row of the dataset represents a particular mushroom instance characterized by characteristics like environment, season, stem dimensions, gill characteristics, cap form, and color. The dataset also includes a feature ‘class’,
That indicates either the mushroom is edible or poisonous.

The dataset is appropriate for implementing a variety of machine learning approaches, especially supervised classification algorithms, because the features include a combination of numerical and categorical variables.

The primary objective of this project is to develop a machine learning model using valid features that can reliably predict whether a mushroom is edible or poisonous. The project aims to help identify potentially harmful mushrooms without requiring biological expertise by learning trends from past data.

Problem Statement

Incorrect identification of poisonous mushrooms can lead to health risks and even death. Particularly for almost identical species, manual identification is prone to human mistake and needs specialized understanding. By using data-driven techniques to automate mushroom categorization and lessen dependence on subjective judgment, this study tackles the issue.

Motivation

The motivation of this project is to create a trustworthy machine learning model that can correctly identify edible and poisonous mushrooms based on their ecological and physical characteristics. A major concern to public health is mushroom poisoning, particularly in areas where wild mushrooms are often harvested. This study intends to enhance safer decision-making, lower the danger of unintentional poisoning, and show how machine learning can be applied to practical biological and safety-critical issues by utilizing data-driven categorization algorithms.

Dataset description

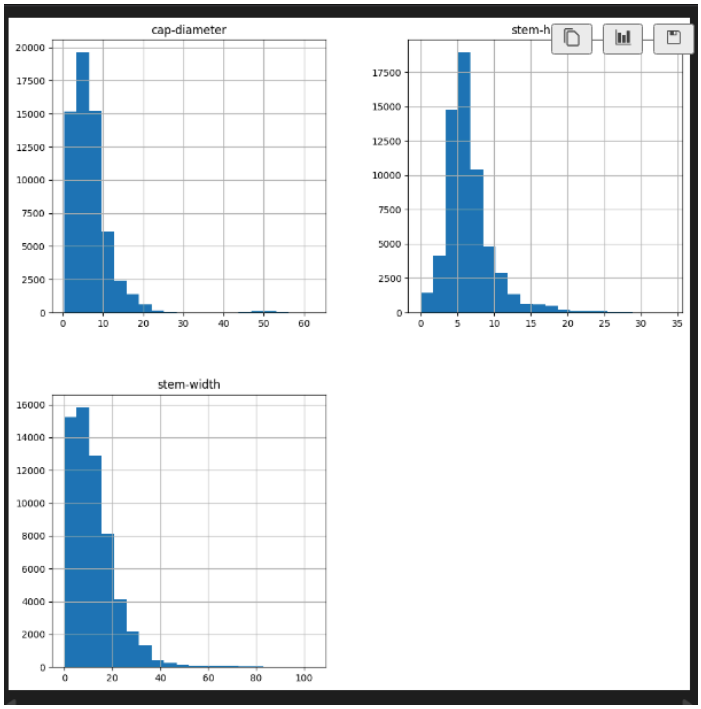
In this dataset, there are initially 61069 rows and 21 columns. The target column is “class” that defines whether the mushroom is edible or poisonous. This makes the problem a binary classification problem, as the categorical target is either ‘e’ (edible) or ‘p’ (poisonous). There are a total of 61069 datapoints. The dataset has mixed type of features. The numerical features are ['cap-diameter', 'stem-height', 'stem-width']. The categorical features are ['class', 'cap-shape', 'cap-surface', 'cap-color', 'does-bruise-or-bleed', 'gill-attachment', 'gill-spacing', 'gill-color', 'stem-root', 'stem-surface', 'stem-color', 'veil-type', 'veil-color', 'has-ring', 'ring-type', 'spore-print-color', 'habitat', 'season'].

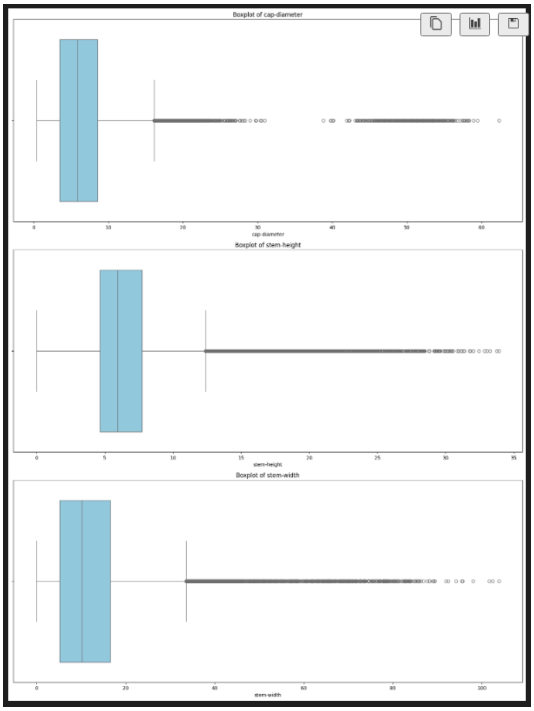
Exploratory Data Analysis

The numerical features of the dataset have high variance and all the numerical features are highly right skewed. This brings a good amount of outliers.

