Mushroom Edibility Classification Data Anlysis

Yanfeiyun (Kathy) Wu

Data Science Institute, Brown University

https://github.com/kathywu1201/Mushroom_Edibility_Classification_ML_Project

December 10, 2023

1 Introduction

Mushroom poisoning has become a serious food safety issue in China. According to the investigation of China CDC, in 2022, there are 482 mushroom poisoning incidents involving 1,332 patients and 28 deaths, with a total case fatality rate of 2.1% [1]. In view of the extensive impact and harm of poisonous mushrooms on public health, it is necessary to promote prevention and most importantly improve the ability of professionals to identify and diagnose mushroom poisoning. In addition to the current existing mushroom species, it is possible that there might appear a new mushroom that was not recorded, how should we categorize this mushroom, edible or poisonous? As a result, in building up this Mushroom classification Machine Learning model, we can potentially reduce the risk of death and intoxication due to poisonous mushroom.

This Mushroom Edibility Classification Dataset [2] was found in Kaggle, collected from Patrick Hardin's Mushrooms & Toadstools, and inspired by Jeff Schlimmer's Mushroom Data Set . This dataset includes 61069 hypothetical mushrooms with caps based on 173 species where each mushroom is identified as definitely edible, definitely poisonous, or of unknown edibility which is classified as poisonous.

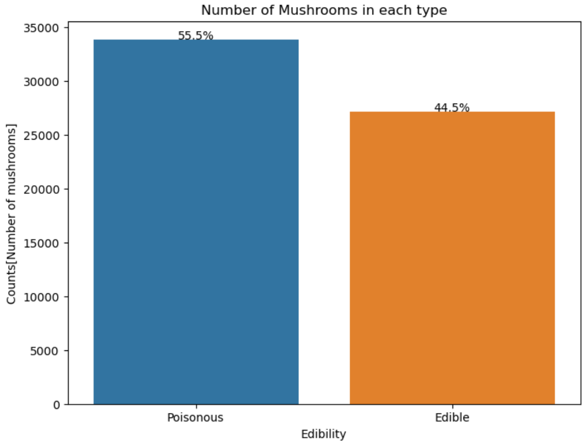
The Mushroom Edibility Classification Dataset is a classification which originally contains 61069 observations and 21 features including the target variable 'class.' In the 'class' variable, 'e' meaning edible is encoded as 0 and 'p' meaning poisonous is encoded as 1 in preprocessing.

In the recent two Kaggle projects centered on the Mushroom Edibility Classification Dataset, different machine learning models were employed to determine the edibility of mushrooms. Lucas Agra utilized a Random Forest Model, achieving remarkable predictive power with a 100% accuracy rate [4]. In addition, Levi Payne opted for the XGBoost model in their analysis of the same dataset, demonstrating a high level of accuracy with a score of 99.96% [3]. These results highlight the effectiveness of both Random Forest and XGBoost models in accurately classifying mushroom edibility.

2 Exploratory Data Analysis

In the initial phase of the project, I conducted a comprehensive exploratory data analysis (EDA) to gain valuable insights into the characteristics and patterns within the mushroom dataset. Starting with an analysis of the distribution of the target variable, Figure 1(a) reveals that the ratio of the target variable is approximately 6:4 for poisonous and edible instances, respectively.

When inspecting the missing values in the dataset, we found that among the 20 features, 9 categorical features contain missing values, with 5 features having more than 60% of missing values (see Figure 1(b)). However, we decided to keep all of them, as missing values in these features indicate that some mushrooms do not have certain features. Yet, this does not necessarily imply that an unseen mushroom lacks these features either.



(a) Distribution of target variables.

Feature (categorical features)	**Percentage of Missing**
cap-surface	0.231214
gill-attachment	0.161850
gill-spacing	0.410405
stem-root	0.843931
stem-surface	0.624277
veil-type	0.947977
veil-color	0.878613
ring-type	0.040462
spore-print-color	0.895954

(b) Percentage of missing value in each feature.

