University of Science and Technology of Ha Noi

Analysis of Spatial and Temporal Data

Final Project Report

Poisonous Mushroom Analysis and Classification

Student Name	Student ID
Le Linh Long	22BI13262
Pham Tuan Nam	22BI13326
Nguyen Tuan Khai	22BI13202
Nguyen Quang Huy	22BI13195
Pham Thai Son	22BI13397
Nguyen Hai Dang	22BI13073

Lecturer: Dr. Nguyen Xuan Thanh

Submission Date: 04/04/2025

Contents

List of Figures

List of Tables

1 Introduction

With the advent of machine learning techniques, we have been able to extract and glean meaningful insights from complex datasets across numerous domains, including biological sciences, agriculture, and food safety [4]. One such application that has risen to prominence lies in the classification of mushrooms for edibility, a task that bears significant public health implications. Traditionally, the task is done manually through visually examining the mushroom for certain characteristics, which is not only labor-intensive but prone to misidentification, possibly resulting in severe consequences. Therefore, there needed to be an accurate and efficient automated classification system for the task of poisonous mushroom classification [11], something this report will investigate.

In this study, we rely on the dataset from the Kaggle competition "Playground Series - Season 4, Episode 8" [5]. The dataset consists of over 3 million observations and 22 features, with mainly categorical variables and few numericals. The objective is to discern which attribute of the mushroom can best lead to identification and distinguishing of poisonous mushrooms from edible ones, using various approaches both in the traditional statistical analysis domain and by employing modern machine learning techniques.

The remainder of this report shall be structured as follows: Section 2 outlines the dataset as well as delve further into exploratory data analysis, along with describing the methodology applied. Section 3 will present the results of our findings in terms of statistical analysis and predictive modeling; and lastly Section 4 will conclude with a discussion of our results, as well as addressing unanswered questions that were beyond our purview but may become directions for future works.

Table 1 summarizes the responsibility of each members in our work.

Nguyen Quang Huy	Explanatory Data Analysis
Pham Thai Son	Data Preprocessing
Le Linh Long	Distribution Analysis
Nguyen Hai Dang	Hypothesis Testing
Nguyen Tuan Khai	Apply Classification Models
Pham Tuan Nam	Model Performance Evaluation

Table 1: Contribution of each member in our work.

2 Data and Methodology

The dataset consists of 3,116,945 instances and 22 features, designed for a binary classification task to distinguish between poisonous and edible mushrooms. The features pertain to attributes describing physical characteristics such as cap shape, odor, gill color, spore print, and bruising, among others. The target variable `class` is a binary label indicating whether a mushroom is poisonous (1) or edible (0). Preliminary inspection revealed the presence of missing values and variable data types, necessitating appropriate preprocessing prior to model development.

Feature	Type	Description
cap-diameter	Numerical	Size of mushroom cap
cap-shape	Categorical	Shape of mushroom cap
cap-surface	Categorical	Surface texture of cap
cap-color	Categorical	Color of cap
does-bruise-or-bleed	Binary	Whether mushroom bruises upon impact
gill-attachment	Categorical	Attachment type of the gills to the stalk
gill-spacing	Categorical	Spacing between gills
gill-color	Categorical	Color of the gills
stem-height	Numerical	Height of mushroom stem
stem-width	Numerical	Width of mushroom stem
stem-root	Categorical	Root type of mushroom stem
stem-surface	Categorical	Surface texture of the stem
stem-color	Categorical	Color of the stem
veil-type	Categorical	Type of veil on mushroom
veil-color	Categorical	Color of veil
has-ring	Categorical	Presence of rings on stalk
ring-type	Categorical	Types of ring present
spore-print-color	Categorical	Color of the mushroom's spore print
habitat	Categorical	Mushroom's natural habitat
season	Categorical	Season during which the mushroom appears
class	Binary label	Whether mushroom is poisonous or edible

