Mushroom Poisonousness Analysis

Final project Report
DSI Linzhuo Zhang
Github: zlzdeGH/DATA1030_PROJECT

Introduction (5 points)

Purpose

Identifying whether a mushroom is edible or poisonous is critical for consumer health and foraging safety, with a reliable classification model potentially reducing the risk of poisoning and offering guidance to ensure the informed decision-making of agricultural stakeholders, including mushroom foragers and food safety inspectors. While manual identification methods may be time-consuming and prone to human error, automated classification techniques that leverage detailed morphological characteristics are cost-effective and highly accurate.

UCI Secondary Mushroom Dataset (UCI SMD)

61,069 mushroom samples from 173 species are listed in the UCI Secondary Mushroom Dataset (UCI SMD), and 20 variables are used to describe each mushroom. These include 17 categorical features, such as cap shape, gill characteristics, stem attributes, and ring type, and 3 continuous features, such as cap diameter, stem height, and stem width, which derive from field observations and measurements taken in different seasons and habitats, and where the edibility of a mushroom is uncertain, it is labelled as poisonous for safety reasons.

Target Variable and Features

A classification problem is formed by this dataset, namely where the target variable demonstrates whether mushrooms are edible or poisonous, with ‘edible’ encoded as ‘0’ and ‘poisonous’ as ‘1’ after preprocessing.

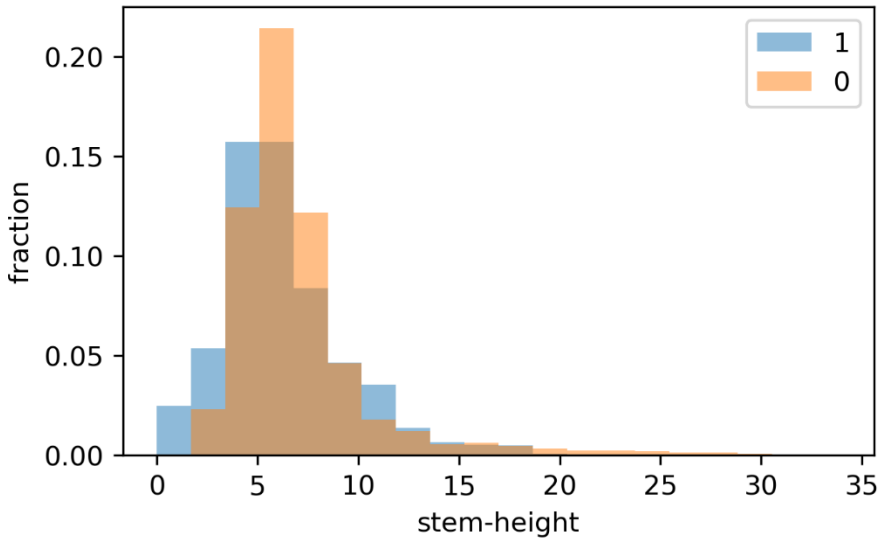
The features recorded include morphological descriptions, such as cap diameter and stem width, as well as categorical characteristics, including veil colour and ring type. To retain valuable information, missing values in categorical variables are categorised as “Missing” as opposed to being removed or arbitrarily imputed.