Figure 1: Target variable distribution & Percentage of missing value

According to the Boxplot of Cap Diameter Based on Different Cap Shape (see Figure 2), the bell cap shape mushroom overall have a small cap-diameter while the spherical cap shape mushroom relatively have a large cap-diameter.

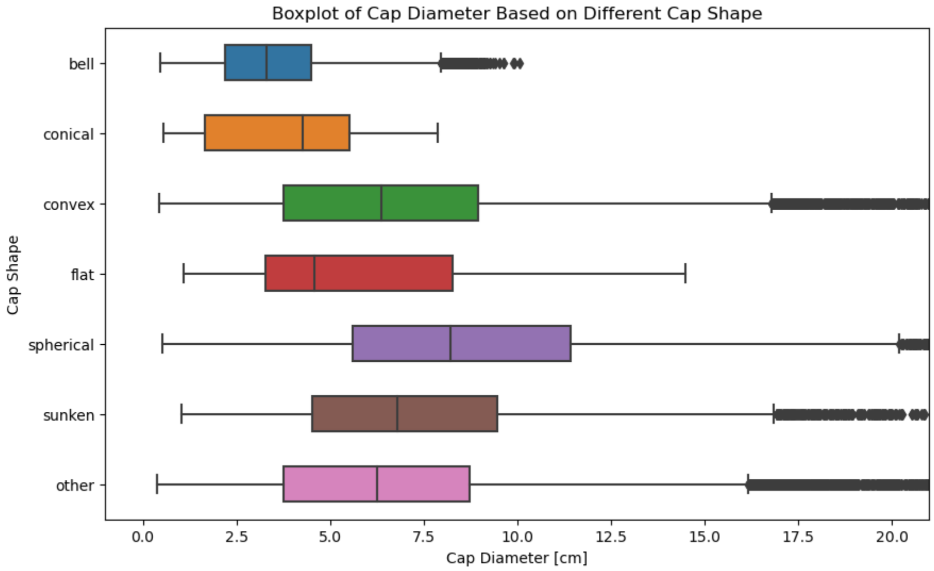


Figure 2: Boxplot of cap shape vs. cap diameter.

In this stacked bar plot calculating the percentage of edible and poisonous mushroom based on the habitat of mushroom (see Figure 3), is appears to have 100% of edible mushroom grown in urban and waste area, and 100% of poisonous mushroom grown in paths area according to the dataset.

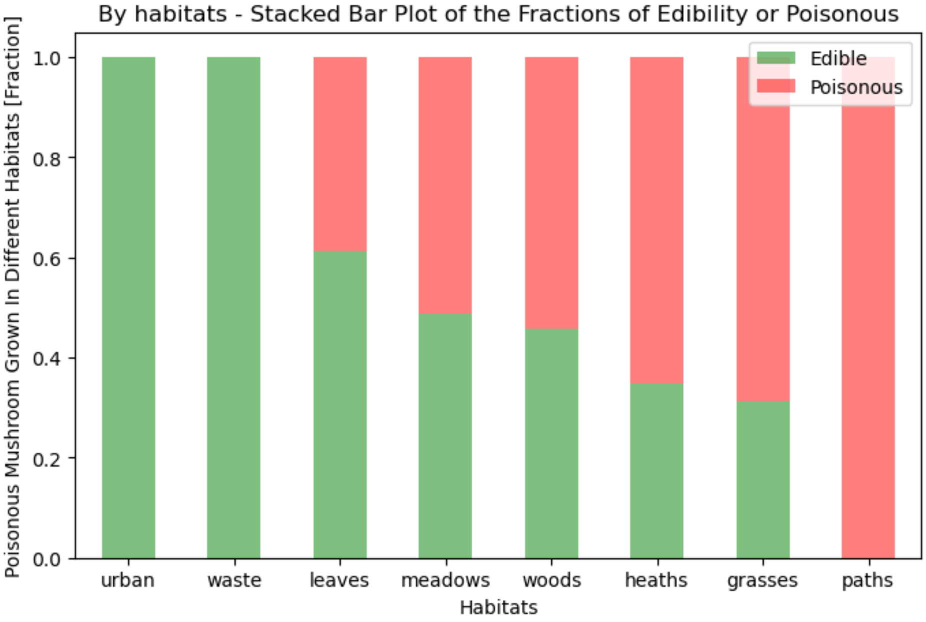


Figure 3: Stacked Bar Plot of mushroom edibility based on habitat.