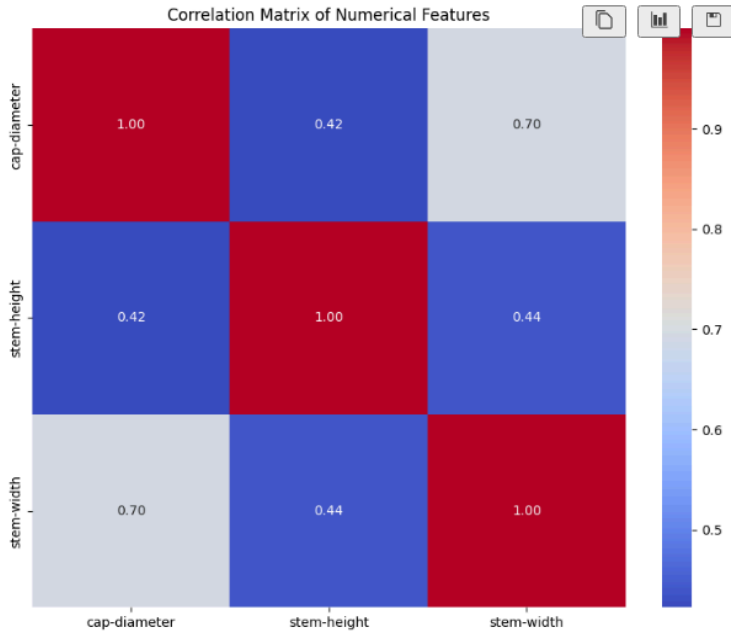
```
numerical_data.var()
✓ 0.0s

cap-diameter      27.718592
stem-height       11.357014
stem-width       100.720394
dtype: float64
```

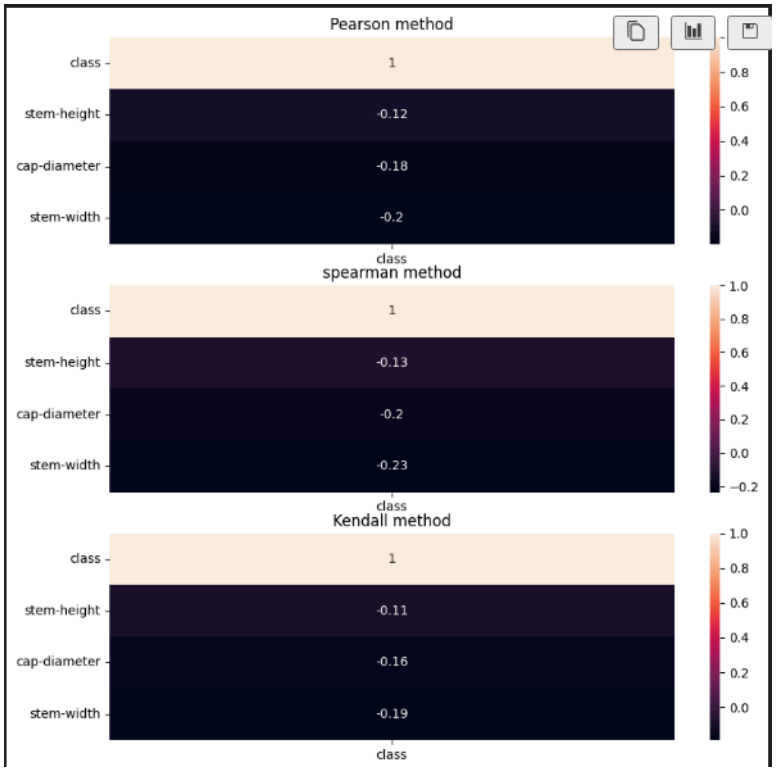
```
numerical_data.skew()
✓ 0.0s

cap-diameter      3.822844
stem-height       2.020904
stem-width        2.164957
dtype: float64
```





Correlation Matrix of Numerical Features:





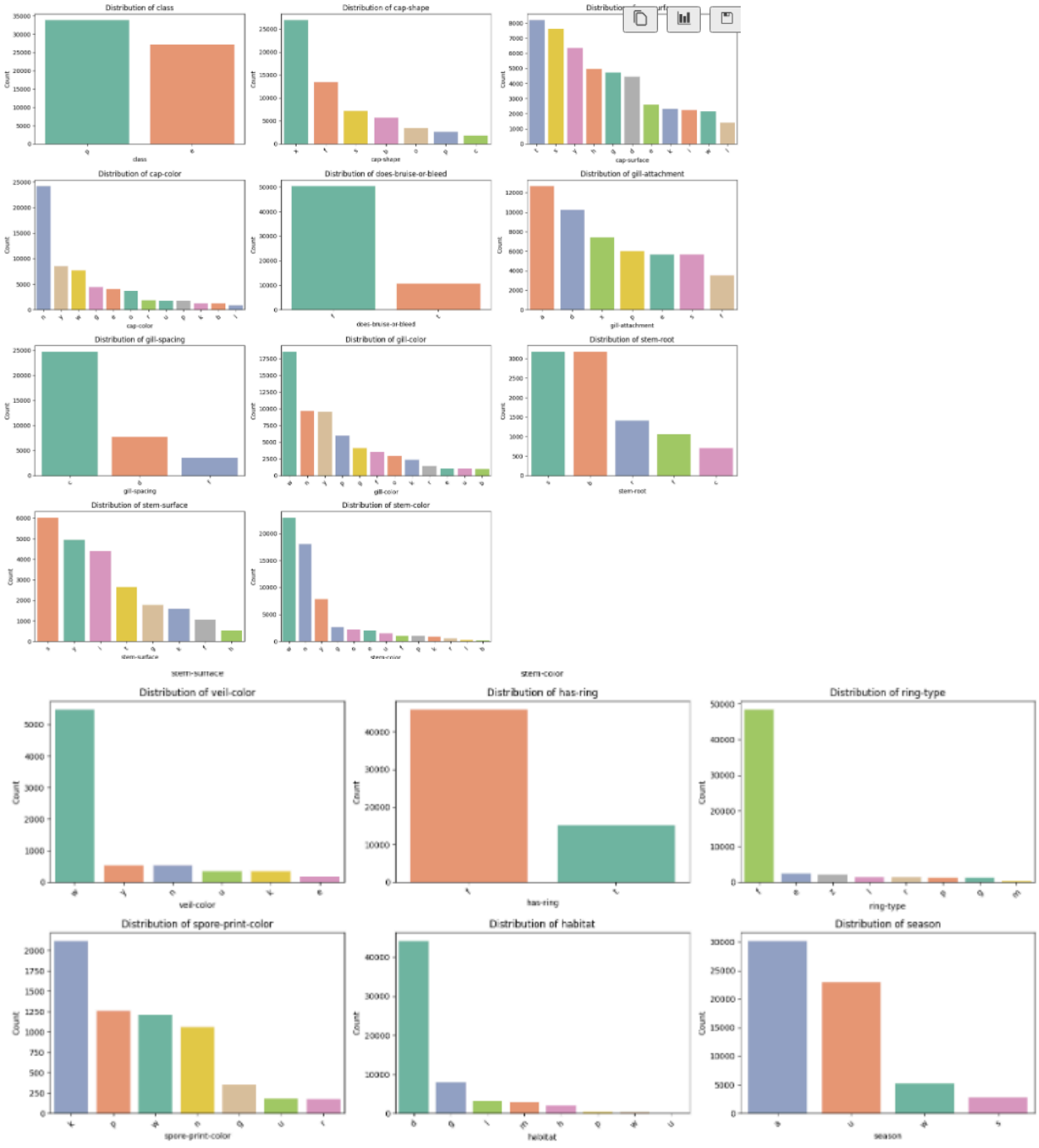
The results of the correlation analysis show that the numerical features in the dataset have small correlations, with a positive correlation (≈ 0.70) between cap diameter and stem width. However, the relationships with the target variable (mushroom edibility) are weak, with values ranging

from −0.1 to −0.2 across different correlation methods, suggesting these features do not strongly influence toxicity. This indicates that the edibility of mushrooms is probably more influenced by categorical traits.

The categorical features unique categories:

```
class                     2
cap-shape                 7
cap-surface              11
cap-color                12
does-bruise-or-bleed      2
gill-attachment           7
gill-spacing              3
gill-color               12
stem-root                 5
stem-surface              8
stem-color               13
veil-type                 1
veil-color                6
has-ring                  2
ring-type                 8
spore-print-color         7
habitat                   8
season                    4
dtype: int64
```

The distribution of data in each categorical feature:



As we can see where there are some noticeable amounts of imbalance in some of the categorical features. Although, the target column is not significantly imbalanced.

Feature Selection and Data Preprocessing

Null Values Handaling

Dropped the feature veil-type. Because it has only 1 unique category. Hence, this feature is meaningless for model training.
Null values count in each features:

```
class                     0
cap-diameter              0
cap-shape                 0
cap-surface           14120
cap-color                 0
does-bruise-or-bleed      0
gill-attachment        9884
gill-spacing          25063
gill-color                0
stem-height               0
stem-width                0
stem-root             51538
stem-surface          38124
stem-color                0
veil-color            53656
has-ring                  0
ring-type              2471
spore-print-color     54715
habitat                   0
season                    0
dtype: int64
```

Some of the columns have a very high amount of null values. That makes these features redundant. So, dropping the features with ≥40% missing values. Dropped columns due to ≥49% missing values: gill-spacing, stem-root, stem-surface, veil-color, spore-print-color.

Still there are some null values in 'cap-surface', gill-attachment, ring-type’ features . The Mode imputation is suitable for these features. Because, it preserves the distribution of the data and avoids creating artificial categories.