Table 2: Features in the dataset

2.1 Exploratory Data Analysis

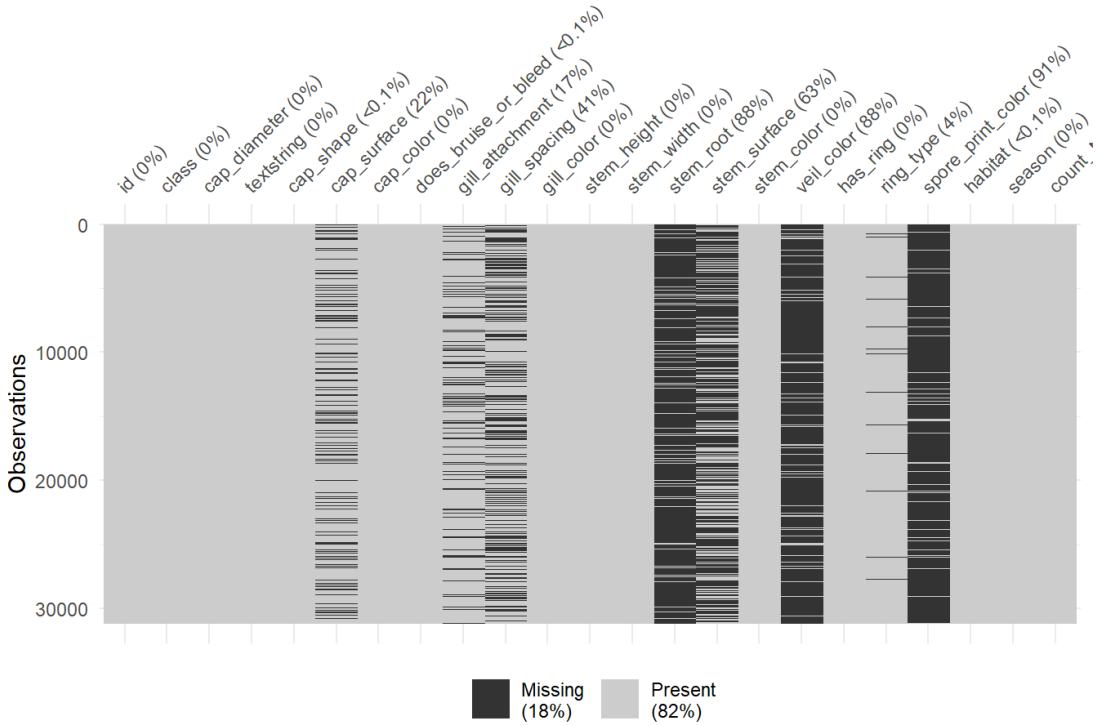

Figure 1: Missingness map of dataset variables

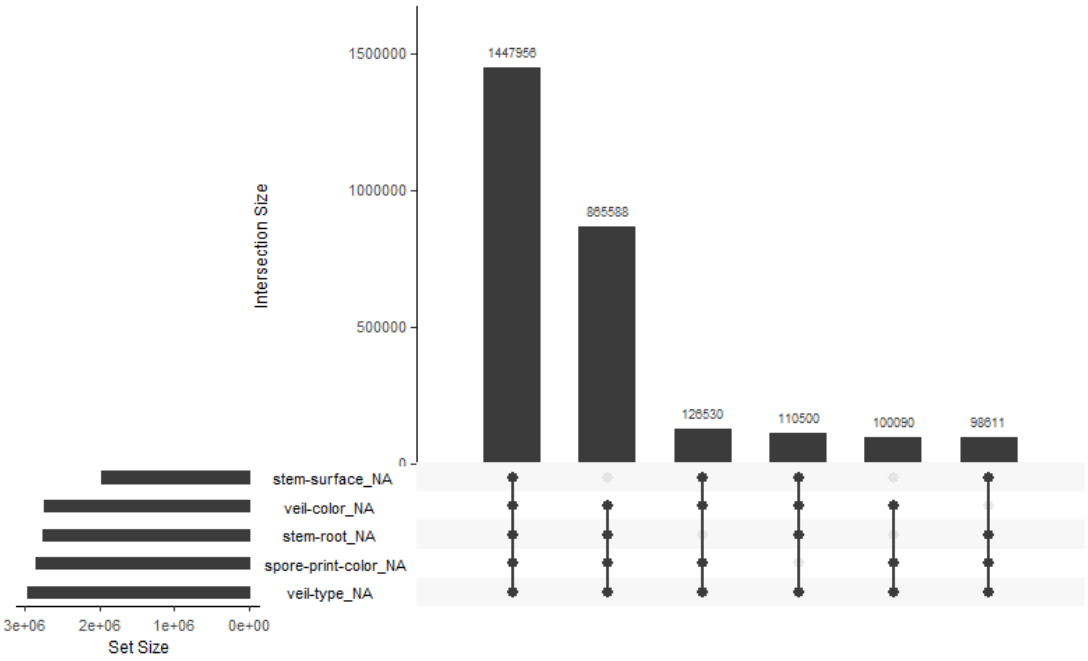

Figure 2: Plot of missing data intersections

Before we delve into feature analysis, we first must understand the pattern and extent of missing data, to ensure the reliability of subsequent modeling efforts. Figure 1 presents us with a missingness map, which describes which observation lacks which feature by displaying them as black. From here we learn that about 18% of the data is missing from our dataset, particularly in features such as `stem-root`, `stem-surface`, `veil-type`, `spore-print-color` and `veil-color`. Due to the large number of missing values these features do not contain in themselves relevant information to the analysis, signaling that we can remove them from further scrutiny.

Figure 2 further expounds upon the pattern of missing data by describing the intersection of missing values across variables. The most frequently co-occurring missing pattern involves all five of the aforementioned features and encompasses over 1.4 million observations. Other subsets of variables also exhibit shared missingness, suggesting that these features may be conditionally missing under specific circumstances. With this insight in mind, we elected to set a threshold so as to remove features with a large part of its values missing ($\geq 50\%$) from the dataset.

2.1.1 Continuous features

The dataset includes three continuous features: `cap_diameter`, `stem_height`, and `stem_width`. Here, we will investigate their relationship with each other as well as with the target variable `class`.

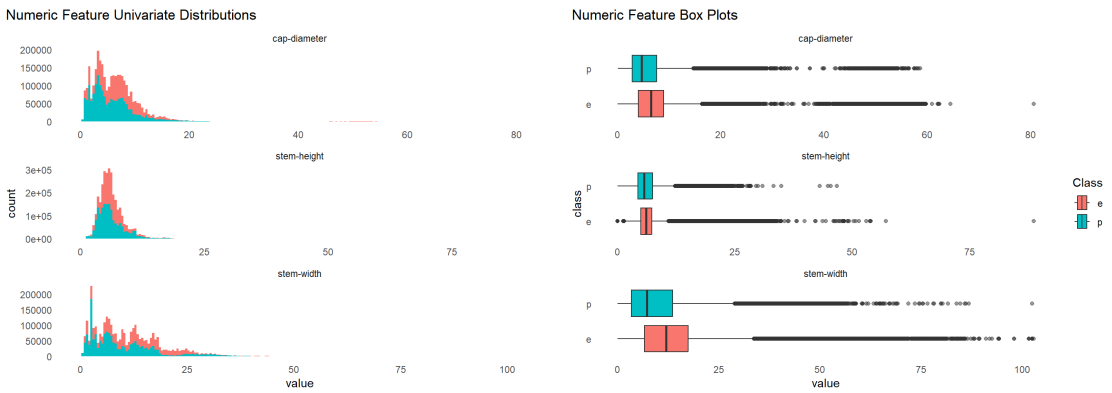

Figure 3: Univariate analysis of continuous features within our dataset

Univariate distributions (Figure 3a) reveal that our numeric features are heavily skewed left, and suggests the presence of long-tailed distribution and potential outliers. What's more interesting however, is when examining the distribution of these variables, we observe that the range of these variables differs based on class, as poisonous mushrooms tend to show slightly higher variability. These observations are further consolidated in the box plots in Figure 3b, which confirms the presence of outliers while also showing that the median and interquartile range for features `cap-diameter` and `stem-width` show rightward shift in distribution in the edible class. These patterns can be understood from an evolutionary perspective, as when we understand the mushrooms themselves as fruiting bodies to spread the spore and thus the genetic material for the mycelium beneath, larger mushrooms can attract mutualists such as animals or humans that come to eat or cultivate them, thus spreading their spores elsewhere to multiply. The opposite however can be said for poisonous mushrooms, whose strategy for multiplication do not involve other organisms. These species may develop toxins as a way to ward off animals, a process which naturally draws energy and sustenance from the mushroom and resulting in less energy to allocate towards growing its fruiting body[3, 10]. This is one of many hypotheses, but these patterns already indicate that these continuous features carry moderate predictive information, and may serve as useful components in downstream modeling.