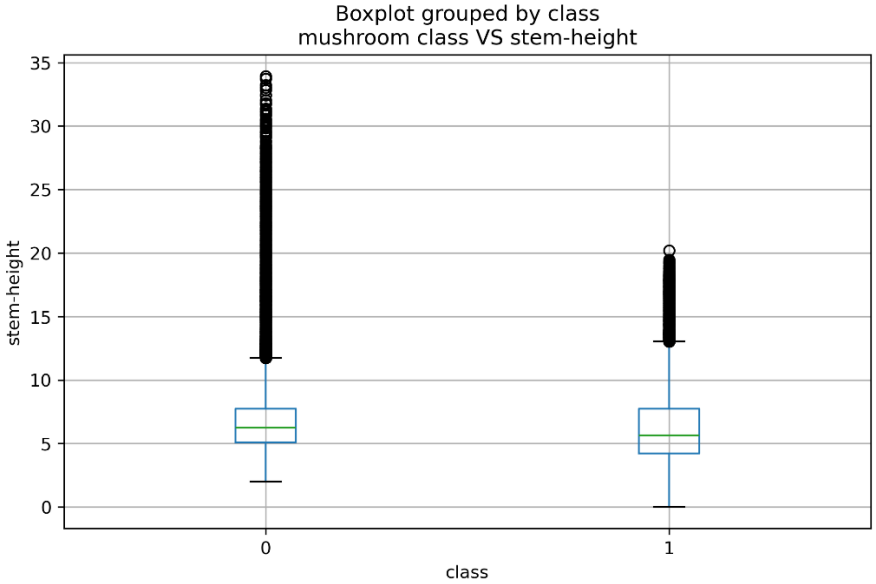
Previous Work

Recent mushroom classification research has typically utilised machine learning models, which include logistic regression, random forests, and XGBoost methods, and this approach has tended to achieve high accuracy levels, which indicates that well-tuned models can make almost perfect predictions in relation to simpler mushroom datasets from the UCI repository. Classical research on similar datasets has found model accuracies above 99%, which evidences the robust signals in mushroom morphology and spore characteristics. These studies indicate that advanced models and diligent feature engineering can support safer identification of edible mushrooms, reducing the risk of inadvertent poisoning.

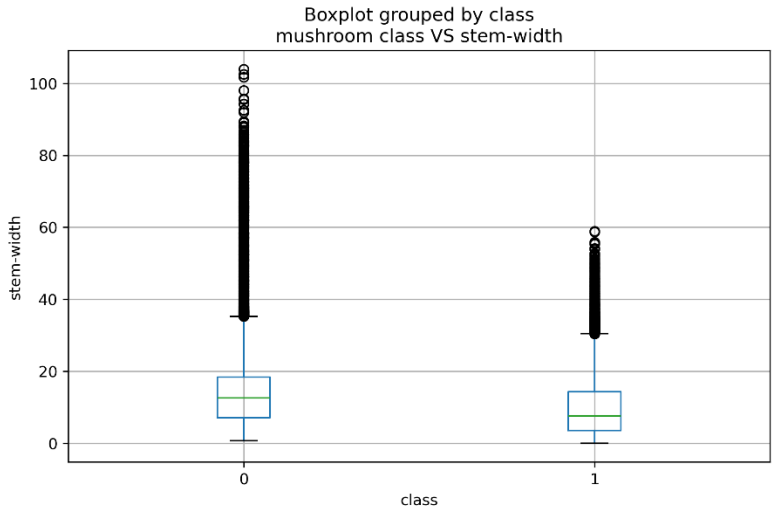
Explanatory Data Analysis

This study’s Exploratory Data Analysis (EDA) sought to identify mushroom edibility patterns, distributions, and potential predictors, with the following initial findings:

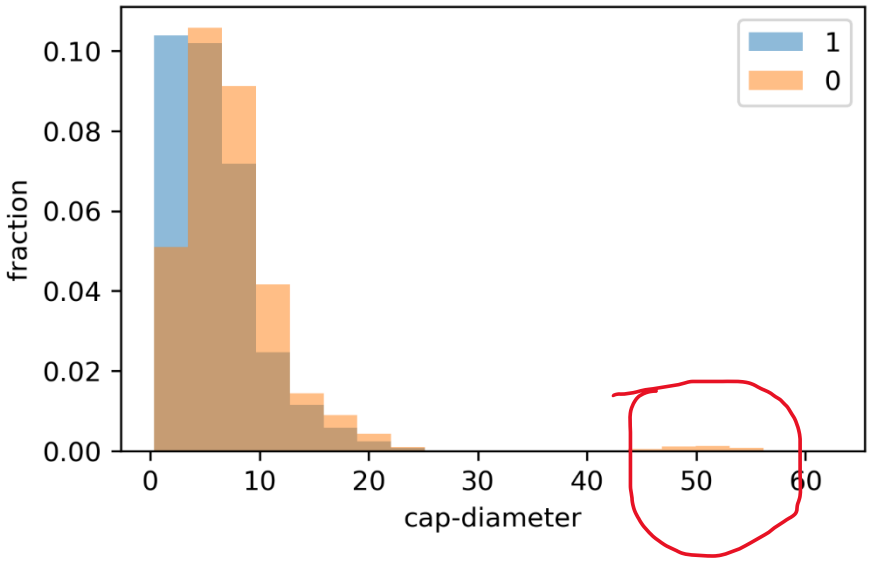


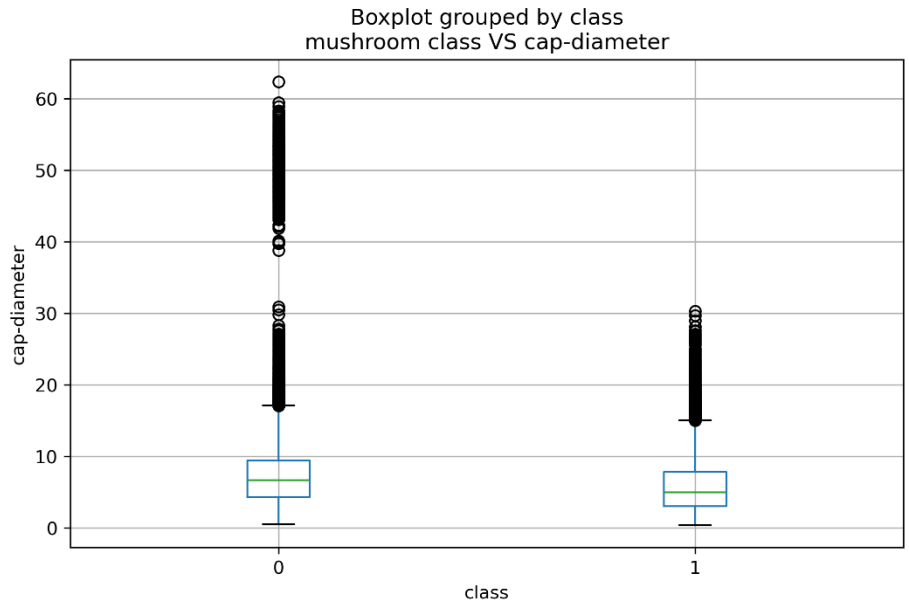


- **Height Distribution:** For poisonous and edible mushrooms, the height distribution overlapped substantially, yet a subset of particularly tall mushrooms was edible only.

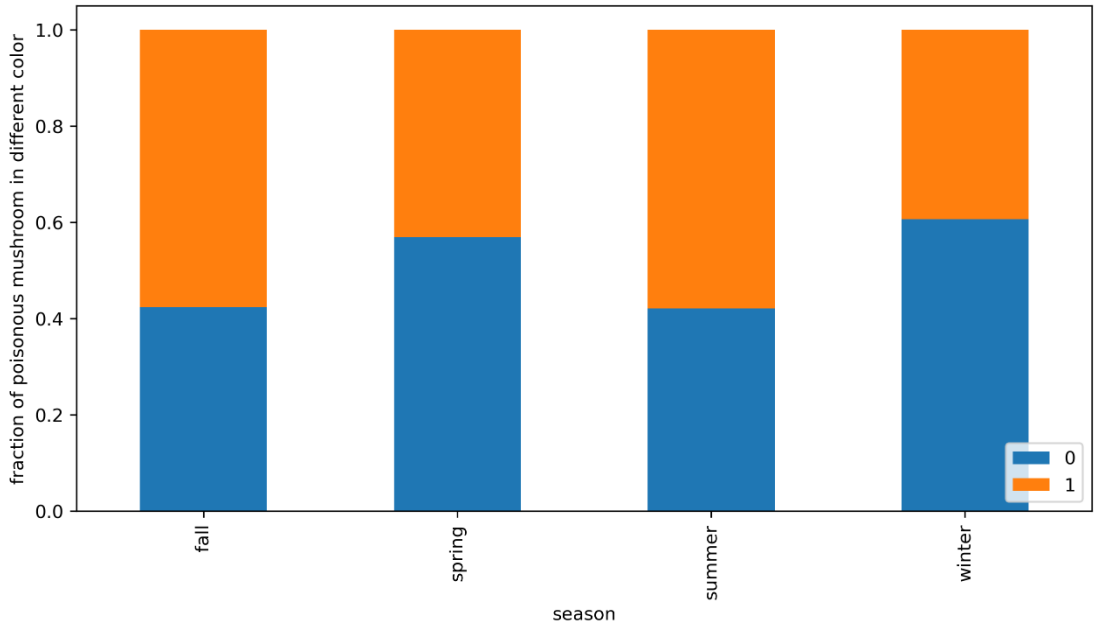


- **Stem Width:** The average stem width of poisonous mushrooms tended to be lower, while a close association was found between wider stems and edibility, and wide-stemmed mushrooms were all edible.