Data Preprocessing

For high variance and high skewness the suitable solution is log transformation. Because, it improves the symmetry of the distribution and reduces the impact of serious outliers by compressing large values.

For imbalance in categorical values. Grouping rare categories in categorical features into 'Other' if they are <5% of total samples, increases model performance significantly. Also, this reduces noise and overfitting, simplifies the model and improves statistical reliability.

Preparing Data for model training

For model training, encoding the categorical features is important. Because it makes the data suitable for machine learning algorithms and also reduces biases. For this dataset's categorical features, One Hot encoding would be a perfect pick as the categorical features are non-hierarchical. Also for Numerical features scaling is also important. Because, this brings all features to the same scale, making model performance better. For this dataset's numerical features we used StandardScaler. We used a Label encoder for the target column for better learning.

Data Splitting

Train data: 80%
Test data: 20%
Stratified the data while splitting. This ensures that the class distribution in the split is similar to the real dataset. This reduces imbalance in train and test datasets.

Model Accuracy and Evaluation

Logistic Regression

```
Logistic Regression:
              precision    recall  f1-score   support

           0       0.69      0.64      0.67      5158
           1       0.73      0.76      0.74      6352

    accuracy                           0.71     11510
   macro avg       0.71      0.70      0.71     11510
weighted avg       0.71      0.71      0.71     11510
```

SVM Classifier

```
Support Vector Machine:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5158
           1       1.00      1.00      1.00      6352

    accuracy                           1.00     11510
   macro avg       1.00      1.00      1.00     11510
weighted avg       1.00      1.00      1.00     11510
```

Decision Tree

```
Decision Tree:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5158
           1       0.99      0.99      0.99      6352

    accuracy                           0.99     11510
   macro avg       0.99      0.99      0.99     11510
weighted avg       0.99      0.99      0.99     11510
```

Random Forest

```
Random Forest:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5158
           1       1.00      1.00      1.00      6352

    accuracy                           1.00     11510
   macro avg       1.00      1.00      1.00     11510
weighted avg       1.00      1.00      1.00     11510
```

K-Nearnest Neighbours

```
K-Nearest Neighbors:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5158
           1       1.00      1.00      1.00      6352

    accuracy                           1.00     11510
   macro avg       1.00      1.00      1.00     11510
weighted avg       1.00      1.00      1.00     11510
```

Gaussian Naive Bayes

```
Gaussian Naive Bayes:
              precision    recall  f1-score   support

           0       0.65      0.71      0.68      5158
           1       0.75      0.68      0.71      6352

    accuracy                           0.70     11510
   macro avg       0.70      0.70      0.70     11510
weighted avg       0.70      0.70      0.70     11510
```

Bernoli Naive Bayes

```
Bernoulli Naive Bayes:
              precision    recall  f1-score   support

           0       0.64      0.65      0.65      5158
           1       0.72      0.71      0.71      6352

    accuracy                           0.68     11510
   macro avg       0.68      0.68      0.68     11510
weighted avg       0.68      0.68      0.68     11510
```

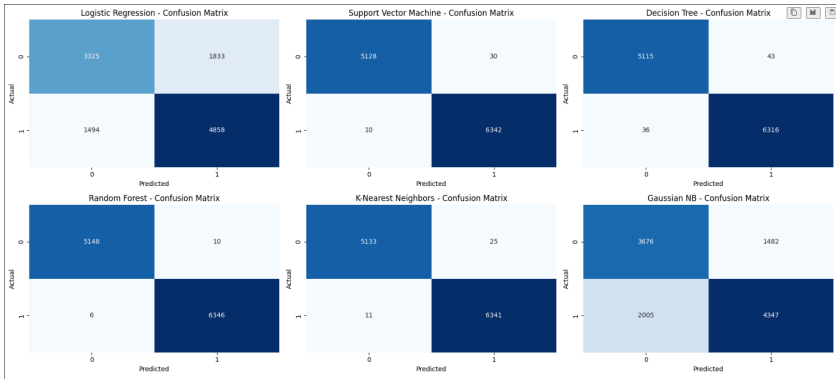
XGBoost Classifier

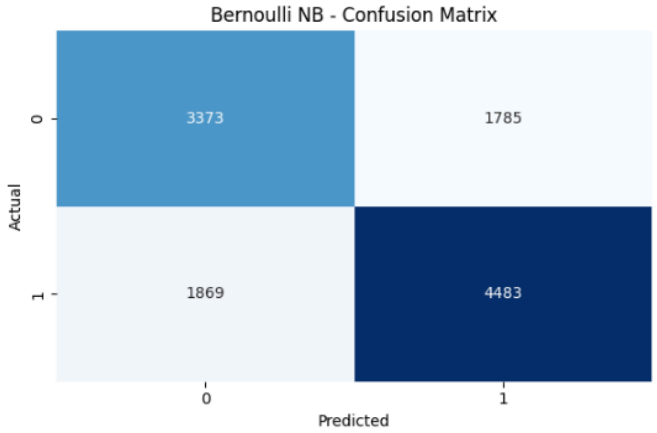
```
XGBoost Classifier:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5158
           1       1.00      1.00      1.00      6352

    accuracy                           1.00     11510
   macro avg       1.00      1.00      1.00     11510
weighted avg       1.00      1.00      1.00     11510
```