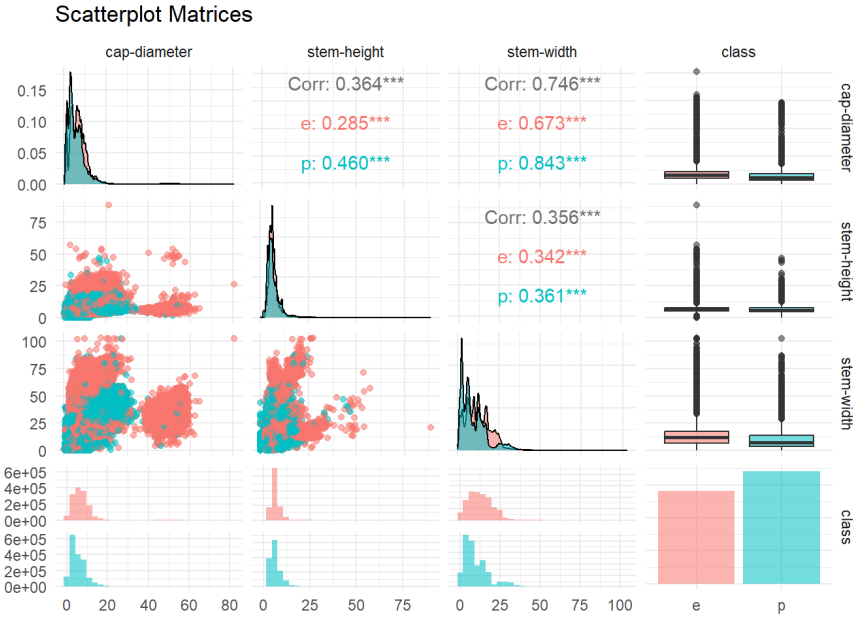

Figure 4: Bivariate analysis of continuous features within our dataset

Figure 4 presents a scatterplot matrix that highlights the pairwise correlations between numeric features, both globally and class-conditioned. A strong positive correlation is shown between `cap-diameter` and `stem-width`, with the *** indicating $p < 0.001$ reflecting that the relationship is very statistically significant. This is true to a lesser extent when analyzing the other two relationships, reflecting that they may not strongly reflect each other, with global correlations around 0.35.

2.1.2 Categorical features

The Figure 5 represents the number of observed instances per unique sample. Its distribution reveals to us how frequently particular combinations of mushroom characteristics appear. Not withstanding NA cases corresponding to missing values, certain characteristics are shown to be more present within edible mushrooms like brown cap color or smooth cap surface.

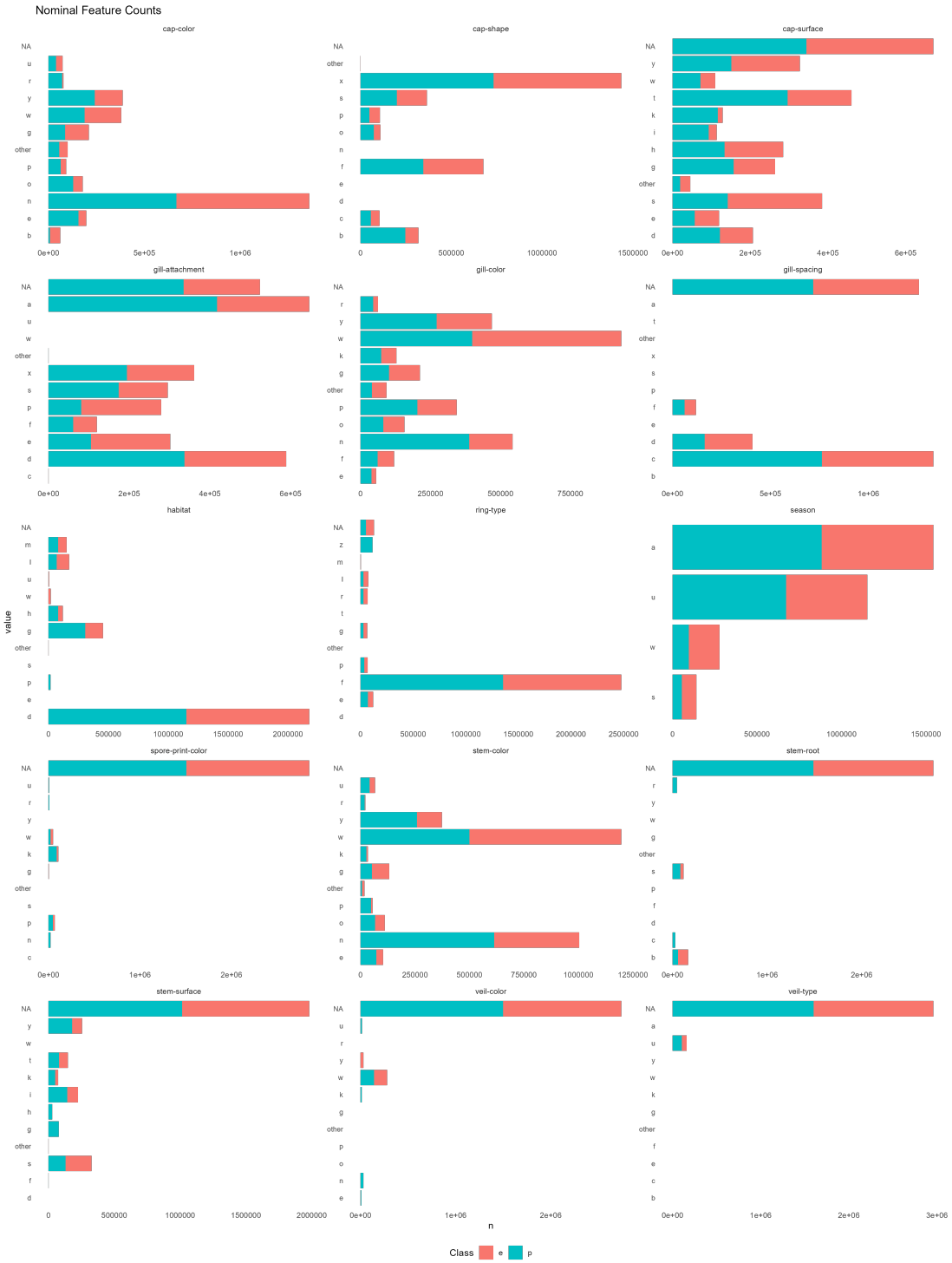

Figure 5: Number of samples for categorical features within our dataset, separated by class

For this analysis, we will focus only on the features Cap Shape and Cap Color, and how they help reflect whether a mushroom is poisonous or edible. In the field, when either professional mycologists or general foragers examine the mushroom for edibility, they first examine them for these characteristics. As [9] stated, "*One of the initial steps in identifying mushrooms is examining their* ***color and cap shape***." These two features are generally emphasized by professionals both due to their accessibility as well as their taxonomic significance.