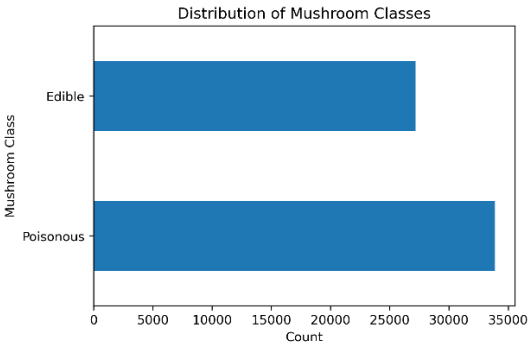
- **Cap Diameter Trends:** Smaller caps were more frequently poisonous, while larger caps generally indicated edibility.

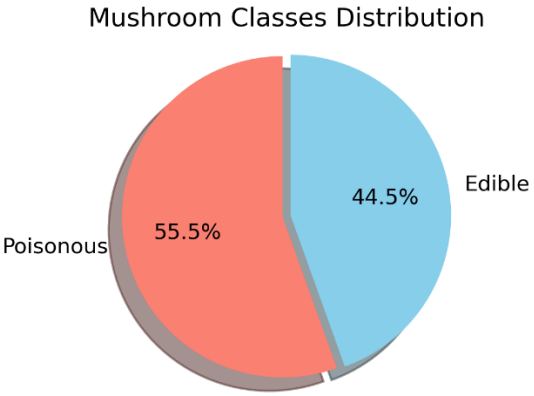


- **Seasonal Variation:** Poisonous mushrooms were more commonplace in summer and autumn, while fewer poisonous varieties were found in spring and winter, with this seasonal trend potentially reflecting environmental conditions which are favorable to particular species.

Target Variable: This dataset's target variable was 'class', and each mushroom data was classified to groups 'p' or 'e', indicating poisonous and edible.

In addition, my dataset is balanced and independent and identically distributed (i.i.d) and I use accuracy as the baseline evaluation metric. Therefore, the baseline score is 55.5%





Methods

Splitting Strategy:

To evaluate most models (Logistic Regression, Random Forest, KNN, SVM models) in this study, the dataset was separated into an 80% training and 20% test set, and to estimate model performance K-Fold Cross-Validation (K=4) was applied on the other portion. Each fold provided a 75%-25% training-validation split, ensuring that the sizes for validation and test were identical (20%), thereby mitigating overfitting and providing a more reliable measure of generalization performance.

For the XGBoost model, two regular splitting strategies were used, and the dataset was split into train, validation, and test set, with a ratio of 6:2:2.

Data Preprocessing:

There were 17 features, and 9 were missing values, while three features were not missing any values.

Missing values in the data set were encoded as a distinct category ("Missing"), rather than discarding them or imputing with statistical measures, as this approach preserves signals which may be informative.

All categorical features were one-hot encoded, and continuous variables were scaled by Standard Scaler ().

ML Pipeline:

Five different machine learning algorithms were tested, including linear and non-linear models:

model	parameters	Best parameters
Logistic Regression	C: 0.01, 0.1, 1.0, 10, 100	C: 100
Random Forest	max_depth: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 max_features: sqrt, log2, None	max_depth: 10 max_features: sqrt
KNN	n_neighbors: 1, 5, 10, 20, 30, 100 weights: uniform, distance	n_neighbors: 5 weights: uniform
SVM	gamma: 1e-3, 1e-1, 1e1, 1e3, 1e5 C: 1e-2, 1e-1, 1e0, 1e1, 1e2	gamma: 1e1 C: 1e-1
XGBoost	max_depth: 1,3,10,30,100 reg_alpha: 0e0,1e-2, 1e-1, 1e0, 1e1, 1e2 reg_lambda: 0e0,1e-2, 1e-1, 1e0, 1e1, 1e2	max_depth: 30 reg_alpha: 1e0 reg_lambda: 0e0

Evaluation Metrics:

Five multiple random states (seeds) were introduced when performing the train-test split and K-fold procedure, and all models, including Logistic Regression, Random Forest, XGBoosting, and SVM, KNN were trained and evaluated across these seeds, which revealed the models' sensitivity to variations in data partitioning.