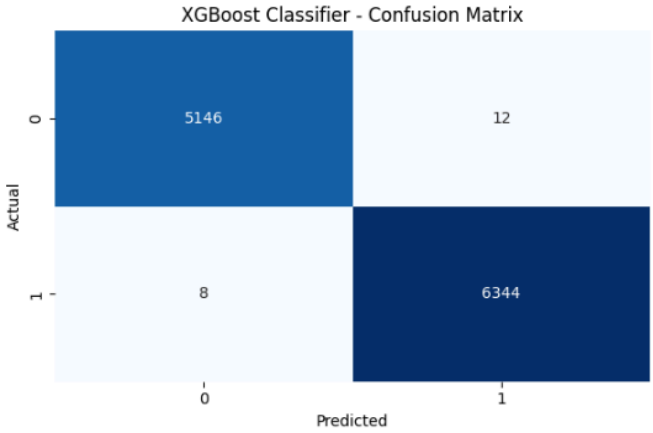
Neural Network

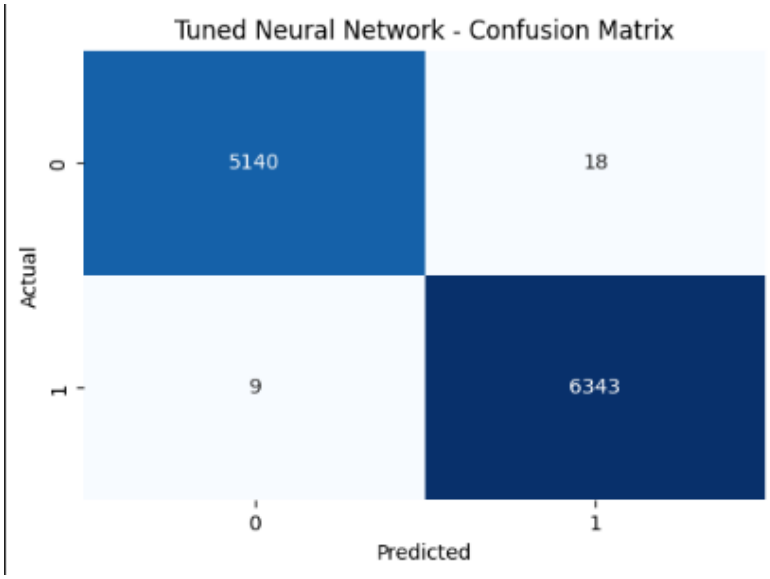
```
Epoch 27: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
1151/1151 - 1s - 1ms/step - accuracy: 0.9972 - loss: 0.0070 - val_accuracy: 0.9987 - val_loss: 0.0031 - learning_rate: 2.5000e-04
Epoch 28/50
1151/1151 - 3s - 2ms/step - accuracy: 0.9977 - loss: 0.0063 - val_accuracy: 0.9987 - val_loss: 0.0034 - learning_rate: 1.2500e-04
Epoch 29/50
1151/1151 - 2s - 1ms/step - accuracy: 0.9980 - loss: 0.0059 - val_accuracy: 0.9986 - val_loss: 0.0031 - learning_rate: 1.2500e-04
Tuned Model Test Accuracy: 0.9980
```