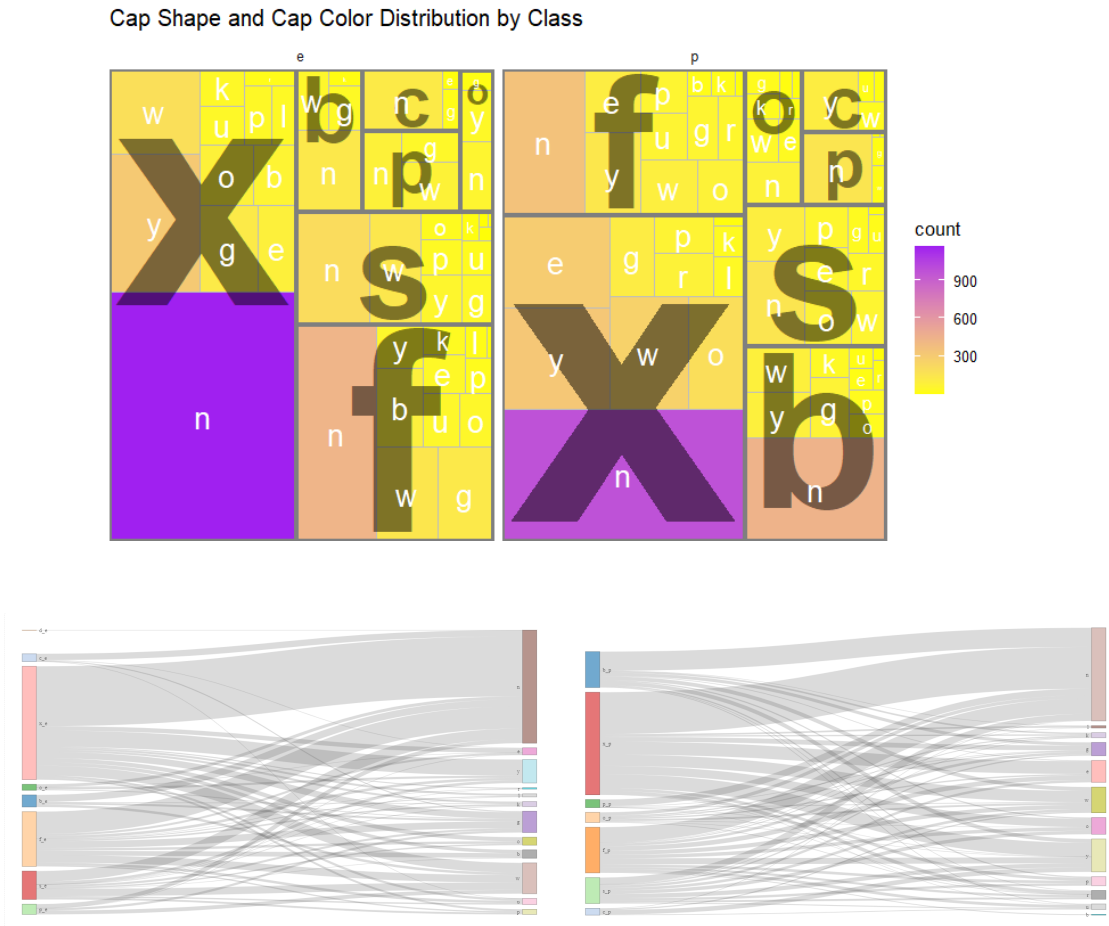

Figure 6: Bivariate analysis of categorical features `cap-shape` and `cap-color`

In both Figures above, we see that the most common shape-color combinations of mushroom caps are convex-brown or convex-yellow, both of which appear in edible as well as poisonous mushrooms and are thus not helpful towards discerning its class. Much more significant than the dominant phenotypes, those that occur less frequently can help us identify them much more accurately. In the Figures, we see that cap shape and color combinations like bell-yellow or flat-gray are more dominant in the poisonous class, while mostly absent from the edible class. This tells us that the cap morphology and pigmentation can have discriminatory power in discerning poisonous mushrooms from edible ones, which is a very informative insight.

2.2 Methodology

2.2.1 Distribution Analysis

Kernel Density Estimation In statistics, kernel density estimation (KDE) is the application of kernel smoothing for probability density estimation, i.e., a non-parametric method to estimate the probability density function of a random variable based on kernels as weights. KDE answers a fundamental data smoothing problem where inferences about the population are made based on a finite data sample. Key Components of KDE:

- **Kernel Function:** The kernel is a smooth, symmetric function that is used to place a "weight" on each data point. The most commonly used kernel is the Gaussian kernel, which takes the form of a bell curve. However, other kernel functions (such as Epanechnikov, uniform, or triangular) can also be used.
- **Bandwidth:** The bandwidth parameter determines the width of the kernel and, consequently, the smoothness of the resulting density estimate. A small bandwidth can result in a very rough (high-variance) estimate, while a large bandwidth can lead to oversmoothing (underfitting) the data.

The KDE is computed as the sum of the kernels placed at each data point, scaled by the bandwidth. Mathematically, for a set of data points $\{x_1, x_2, \ldots, x_n\}$, the KDE is defined as:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Where:

- $\hat{f}_h(x)$ is the estimated density at point x,
- n is the number of data points
- h is the bandwidth,
- K is the kernel function (like Gaussian),
- x_i are the data points.

Quantile-Quantile Plot A quantile-quantile (Q-Q) plot is a graphical tool used to assess whether a dataset follows a specific probability distribution or whether two datasets come from the same population. It's particularly useful for checking if data follows a normal distribution or another known distribution. Q-Q plots are widely used in statistics, data analysis, and quality control to verify assumptions and identify any discrepancies from expected distributions.

Quantiles are values that divide a dataset into intervals containing equal probabilities or proportions of the total distribution. They help describe the spread and shape of the data. Some of the most common types of quantiles include:

- **Median (50th percentile):** The median is the middle value in a dataset when arranged from smallest to largest. It divides the dataset into two equal halves.
- **Quartiles (25th, 50th, and 75th percentiles):** Quartiles break the dataset into four equal parts. The first quartile (Q1) marks the value below which 25% of the data falls, the second quartile (Q2) is the median, and the third quartile (Q3) represents the value below which 75% of the data falls.
- **Percentiles:** These are similar to quartiles but divide the data into 100 equal parts. For example, the 90th percentile is the value below which 90% of the data lies.