After the runs for all candidate models under various seeds were completed, the highest performing parameter and its corresponding model and test scores were recorded. Carrying out a comparison of scores from multiple runs enabled the most stable and reliable model to be identified.

Accuracy was used as the primary metric to ensure consistency with previous mushroom datasets research. While accuracy is intuitive, clearly understood, and is in alignment with safety objectives given that misclassifying a poisonous mushroom as edible is a critical error, the confusion matrix was employed to closely examine false negatives.

Results

Baseline Performance:

An accuracy of approximately 55.5% was achieved by using a naive baseline, namely classifying all mushrooms as the majority class, hence it corresponds to the inherent class distribution (33,888 poisonous versus 27,181 edible samples).

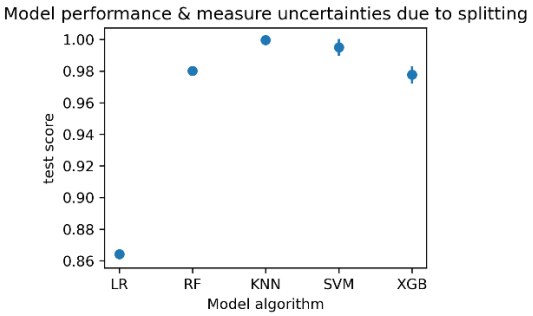
This mushroom dataset's baseline accuracy was approximately 0.5549, which was found by dividing the number of poisonous samples by the total number of mushrooms, and all evaluated models, including Logistic Regression, Random Forest, K Neighbors, XGBoosting, and SVM, outperformed this baseline significantly. The Logistic Regression model yielded the lowest mean accuracy, while the K Neighbours model achieved the highest performance, and an accuracy of almost 100% was attained through the best Gradient Boosting model, thereby accurately classifying most edible and poisonous samples and surpassing the baseline and other evaluated models.

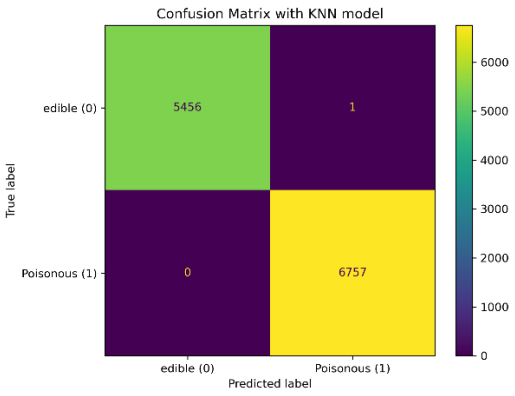
Model Performance:

All tested models outperformed the baseline considerably, with the table below summarising performance (mean accuracy ± standard deviation over 4-fold CV and multiple seeds):

Model	Accuracy (Mean ± Std)
Logistic Regression	~86.4% ± 0.00196%
Random Forest	~97.8% ± 0.00177%
KNN	~99.9% ± 0.00013%
SVM (RBF Kernel)	~99.5% ± 0.00528%
XGBoost	~97.8% ± 0.00556%

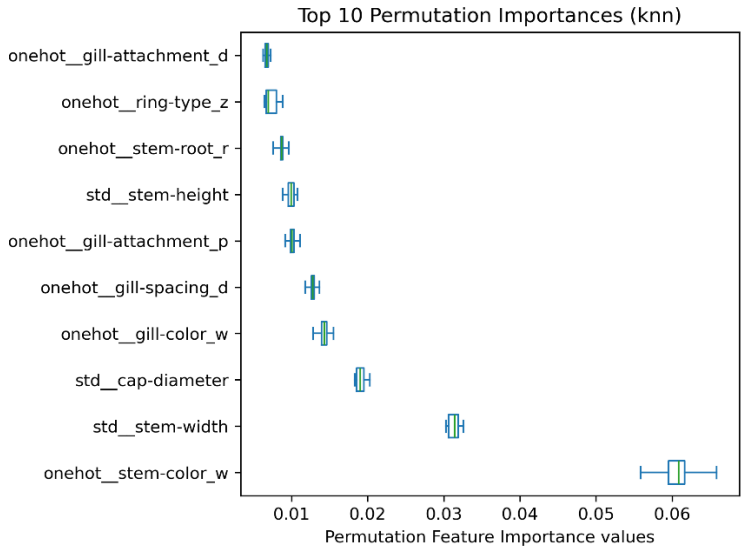
K Neighbours was identified as the top performer, with around 99.9% accuracy, which is roughly (99.9% - 55.5%) = 44.4 percentage points above the baseline, as well as being comfortably within expectations given existing research on similar mushroom classification tasks. Therefore, I used it in confusion matrix and feature Importance finding.