Confusion Matrix for Each Model

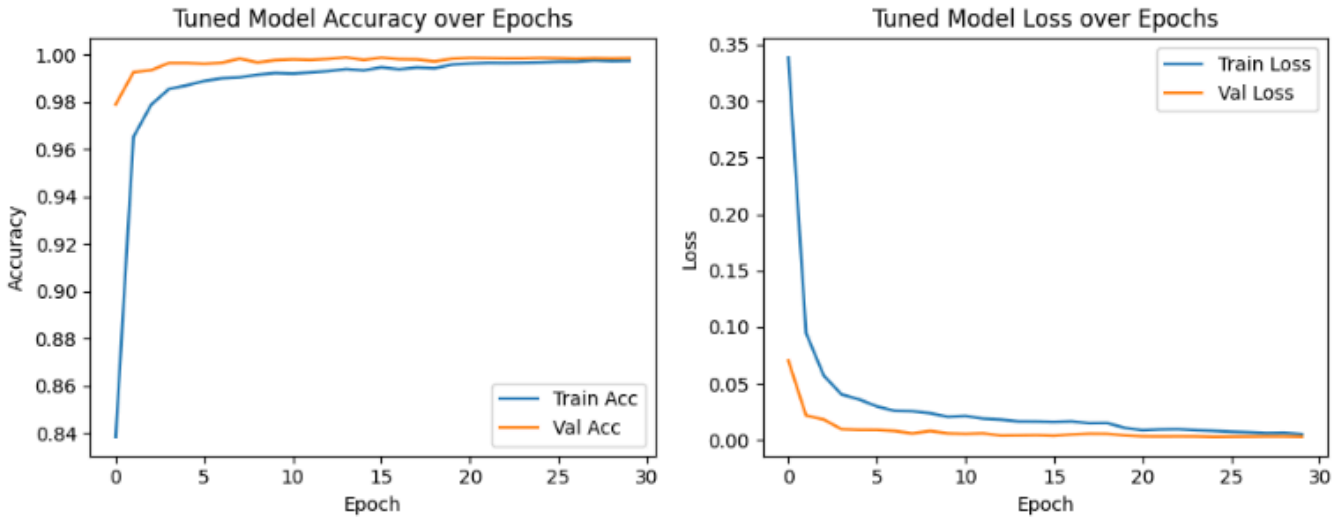

Logistic Regression - Confusion Matrix
3325
1833
1494
4858
Support Vector Machine - Confusion Matrix
5128
30
10
6342
Decision Tree - Confusion Matrix
5115
43
36
6316
Random Forest - Confusion Matrix
5148
10
6
6346
K-Nearest Neighbors - Confusion Matrix
5133
25
11
6341
Gaussian NB - Confusion Matrix
3676
1482
2005
4347
Actual
Predicted


Bernoulli NB - Confusion Matrix
3373
1785
1869
4483
Actual
Predicted

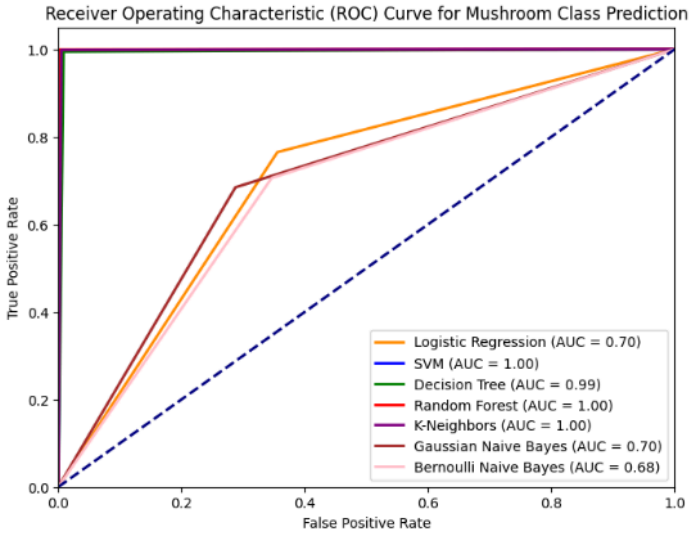

XGBoost Classifier - Confusion Matrix
5146
12
8
6344
Actual
Predicted


Tuned Neural Network - Confusion Matrix
5140
18
9
6343
Actual
Predicted

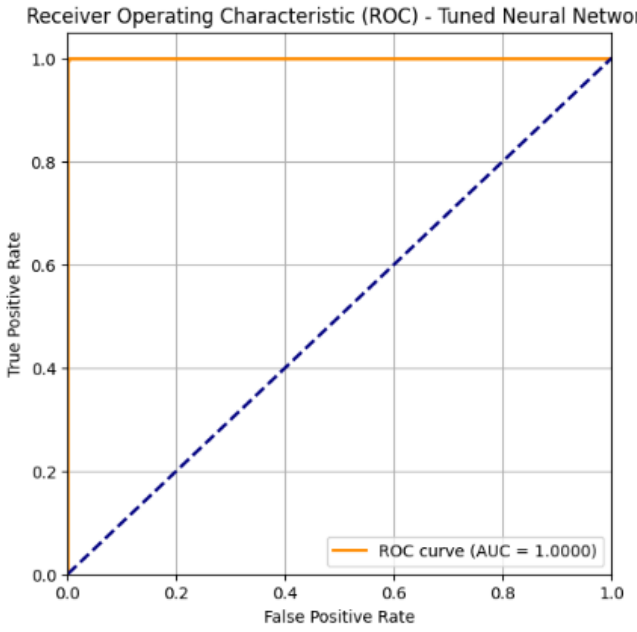
Loss and accuracy of Neural Network model over Epochs