Interquartile Range Outliers are observations that deviate significantly from the overall pattern of a dataset and this deviation can lead to poor results in analysis. Interquartile Range (IQR) is a technique that detects outliers by measuring the variability in a dataset. In this article we will learn about it.

IQR is used to measure variability by dividing a data set into quartiles. The data is sorted in ascending order and then we split it into 4 equal parts. The values Q1 (25th percentile), Q2 (50th percentile or median) and Q3 (75th percentile) separate the dataset in 4 equal parts.

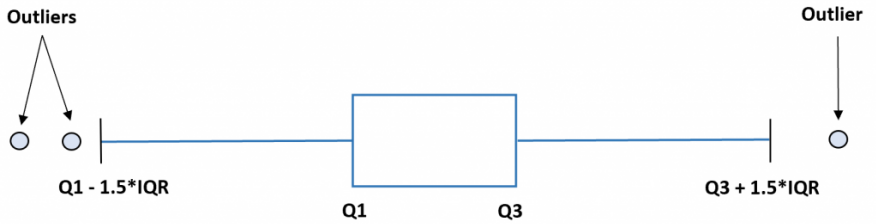

Figure 7: Find outliers with IQR

2.2.2 Statistical Tests

For binary features, we utilized one-sample and two-sample z-test to test for the proportions.

One sample z-test proportions Given a binary variable, we want to test whether its population proportion is equal to a specific value. We declare the null hypothesis:

$$H_0 : p = p_0$$

Here, p_0 is a predefined number, for which we let a value of 0.5 based on the sample proportion. Depending on the type of test, i.e., one-sided or two-sided test, we have the following alternative hypotheses:

$$H_a : p \neq p_0 \text{ (Two-sided test)}$$
$$H_a : p \geq p_0 \text{ or } p \leq p_0 \text{ (One-sided test)}$$

The one-sample z-test is calculated using the following formula:

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

where $\hat{p}$ is the sample proportion, n is the number of samples. The null distribution for this test is Gaussian distribution because it can be used to approximate the binomial distribution due to the Central Limit Theorem.

Two sample z-test proportions We also utilized the two-sample proportion z-test, which tests for proportion equality between 2 groups of a binary variable. We have the following hypotheses:

$$H_0 : p_1 - p_2 = 0$$
$$H_a : p_1 - p_2 \neq 0 \text{ (Two-sided test)}$$
$$H_a : p_1 - p_2 \leq 0 \text{ or } p_1 - p_2 \geq 0 \text{ (One-sided test)}$$

where p_1, p_2 are the proportions of positive samples in groups 1 and 2, respectively. The test is calculated by:

$$z = \frac{\hat{p_1} - \hat{p_2}}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}}$$

The p-value and critical value are calculated based on the Gaussian distribution, similar to the one-sample z-test.

For continuous features, we used the Shapiro-Wilk normality test and the Mann-Whitney U test.

Shapiro-Wilk normality test Given a sample of continuous values $X = (x_1, x_2, ...x_n)$, the Shapiro-Wilk test [14] is used to determine whether the sample follows normal distribution, that is:

$$H_0 : \text{X is normally distributed} H_a : \text{X is not normally distributed}$$

The test statistic is calculated as:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

where $\bar{x}$ is the sample mean, $x_{(i)}$ is the ith order statistics (i.e., the ith smallest number in the sample. a_i is calculated by:

$$\boldsymbol{a} = [a_1, ..., a_n]^T = \frac{\boldsymbol{m}^T \boldsymbol{V}^{-1}}{\sqrt{\boldsymbol{m}^T \boldsymbol{V}^{-1} \boldsymbol{V}^{-1} \boldsymbol{m}}}$$

where $\boldsymbol{m} = [m_1, ..., m_n]^T$ is the expected values vector of order statistics, and $\boldsymbol{V}$ is the corresponding covariance matrix. The test is implemented using [13] to deal with the large number of samples.

Mann-Whitney U test Mann-Whitney U test [8] is a non-parametric test that examines the distributions of a features between 2 groups. The reason why we used non-parametric tests was because of the result of the Shapiro-Wilk test, which will be discussed further in Section 3. Formally, the hypothesis is defined as:

$$H_0 : \text{The distributions of 2 groups are identical}$$
$$H_a : \text{The distributions of 2 groups are not identical}$$

The U test is formulated as:

$$U = min(U_1, U_2)$$

where:

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$$
$$U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2$$

R_1, R_2 are the sum of ranks in group 1 and 2. Rank is a value assigned to each sample such that the smallest sample has rank 1 and the largest sample has rank n.

2.2.3 Machine Learning

A pre-processing step is crucial before fitting the data to a prediction model. Our data set suffers from the problem of severe missing values as almost every sample has null values for some features. To cope with this, we implemented KNN-impute algorithm, which estimates missing values based on the values of the k most similar samples (neighbors) in the dataset. It first identify the samples with null feature, then locate its neighbors (i.e. samples without missing value) using Euclidean distance. Finally, it replaces the missing value by taking the average of that feature from the neighboring samples. By leveraging the local structure of the data, this method preserves relationships between variables more effectively than mass imputation using using mean or median value.

For the task of binary classification, we employed the Extreme Gradient Boosting model, commonly referred to as XGBoost, a powerful gradient boosting and decision-tree-based ensemble machine learning algorithm introduced by Chen et al. [1] Its ability to handle both numerical and categorical data types, as well as to leverage a large collection of features, makes it ideal for our dataset. Its underlying principle involves the aggregation of predictions from many weak decision tree models, leading to the creation of a more accurate and robust one. The objective of our model is to minimise the loss function for each round of boosting t, expressed as followed:

$$Objective(t) = \mathcal{L}^{(t)}(y, \hat{y}) + \Omega(f_t)$$

$\mathcal{L}$ represents the loss function (i.e binary cross entropy or log loss) and can be calculated as

$$\mathcal{L}^{(t)}(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

We split the data into training and testing sets with a ratio of 80:20 and let the boosting process run for 100 rounds. After hyperparameter tuning, we found the best configuration for our model: Maximun depth of each tree is set to 14 and sub-sample ratio is 80%. A split is only valid if reduced loss is 1e-6 or above and the weight of child node is higher than 7. We also applied a regularisation term $\Omega(f_t)$ that is based on Lasso regularisation (L1). This is mathematically described as

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} |w_j|$$

where T stands for the number of leaf in each tree. The parameter λ, which controls the strenth of regularisation, is assigned as 0.1. The addition of regular- isation term helps to smooth the final learned weights to avoid over-fitting.