This category specific histogram on stem width (see Figure 4) demonstrates that the edible and poisonous mushroom have two different peak regarding the stem width. A mushroom with wider stem width seems to be less pernicious than those with thinner stem width.

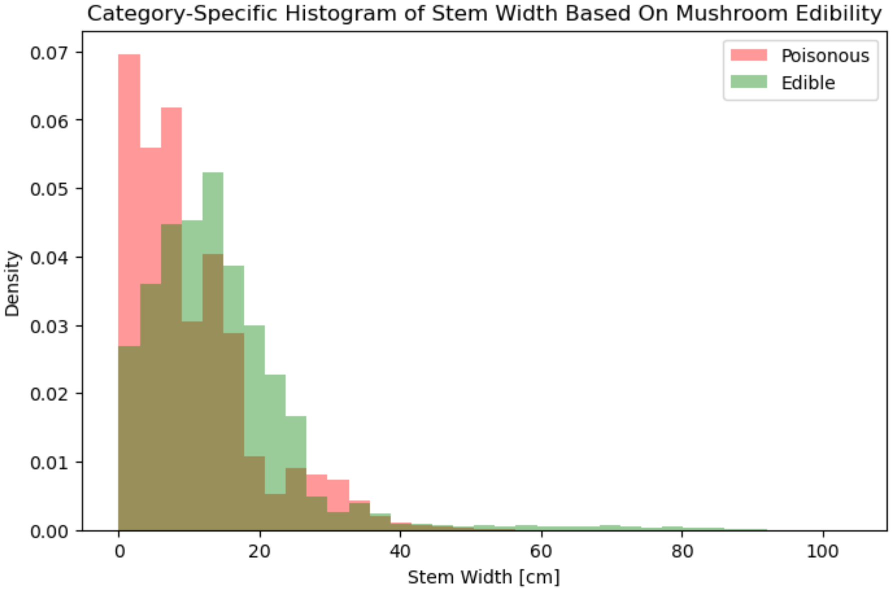


Figure 4: Category Specific Histogram on stem width.

3 Methods

3.1 Data Splitting

The Mushroom Edibility Classification dataset exhibits relatively balanced target variable classes, with a ratio of approximately 4 to 6 for the edible and poisonous classes, respectively. The general train_test_split is applied first then using Kfold cross validation with 4 splits. After splitting, there will be 80% of (train set + validation set) and 20% of test set.

3.2 Data Preprocessing

The dataset contains a total of 20 features for prediction, while 3 of them are continuous features and 17 of them are categorical features. For the continuous features, all of them follows a heavy-tailed distribution (see Figure 5) and are not bounded. Therefore, 2 preprocessors - one-hot encoder and standard scalars- are applied to the dataset. After preprocessing, the dataset contains 124 features and no feature has missing value.

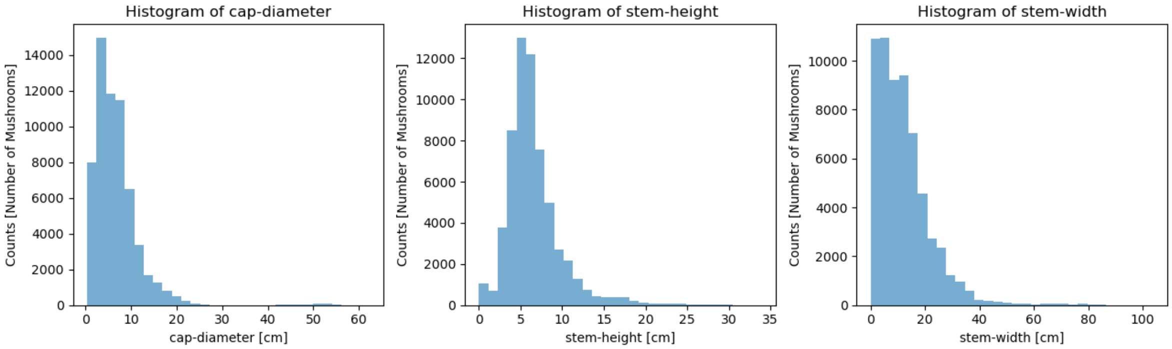


Figure 5: Histograms showing heavy-tailed distribution of continuous features.