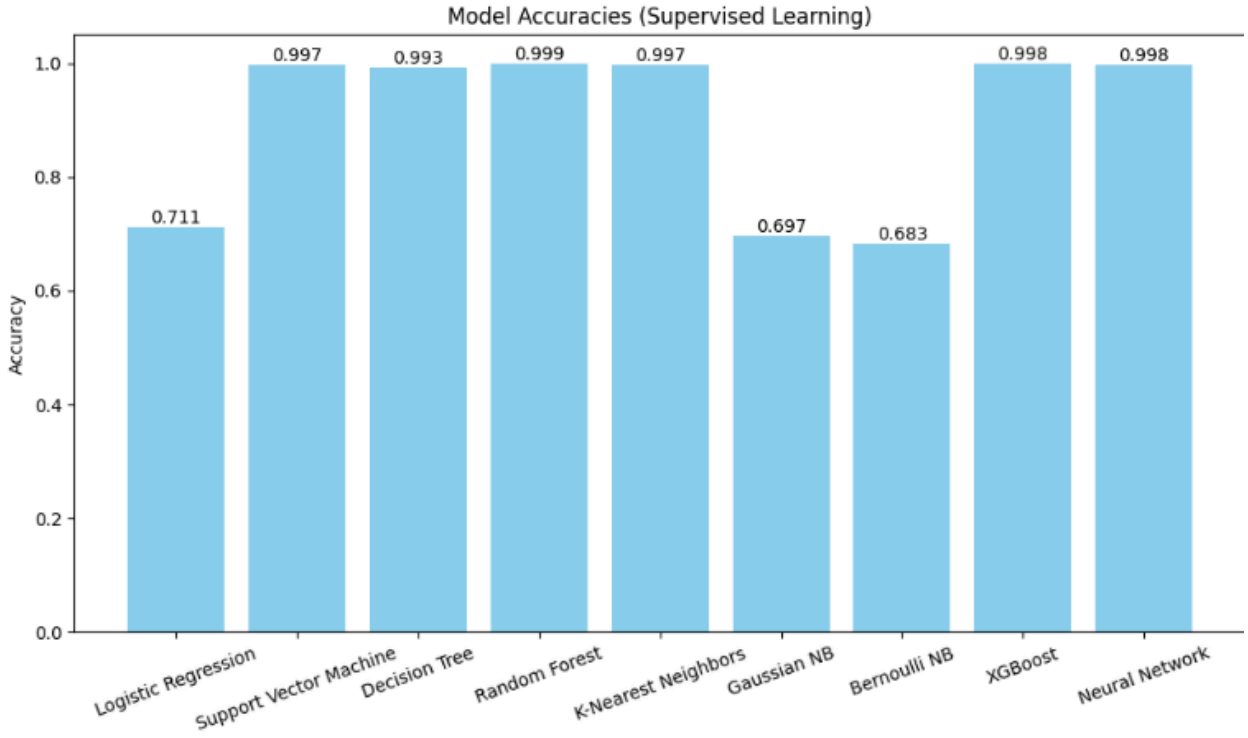
AUC score, ROC curve for each mode (without Neural Neural Network)



AUC score, ROC curve for Neural Network



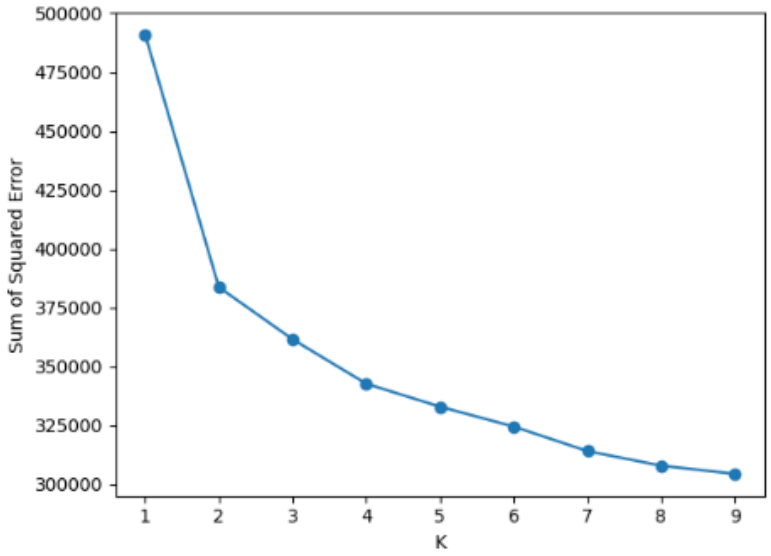
Model Accuracy Comparison Bar chart



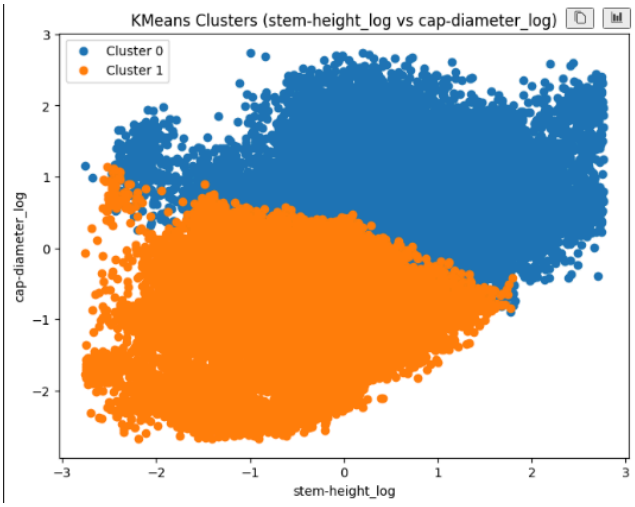
The bar chart demonstrates that models like SVM, Decision Tree, Random forest, K-Nearest neighbours, XGBoost and Neural Network outperformed other models. But the model with most accuracy is Random Forest.