ROC Curve To evaluate the effectiveness of our classification model, we used Receiver Operating Characteristics (ROC) curve analysis, a widely used graphical method for calculating the trade-off between the true positive rate (sensitivity) and false positive rate (1-specificity) in various threshold settings. This approach is particularly useful for binary classification problems, especially when class distributions are not balanced or misclassification costs vary. We define sensitivity, or True Positive Rate (TPR), according to Fawcett (2006) [2] as follows:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

False Positive Rate (FPR) as:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

The ROC curve is constructed by varying the decision threshold of the classifier and computing the corresponding TPR and FPR values. The area under the ROC curve (AUC) provides a single scalar value summarizing the overall performance of the model. An AUC of 0.5 indicates random performance, while an AUC of 1.0 corresponds to a perfect classifier.

Area Under the Curve (AUC) served to measure discriminative capacity through the pROC package in R according to Robin et al. (2011)[12]. The pROC package uses trapezoidal computation for numerical integration.

The AUC is determined through the trapezoidal rule as it measures the areas formed below the ROC curve with trapezoidal sections. The algorithm determines the areas under straight lines which connect successive points appearing on the ROC curve. Since its implementation helps save time while being widely used within computational statistics for numerical integration this method serves as a popular approach.

The AUC estimation method exists within both the pROC package of R and the sklearn.metrics.auc module of Python making them popular among users. The method proves useful for measuring performance in binary classification settings due to its reliability.

SHAP-Based Feature Attribution To interpret the predictions of our XGBoost model, we employed SHapley Additive exPlanations (SHAP), a game-theoretic framework designed to fairly allocate the contribution of each input feature to the model's output. SHAP values are derived from Shapley values in cooperative game theory, which provide a mathematically principled way of attributing the output of a model to its input features [7].

Given a predictive model $f(x)$, where x is a vector of input features, the SHAP value for a feature x_i is computed as:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{i\}) - f(S) \right] \tag{1}$$

where:

- N is the set of all features,
- S is any subset of N that does not include x_i,
- $f(S)$ represents the model's output when only the features in S are included,
- ϕ_i is the SHAP value for feature x_i, representing its average marginal contribution across all possible feature subsets.

The computation of SHAP values considers all possible feature combinations, making it a fair and consistent method for feature attribution. However, due to the factorial complexity of the exact computation, efficient approximation methods such as TreeSHAP have been developed for tree-based models [6].

SHAP values satisfy the following important properties:

- **Efficiency:** The total sum of SHAP values across all features equals the difference between the model output and the expected value of the output.

- **Symmetry:** Features that contribute equally to every possible subset receive the same SHAP value.
- **Dummy Property:** A feature that does not change the model's prediction in any subset has a SHAP value of zero.
- **Additivity:** SHAP values can be combined across different models in an ensemble setting.

The total prediction for an instance x can be expressed as:

$$f(x) = E[f(x)] + \sum_{i=1}^{n} \phi_i \tag{2}$$

where $E[f(x)]$ is the expected model output over the dataset, and $\sum_{i=1}^{n} \phi_i$ represents the total contribution from all features. This decomposition enables the interpretability of individual predictions.

To quantify global feature importance, we compute the mean absolute SHAP values across all instances:

$$\text{Feature Importance} = \frac{1}{m} \sum_{j=1}^{m} |\phi_{ij}| \tag{3}$$

where m is the number of instances, and ϕ_{ij} represents the SHAP value of feature i for instance j. Features with higher mean absolute SHAP values are considered more influential in the model's decision-making process.

By leveraging SHAP values, we gain a comprehensive understanding of both global and local model behavior. This allows us to ensure model transparency, identify key features, and improve trust in model predictions.

3 Results

3.1 Distribution Analysis

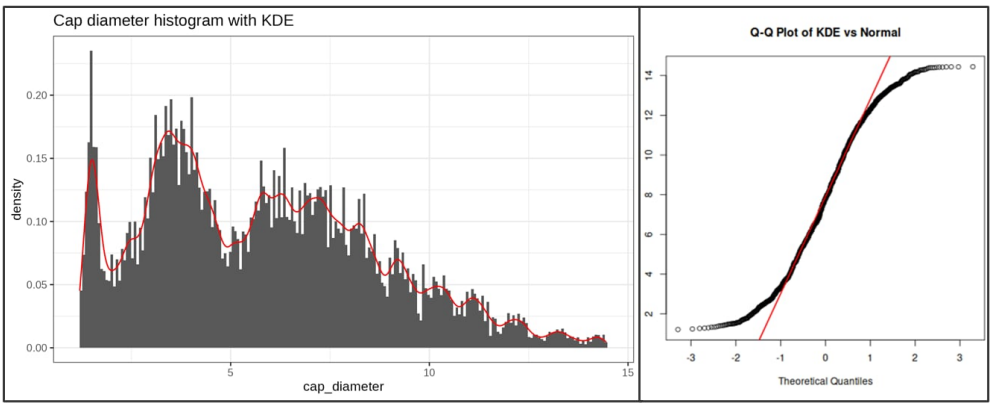

Figure 8: Cap Diameter Distribution

The kernel density estimate (KDE) suggests that the distribution of cap diameters is multimodal, implying the presence of multiple underlying subpopulations. The density of the data is concentrated between 2 and 6 units, with several peaks visible, which points to variation in cap size. Additionally, there is a noticeable right-skewed tail, indicating that while most mushrooms have smaller cap diameters, there are a few specimens with much larger caps. The Q-Q plot further supports the finding that the data does not follow a normal distribution. The points on the plot deviate significantly from the red reference line, particularly in the tails, signaling a lack of normality. The S-shape of the Q-Q plot highlights the skewness and multimodal nature of the distribution, aligning with the insights gathered from the histogram.