3.3 Evaluation Metric

The Accuracy Score serves as the evaluation metric to examine the model's performance. Accuracy, a straightforward and intuitive measure, represents the proportion of correctly classified instances over the total number of instances. Given the balanced nature of this mushroom dataset, accuracy provides a clear and easily interpretable metric for overall model performance.

3.4 Machine Learning Models

In this classification problem, five machine learning models are utilized. The models and tuned parameters are in the following Table:

Machine Learning Model	**Parameter(s)**
Logistic Regression	C = 1/alpha: [1/0.001, 1/0.01, 1/0.1, 1/1.0]
Random Forest	max_depth: [5, 10, 20, 30] max_features: [0.25, 0.5, 0.75, 1.0] n_estimators: [20, 50, 100]
K Nearest Neighbors (KNN)	n_neighbors: [3,9,12,15,30,50,100]
XGBoost	max_depth: [3, 5, 7, 10] min_child_weight: [1, 3, 5] learning_rate: [0.1] lambda: [0.01, 0.1, 1] alpha: [0.01, 0.1, 1]
Support Vector Classification (SVC)	gamma: [1e-2, 1e-1, 1e1, 1e3] C: [1e-1, 1e0, 1e1, 1e2]

Table 1: The parameters tuned for each model.

3.5 Machine Learning Pipeline

To achieve hyper-parameter tuning, GridSearchCV is used to find the best set of parameters for a model by evaluating the model's performance using cross-validation. For each model, 5 different random states are assigned to each splitting to generalize the resulitng test scores. First, general train-test-split is applied to split the dataset into test set and other set. Then, in the gridsearchCV, it will evaluate the accuracy score for each set of parameters and for each splits. Then it will find the best parameters with the highest accuracy score. After that, it will evaluate the test score using the best parameters (see Figure 6).

In addition, in the XGBoost algorithm, similar pipeline is considered but early_stopping is added to halt the training process when the performance of the model on a validation dataset stops improving, avoiding the risk of overfitting.

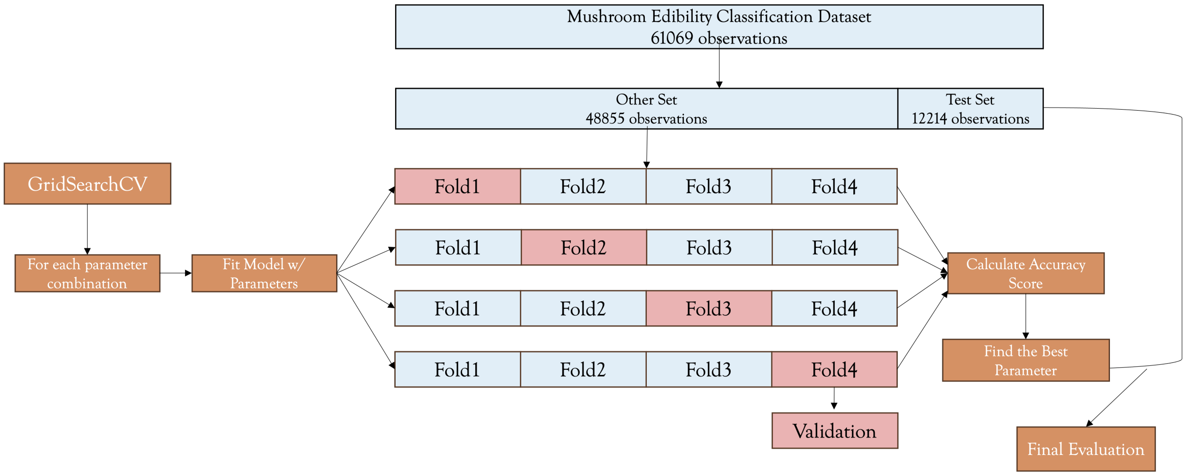


Figure 6: Visualization of finding the best set of parameters using GridSearchCV.