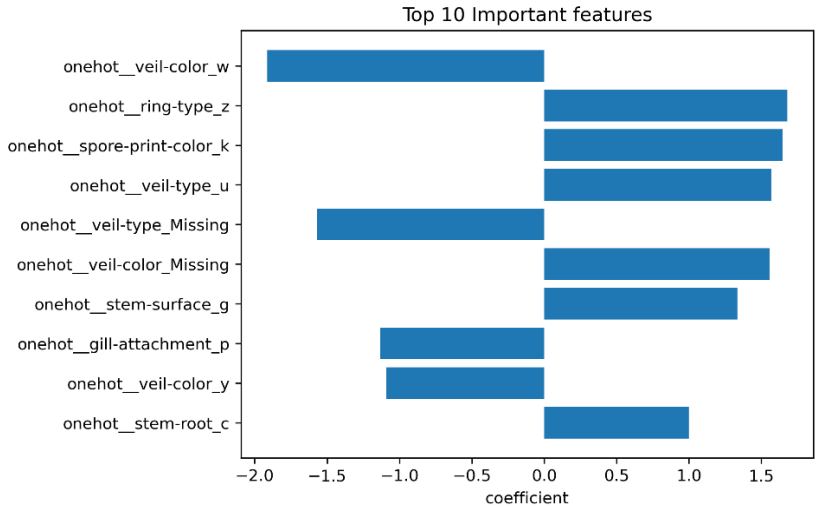


Global Feature Importance:

Global feature importance was ascertained using several methods:

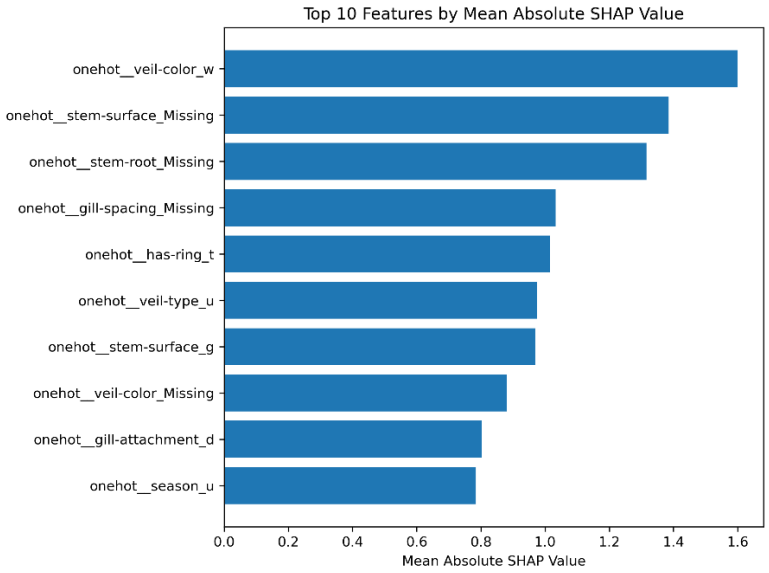


The first plot shows feature importance using the Permutation Importance method, which evaluates the drop in model performance when a feature's values are randomly shuffled. This method is applied to a KNN (K-Nearest Neighbors) model and highlights the most influential features based on how much their absence affects the model's predictions. The top-ranked features, such as onehot__gill-attachment_d and onehot__ring-type_z, demonstrate the largest performance drops, indicating their critical role in the model.