The kernel density estimate indicates that the distribution of stem heights is right-skewed, with the majority of stems clustering around lower values and a gradual decline as the values increase. Among the fitted distributions, the log-normal (green) and gamma (red) distributions appear to fit the data much better than the normal distribution (blue), suggesting that stem

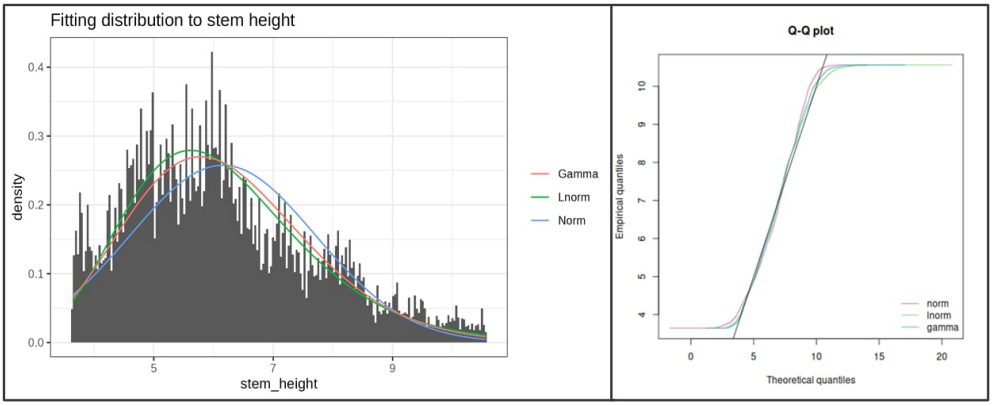

Figure 9: Stem Height Distribution

height does not follow a normal distribution. The Q-Q plot, shown in the right panel, compares the empirical quantiles of the data to those of the theoretical distributions. The normal distribution (pink) deviates significantly, particularly in the tails, which reinforces the idea that stem height does not follow a normal distribution. In contrast, the log-normal and gamma distributions closely match the empirical data, confirming that they provide a better fit.

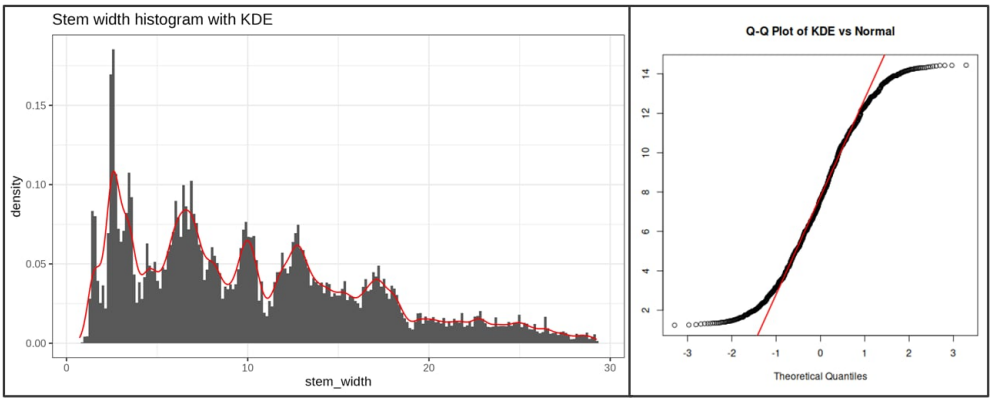

Figure 10: Stem Width Distribution

The kernel density estimate reveals that the distribution is multimodal, with several peaks suggesting the presence of potential subpopulations or clustered data. Additionally, the data shows right-skewness, with a longer tail extending toward higher values. The visible variations in density may point to outliers or distinct groupings within the data. The Q-Q plot indicates that, while most of the data points follow a straight line, there are noticeable deviations at both the lower and upper extremes. This curvature at the ends suggests that the data is not normally distributed, displaying skewness and possible outliers. The right tail, which is evident in the histogram, is further confirmed in the Q-Q plot, as points deviate from the red reference line in the upper quantiles.

3.2 Hypothesis Testing

We began testing on the 2 binary features (`class` and `does-bruise-or-bleed`). We were first interested in the true population proportion of poisonous mushrooms. Using the right-sided one-sample z-test with $\hat{p} = 0.5$, we obtained p-value ≈ 0, so we can conclude that the proportion of poisonous mushrooms is greater than 0.5, with 95% confidence interval = [0.519, 0.52]. We were also interested in whether the proportion of mushrooms having bruise or bleed on interactions is significantly different in 2 types of mushrooms. Applying the two-sample z-test for proportions, we obtained p-value ≈ 0, so we conclude that they are different, with an estimated proportion in the poisonous group = 0.18 and edible group = 0.196. That is, if a mushroom has bruise or bleed on interaction, it is more likely to be edible, though the probability is not significantly greater.

For continuous features (`cap-diameter`, `stem-height`, `stem-width`), it is of great importance to know if the sample follows the normal distribution in order to choose parametric or

non-parametric tests for further analysis. As described in Section 2.2.2, we utilized the Shapiro-Wilk test and all three have p-values ≈ 0, that is they do not follow the normal distribution. Thus, we only used non-parametric test, specifically the Mann-Whitney test. the p-value for all three features are approximately equal to 0, thus, we conclude that the distributions of the numerical features are different in both poisonous and edible mushrooms.

3.3 Model Evaluation

In this section, we compare the performance of our dataset across three different machine learning models: Random Forest, Logistic Regression, and XGBoost. As shown in Table 2, XGBoost outperforms the other two models, particularly in terms of accuracy, achieving an impressive 99.11%. This result highlights XGBoost's effectiveness in capturing patterns and making precise predictions compared to Random Forest and Logistic Regression.

While Random Forest and Logistic Regression also provide reliable results, their performance is slightly lower than XGBoost, which benefits from its advanced boosting techniques and ability to handle complex relationships within the data. The superior accuracy of XGBoost suggests that it is the most suitable model for our dataset, making it a strong candidate for practical implementation.

To evaluate the performance of the XGBoost model, we first used the ROC curve, which illustrates the trade-off between the True Positive Rate and the False Positive Rate at a threshold of 0.5. The model achieved an AUC of 0.911, demonstrating excellent discriminatory power. The ROC curve, shown in Figure 11, visually supports this.

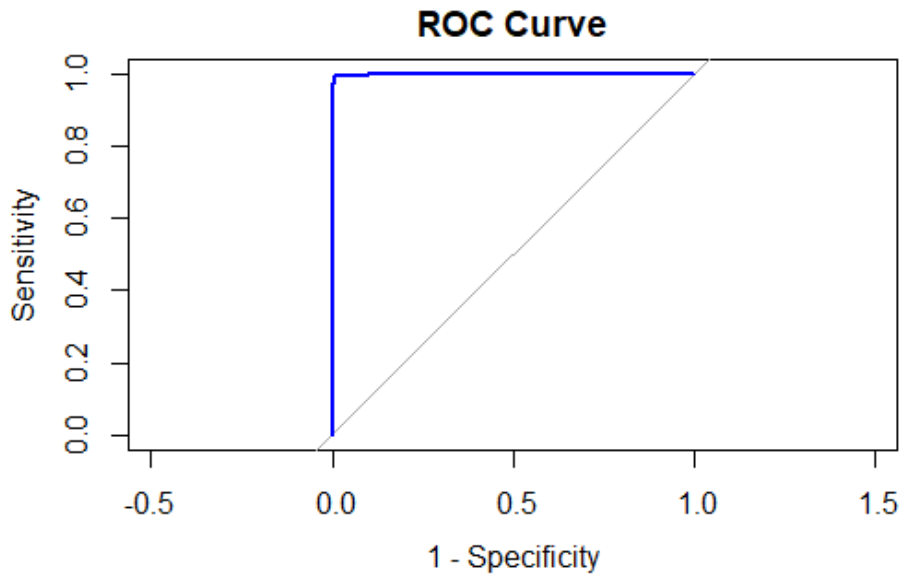

Figure 11: ROC curve

Table 3 compares the results of our proposed approach with other approaches through accuracy, precision, recall, and F1 score.