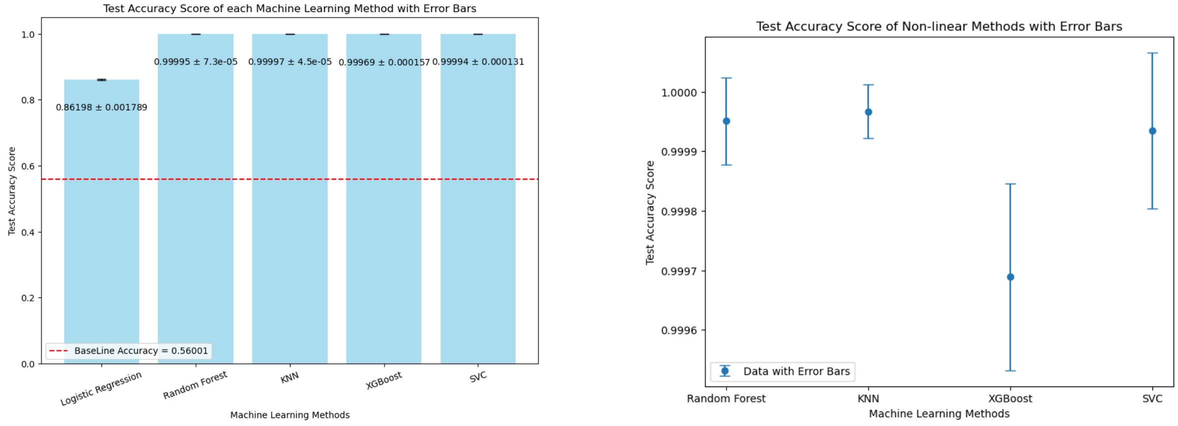
4 Results

4.1 Models' Performance

According to the mean test scores of the models evaluated on the test sets, **K Nearest Neighbors** appears to obtain the highest accuracy score (see Table 2). However, the test accuracy score of Logistic Regression is significantly different from that of other models, with a lowest accuracy score of 0.86 (see Figure 7(a)), while the other non-linear models achieve scores approximately equal to 0.99 (see Figure 7(b)). This strongly suggests that this dataset is non-linear and works better with tree-based models and non-parametric models.

ML Model	**Mean Test Score**	**Standard Deviation**	**# of std above Baseline**
Logistic Regression	0.861978	0.001789	168.789771
Random Forest	0.999951	0.000073	6026.546579
KNN	**0.999967**	**0.000045**	**9776.753340**
XGBoost	0.999689	0.000157	2800.483441
SVC	0.999935	0.000131	3358.182445

Table 2: The performance of the ML models and the corresponding mean test scores, standard deviations, and number of standard deviations above the baseline.



(a) Test Accuracy Score of each Machine Learning Method with Error Bars

(b) Test Accuracy Score of Non-linear Methods with Error Bars

Figure 7: Visualization of Models' Performance with Error Bar.

Next, I employ the best model, KNN, to compute the confusion matrix, visualizing the model's performance for each class (see Figure 8). In this confusion matrix, the KNN model with n_neighbors=3 achieves a perfect accuracy score of 1.0, signifying that all mushrooms in the test set are correctly classified based on their edibility.