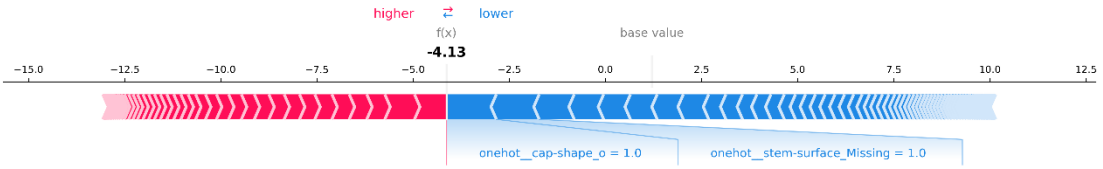
The second plot shows the top 10 important features determined by their coefficients in a Logistic Regression model. Coefficients in this context indicate the strength and direction of the relationship between each feature and the target variable. Positive coefficients suggest that the corresponding feature has a positive correlation with the target, meaning as the feature value increases, the likelihood of a positive outcome increases. For instance, onehot__ring-type_z and onehot__spore-print-color_k have strong positive coefficients, indicating they contribute positively to predictions. On the other hand, negative coefficients

signify a negative correlation, where higher values of the feature decrease the likelihood of the target outcome. .

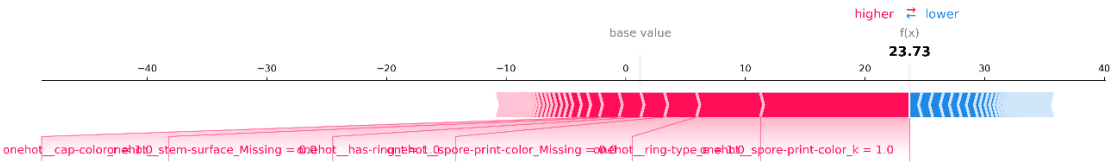


The third plot uses SHAP values to display feature importance. In this plot, onehot__veil-color_w emerges as the most influential feature, contributing significantly to the model's predictions. Features like onehot__stem-surface_Missing and onehot__stem-root_Missing also show notable importance, highlighting their relevance.

Local Feature Importance (SHAP values):



<Shap local value for index 0 >



<Shap local value for index 200 >

The true prediction for index 0 is class 0, and for index 200 it is class 1.

For index 0, the features onehot__cap-shape_o = 1.0 and onehot__stem-surface_Missing = 1.0 push the prediction lower. Their SHAP values pull the final prediction further to the left (edible).

For index 200, onehot__cap-color_w and onehot__stem-surface_Missing are the features that push the prediction strongly to the right(poisonous).

Interpretation:

These findings illustrate the nuanced interplay of morphological and categorical features in determining edibility. Moreover, missing values were shown to have a significant influence, which implies that these "gaps" in data may indicate inherent challenges in measuring certain characteristics of poisonous species.

Outlook

While the models were found to perform well, several enhancements could be made:

- **Advanced Algorithms:** Carrying out more advanced tree-based methods, such as Deep Learning or LightGBM, may uncover more complex interactions between features.
- **Feature Engineering:** Incorporating interaction terms or domain-specific features, such as soil composition, precise habitat microclimates, may refine predictions.
- **Interpretability Tools:** Developing user-friendly dashboards, such as interactive SHAP plots, may be helpful for non-technical stakeholders, including foragers or agricultural inspectors, to comprehend model decisions in real-time.
- **PCA:** Use dimensionality reduction methods (e.g., PCA) to map high-dimensional feature spaces into more intuitive, low-dimensional representations

Reference

[1]Mushroom data creation, curation, and simulation to support classification tasks, Dennis Wagner, D. Heider, Georges Hattab, Published in 14 April 2021 DOI: 10.1038/s41598-021-87602-3
[2]Secondary Mushroom ,Donated on 8/13/2023 DOI: 10.24432/C5FP5Q

[3] Estimating Risk and Uncertainty in Deep Reinforcement Learning
W. Clements, Benoît-Marie Robaglia, Published in arXiv.org 23 May 2019
https://www.semanticscholar.org/paper/Estimating-Risk-and-Uncertainty-in-Deep-Learning-Clements-Robaglia/d6285ff3dcb15c1da84dcbc98141be10ef0e8dd1
[4] Bayesian Learning of Neural Network Architectures
G. Dikov, Patrick van der Smagt, Justin Bayer,Published on 14 January 2019
https://www.semanticscholar.org/paper/Bayesian-Learning-of-Neural-Network-Architectures-Dikov-Smagt/bd862cb691e6db5faba96ebab3e25c7ffdc9abf5