Approach	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Random Forest	91.2	89.6	92.4	91.0
Logistic Regression	86.4	85.2	84.0	84.6
XGBoost	**99.11**	**99.28**	**99.10**	**99.19**

Table 3: Performance comparison of different classification models.

Figure 12 describes the SHAP (SHapley Additive exPlanations) feature importance values for a machine learning model. Each bar represents the mean SHAP value of a feature, showing how much that feature contributes to the model's output across all predictions.

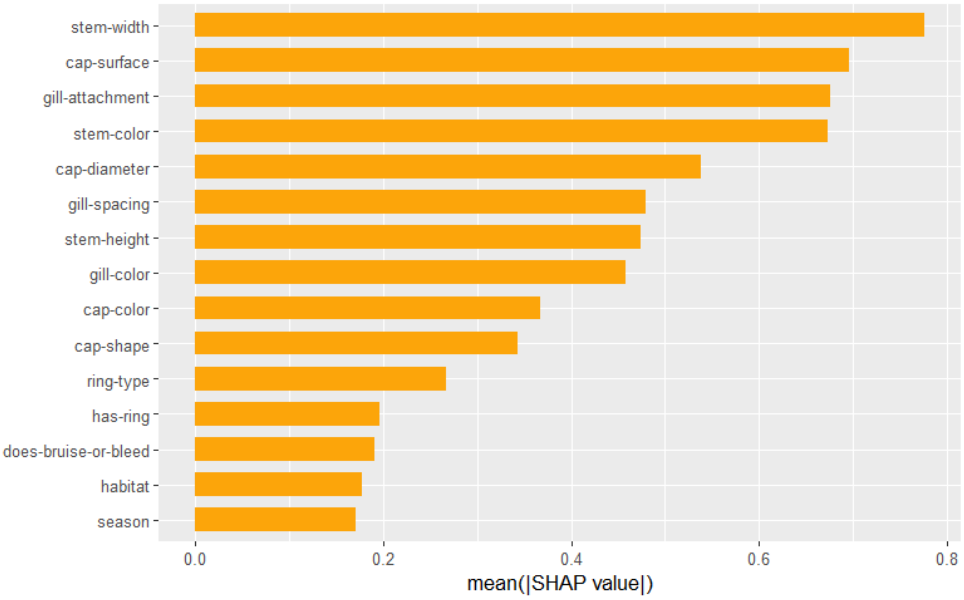

Figure 12: Mean feature importance

Figure 13 and 14 display two SHAP waterfall plots, which visualize how individual features contribute to a model's prediction for a specific data instance. In Figure 13, the base value (expected prediction) is approximately 0.415, and the final prediction for the instance is -5.89, indicating a strong negative classification. The most influential features that decrease the prediction are cap-surface, stem-color, and stem-width, each contributing negatively with SHAP values of -1.22, -1.14, and -0.95, respectively.

In contrast, Figure 14 reflects a very different case where the prediction is pushed upward to 6.33, starting again from a similar base value of 0.415.

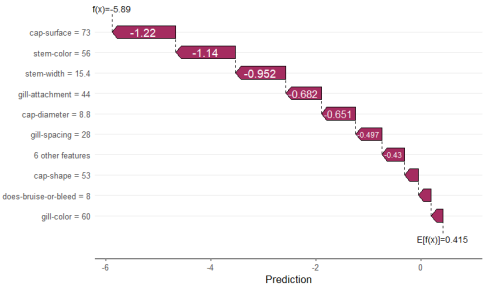

Figure 13: Feature Importance for Poisonous class

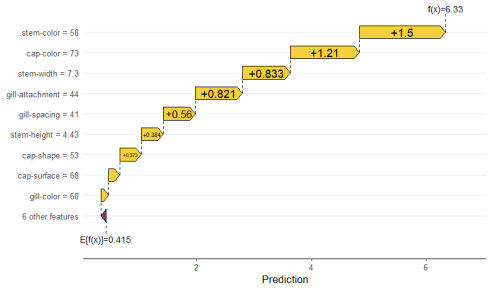

Figure 14: Feature Importance for Edible class

4 Conclusion and Discussion

In this project, we made a full data analysis of the mushroom dataset, including exploratory data analysis, distribution analysis, hypothesis testing and machine learning. We successfully extracted non-trivial information from the dataset and also built a model for predicting whether a mushroom is poisonous or edible. Since the dataset contains many features, future works should focus on further analysis of those that we have not investigated yet, apply multivariate data analysis methods and utilize deep learning models for prediction.

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

[2] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[3] László G Nagy, Renáta Tóth, Enikő Kiss, Jason Slot, Attila Gácser, and Gábor M Kovács. Six key traits of fungi: Their evolutionary origins and genetic bases. *Microbiology Spectrum*, 5(4), July 2017.

[4] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[5] Kaggle. Playground series - season 4, episode 8. https://www.kaggle.com/competitions/playground-series-s4e8/overview.

[6] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Andrew DeGrave, Joshua M. Prutkin, Bharat Nair, Robert Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, January 2020.

[7] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 4766–4777, Long Beach, CA, USA, December 2017.

[8] Donald R. Mann, Henry B.; Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50—-60, 1947.

[9] Xotic Mushrooms. Differentiating poisonous and edible mushrooms: A comprehensive guide, 2023. Accessed: 2025-04-04.

[10] Thomas N. Sherratt, David M. Wilkinson, and Roderick S. Bain. Explaining dioscorides' "double difference": Why are some mushrooms poisonous, and do they signal their unprofitability? *The American Naturalist*, 166(6):767–775, 2005.

[11] Roger Phillips. *Mushrooms: A comprehensive guide to mushroom identification.* Pan Macmillan, 2011.

[12] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12(1):77, 2011.

[13] Patrick Royston. An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics*, 31:115–124, 1982.

[14] M. B. Shapiro, S. S.; Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, pages 591–611, 1965.