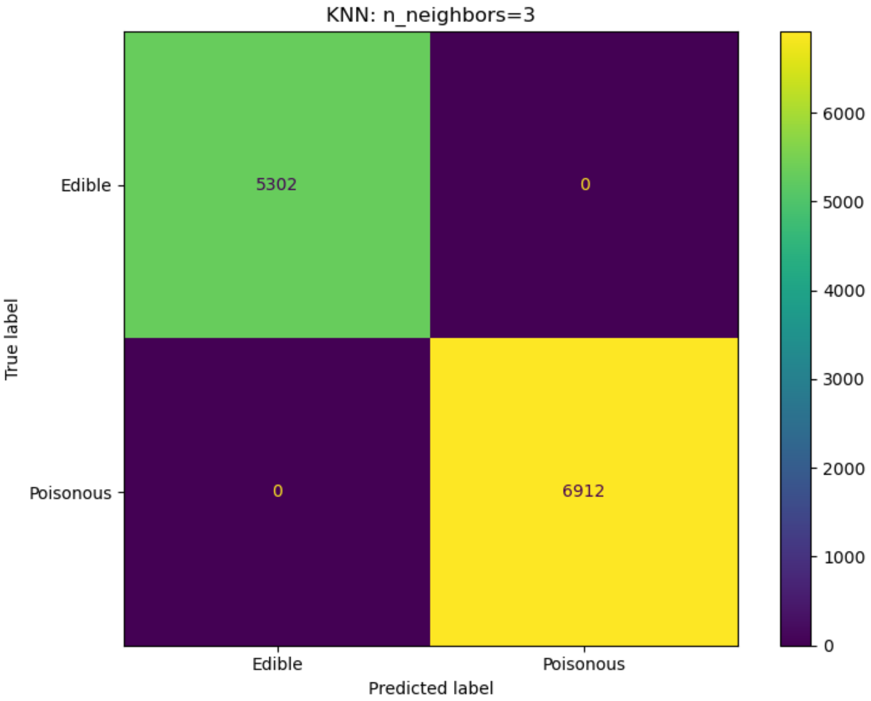


Figure 8: Confusion Matrix of the best KNN model.

4.2 Feature Importance

To make the model more interpretable, the global and local feature importance can help further understand how each feature contributes to the final prediction.

4.2.1 Permutation Importance

According to the Permutation Importance figure (see Figure 9),the stem_height exhibits the largest decrease in performance when its values are permuted, indicating its crucial role in the model's predictive accuracy. Similarly, stem_width and cap_diameter features in the dataset also lead to a significant decrease in the model's performance when permuted. In addition, gill_color and stem_color also indicate a relatively important role in the prediction.

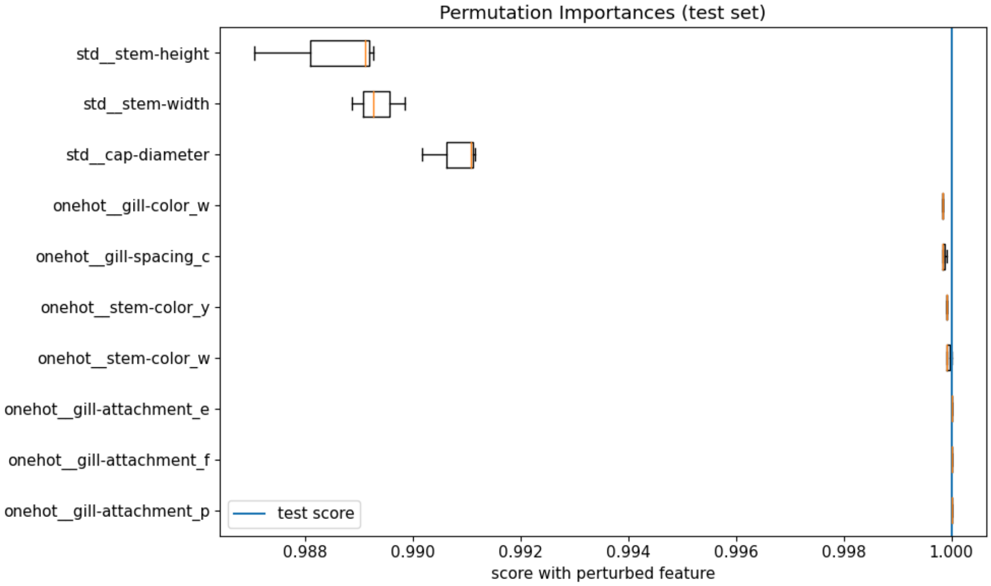


Figure 9: Top 10 Permutation Importance Features.

4.2.2 SHAP values

In this SHAP figure (see Figure 10), the three continuous features - stem_width, stem_heigh and cap_diameter - still play critical roles in the model prediction. Different from the Permutation Importance figure, SHAP metric also considers gill_attachment and gill_spacing as one the of critical features.