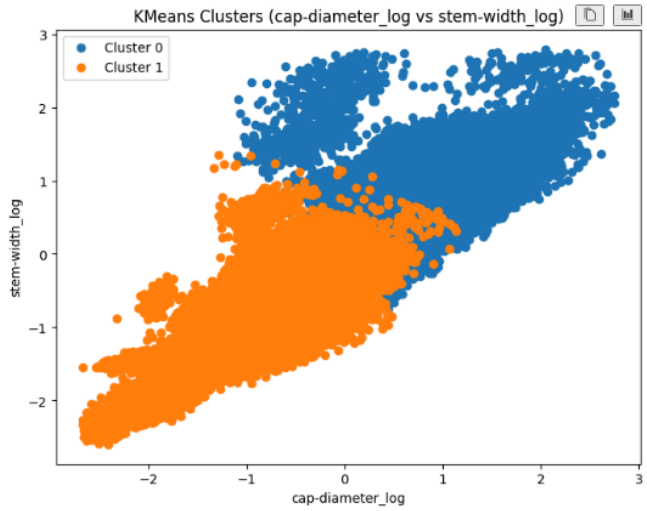
Unsupervised Learning (KMeans Clustering)

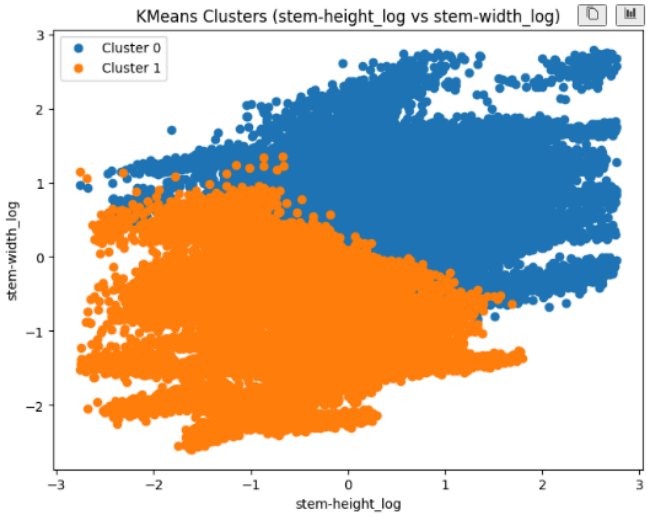
If we consider this as an unsupervised learning problem. The data preprocessing would be the same for this also.
Sum of Squared Error vs K



This indicates that on 2 clusters the error reduction starts to slow down (the elbow method). This cluster number will balance accuracy and simplicity. KMeans Clusters of numerical features:







These Plots demonstrate that the number of clusters is accurate.

Conclusion

This project successfully developed and evaluated different machine learning models to categorize mushrooms as edible or poisonous based on their morphological and ecological characteristics. After thorough data preparation and feature selection, models such as Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors, XGBoost, and Neural Network, achieved near-perfect accuracy with 99%-100%.

The performance of models indicates that selected categorical and numerical features are effective for classifying edible vs poisonous mushrooms. Categorical features like cap shape, gill color, and habitat are more important than numerical features, supporting the idea that visual and

environmental cues are essential in mushroom identification. High performing models like SVM, Random Forest, XGBoost, and KNN achieved approximately 100% accuracy, indicating strong pattern recognition capabilities. Moderate performing models, such as Logistic Regression and Naive Bayes yielded lower accuracy with 68–71%.This is likely due to linear assumptions and sensitivity to feature distributions. The Neural Network reached 99.8% accuracy with tuning, highlighting its learning capability for complex interactions, with higher computational demands.

Strong classification signals were seen from categorical features, which were efficiently utilized by tree-based and kernel-based models. Log transformation and one-hot encoding were used to manage skewness and retain information, while stratified splitting ensured class balance. Nonlinear models outperformed linear models, indicating feature interactions. However, issues developed due to missing values, skewed distributions in numerical features, and categorical feature class imbalance. This necessitates repeated hyperparameter adjustment and unusual category groups to reduce overfitting. Although tree-based models provided some interpretability, the high-dimensional category space made it difficult to create simple rules.

K-Means clustering on log-transformed numerical features identified two natural clusters, indicating potential biological distinctions between edible and poisonous categories. However, interpreting these clusters necessitated cross-referencing with supervised results due to the lack of direct label alignment in unsupervised methods.

This project highlights the effectiveness of machine learning in addressing real-world, safety-critical categorization challenges. The created models can greatly lower the danger of mushroom poisoning by making accurate, data-driven forecasts. Future research might concentrate on model interpretability, real-time deployment, and increasing the dataset to include additional uncommon species for even greater application