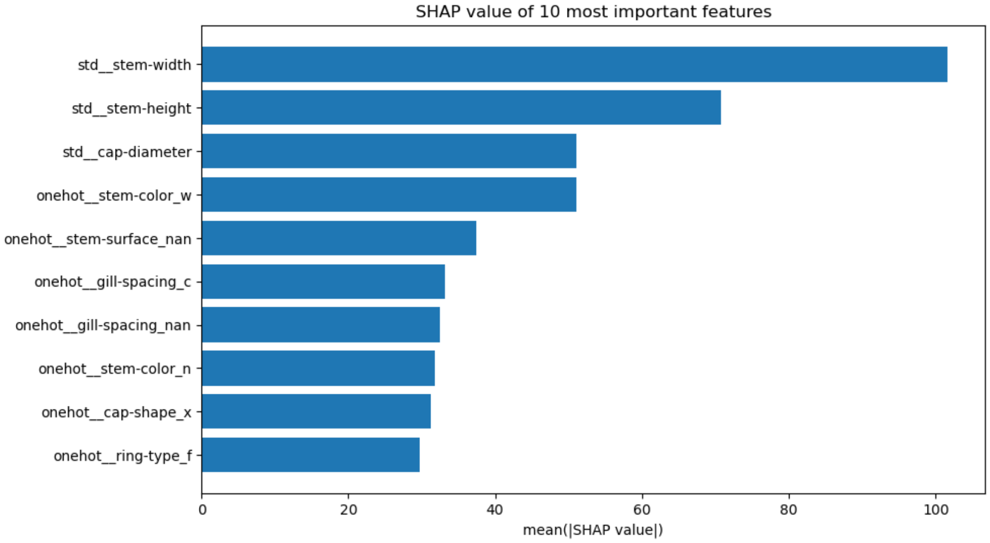


Figure 10: Top 10 most important features for mean SHAP values.

4.2.3 Random Forest Feature Importance Based on Mean Decrease in Impurity

The Random Forest Feature Importance is computed as the mean and standard deviation of accumulation of the impurity decrease within each tree. In this Random Forest Feature Importance figure (see Figure 11), similarly, the three continuous features rank the top 3 and following with stem_color, gill_spacing and stem_surface.

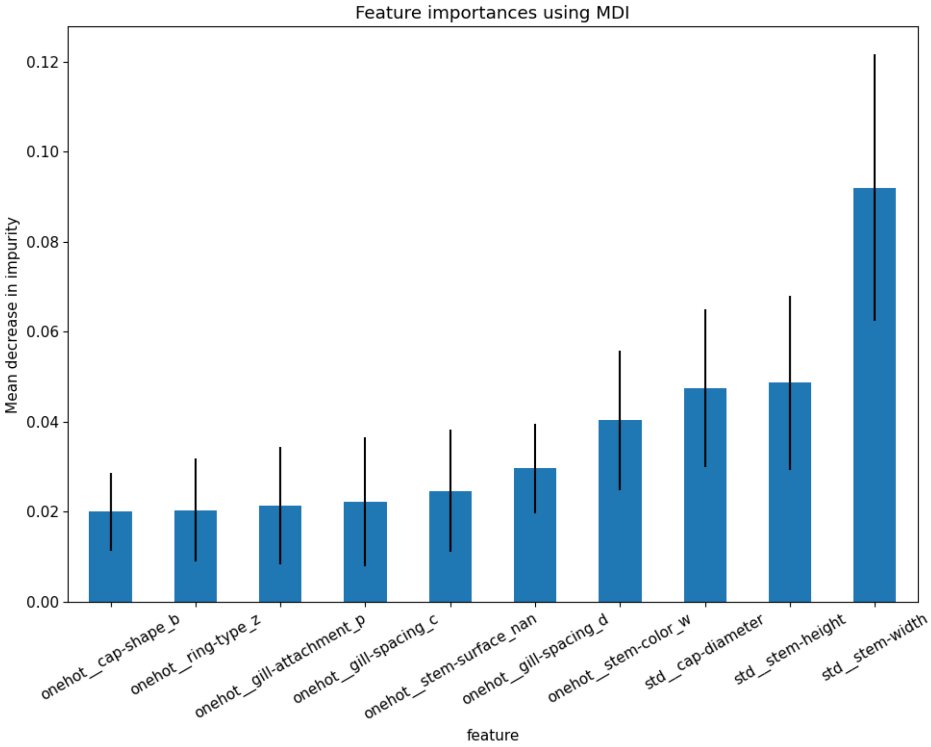


Figure 11: Top 10 most important features using MDI in Random Forest.

4.2.4 Force Plot

Using the Force Plots below can help understand how each feature contribute to the final decision of each observation. For the first instance (see Figure 12), the stem_width, cap_shape and stem_surface indicate the positive contribution to the prediction of the model, strongly suggesting the poisonous nature of this mushroom. Conversely, for the second instance (see Figure 13), the stem_width, gill_spacing and gill_color push the overall decision to the opposite side, indicating that these three features strongly suggest the edibility of this mushroom.

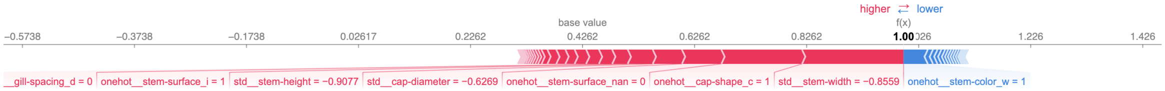


Figure 12: Feature Importance for mushroom with index=99.

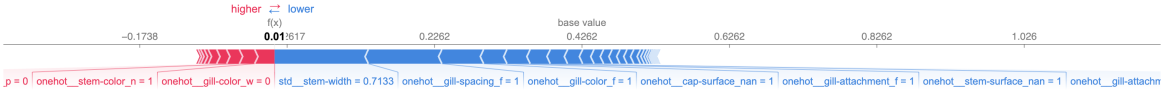


Figure 13: Feature Importance for mushroom with index=0.

4.3 Interesting Finding

After interpreting the feature importance, there is an aspect that aligns with EDA. In the specifically categorized histogram plotting the distribution of stem width based on edibility (see Figure 4), I hypothesize that a mushroom with wider stem width seems to be less pernicious than those with thinner stem width. When examining both force plots for local feature importance (see Figure 12 and Figure 13), the value of stem width for a poisonous mushroom (index=99) is less than 0, indicating a thin stem width. On the other hand, the value of stem width for an edible mushroom (index=0) is greater than 0.5, implying a relatively wider stem width.When interpreting individual predictions, the results from the force plots strongly correlate with the initial hypothesis presented in Figure 4.

5 Outlook

From the models I currently have, which exhibit strong predictive power, I recognize an opportunity to further enhance interpretability by undertaking feature selection from the pool of 124 features post-preprocessing. While more feature set may contribute to the model's accuracy, it can simultaneously introduce complexity and hinder interpretability. By narrowing down to a more concise set of features, I aim to simplify the model and make its decision-making process more transparent and comprehensible. This streamlined approach not only aids in understanding the critical factors influencing predictions but also makes faster execution reducing the burden on computational power.

References

[1] Haijiao Li et al. *Preplanned Studies: Mushroom Poisoning Outbreaks — China, 2022*. China CDC Weekly, 2023. DOI: 10.46234/ccdcw2023.009. Accessed: 7 Dec. 2023.

[2] Dev Zohaib. *Mushroom Edibility Classification*. Kaggle, 2023. URL: https://www.kaggle.com/datasets/devzohaib/mushroom-edibility-classification. Accessed: 7 Dec. 2023.

[3] Levi Payne. *Mushroom Edibility Classification XGBoost*. Kaggle, 2023. URL: https://www.kaggle.com/code/levipayne/mushroom-edibility-classification-xgboost. Accessed: 7 Dec. 2023.

[4] Lucas Agra. *Mushroom Edibility Classification*. Kaggle, 2023. URL: https://www.kaggle.com/code/lucasfca/mushroom-edibility-classification#Conclusion. Accessed: 7 Dec. 2023.