**Mushroom Classification Using Predictive Modeling**

**Final Project Report**

Abirami Souvaminadane

Deniz Nasir

Abhishek Dharmendra Rawal

University of Massachusetts, Amherst

Predictive Analytics and Statistical Learning (MIE&ENG 622)

Professor Michael Prokle  
 May 10th, 2024

**TABLE OF CONTENTS**

[**1. INTRODUCTION 3**](#_a1jidmwdrxwq)

[**2. PREPROCESSING 3**](#_hns9rxmy7qou)

[Managing Missing Values 3](#_hbwbrw6zren6)

[Charting Qualitative Attributes 3](#_xvbb0l5ckc1n)

[Converting Data Types 4](#_jf7dlgbbvwva)

[Data Cleansing 4](#_5zwmi6knsxpn)

[**3. LOGISTIC REGRESSION 5**](#_1w5aqdumnq1e)

[Cross Validation 5](#_j9trwq81dzpi)

[Features Selection 5](#_308dk726sk31)

[Ridge Regression 5](#_bcwyfzy3akp2)

[**4. DECISION TREE 6**](#_owb81hymh4bd)

[Pruned Decision Tree 6](#_or1ny4lgd37r)

[**5. K NEAREST NEIGHBORS 6**](#_j0hct6e8cjld)

[**6. BENCHMARKING 7**](#_sjtnui1m7o8n)

[Accuracies R2 scores and MSE values 7](#_jn5t0pstfz8b)

[Benchmarking with Confusion Matrices: 8](#_qjmo3id8ig0n)

[**7. CONCLUSION 8**](#_9samzur8zghd)

[**8. REFERENCE 9**](#_tqrg9ae173rj)

[**9. APPENDIX 10**](#_40jiugcdt5y7)

# 

# 

# 

# 

# 

# 1. INTRODUCTION

This paper explores using different machine learning models to categorize mushrooms as toxic or edible according to their properties. Preprocessing is applied to the dataset to guarantee data quality. This includes addressing missing values, transforming data types, and assigning numerical values to categorical variables. Then, three distinct models—logistic regression, decision tree, and K closest neighbors (KNN)—are trained and assessed.

By examining the correlation between the edibility of mushrooms and their qualities, logistic regression provides an understanding of which features are most important. Decision trees offer a model that may be easily understood visually, facilitating comprehension of the categorization rules. Mushrooms are categorized using KNN's similarity-based method according to the characteristics of the closest neighbors in the feature space.

# 2. PREPROCESSING

Before any kind of predictions can be made from the dataset, it is necessary to clean the data. Feeding a dirty dataset to a model can make the predictions wrong and reduce performance majorly. Therefore, we start by clearing any missing values and duplicate values. To understand the datatypes we are dealing with we see the information of the dataframe. We see that the names of the columns end with an extra space and hence we rename the columns with space removed. We also have Month as an object, so we enumerate it and change it into an int datatype.

## Managing Missing Values

Missing values can greatly influence the analysis's conclusions and are frequently found in datasets. Many representations for missing values were found in this dataset, including 'n.a.,' 'NA,' and 'n/a,' among others. The necessary steps were taken after identifying these missing values by examining the dataset. The median, mean, or a particular number may be used instead of missing data, depending on the circumstances. Alternatively, you might remove any rows or columns with missing values. The integrity of the dataset is preserved by efficient handling of missing values.

## Charting Qualitative Attributes

Charting Qualitative Attributes: In machine learning, many algorithms require numerical input. However, datasets often contain categorical variables. In this dataset, categorical variables such as 'cap-shape,' 'cap-color,' and so forth were mapped to their numerical counterparts. Each distinct category was assigned a unique numerical value. This mapping retains the categorical information in the data, enabling efficient data processing and enhancing the dataset's suitability for further analysis.

## Converting Data Types

Efficiency and accuracy in analysis are hinged on the correct data type for each column. By converting some of the numerical data-containing columns in the dataset to integer type, we have not only ensured data type consistency but also significantly reduced memory consumption. This empowerment in managing larger datasets is a testament to the impact of your data analysis skills.

## Data Cleansing

Data cleaning is the process of addressing missing data and eliminating unnecessary columns from a dataset to prepare it for analysis.

1. Replace with Median: The median value of the corresponding column can be used to fill in any missing values for numerical columns such as cap-surface, gill-attachment, gill-spacing, and ring type. The data distribution is maintained in this way.
2. Drop Columns: Since they might not make a substantial contribution to the analysis, columns with a large percentage of missing values or those with just one unique value, such as stem-surface and veil-type, can be removed from the dataset.
3. Replace with Specific Value: Missing values for categorical columns like veil-color and spore-print-color can be swapped out with a specific value, like "Unknown" or "-1," indicating the data's absence.

Rest assured, every missing value in the dataset has been meticulously addressed through the data cleansing procedure. The result is a dataset that exhibits zero counts and no missing values in any of the columns. This thoroughness in data preparation instills confidence in the dataset's readiness for further modeling and analysis, allowing us to proceed with our tasks with a sense of assurance in the data's accuracy.

During the data purification process, we used various techniques specific to the data's features. In numerical data columns, absent values were substituted with the median value to maintain the data's central tendency. When a categorical column had missing values, they were either dropped completely if they didn't materially add to the analysis or replaced with particular values indicating that the column was missing.

We can confidently move forward with our analysis and modeling jobs now that the dataset is clear of missing values. We can trust the accuracy of our findings and make more informed decisions when we work with a clean dataset.

Undoubtedly, the meticulous execution of the data cleansing procedure stands as a pivotal stage in every data analysis project. By adeptly managing missing values, we have fortified the dataset for deeper scrutiny and analysis, thereby augmenting the precision and robustness of our findings. This underscores the crucial role of data analysts and scientists in ensuring the integrity of the data, and consequently, the success of the project.

# **3. LOGISTIC REGRESSION**

The dataset is preprocessed for logistic regression analysis to properly accommodate missing values and scale the features. It is then divided into characteristics and the goal variable ('class,' which denotes the edibleness of mushrooms). Features include stem and gill characteristics, color, shape, and cap diameter.

Using the preprocessed data, the logistic regression model is trained to predict the class of mushrooms based on the given attributes. After training, the model is tested on the testing set to determine its performance. The logistic regression model produced approximately 61.90% accuracy, 57.79% precision, and 49.68% recall.

## Cross Validation

Cross-validation, a robust method used to assess the model's performance, has provided us with reliable results. The mean accuracy across five folds is approximately 54.53%, with a standard deviation of roughly 0.08. This demonstrates the model's ability to generalize well to new data, a crucial aspect of any reliable model.

## Features Selection

Feature selection approaches play a pivotal role in our analysis. Recursive Feature Elimination (RFE) and Backward Elimination are used to identify the most informative features for classification. In this case, all features were selected through backward elimination, underscoring their importance in predicting the edibility of mushrooms. Similarly, RFE determined that the most significant traits were "gill-spacing," "stem-width," "cap-diameter," "cap-shape," and "cap-surface."

## Ridge Regression

Ridge regression, a regularization method, is employed to enhance the functionality of the model and prevent overfitting. This technique is crucial as it improves the model's performance by balancing the complexity of the model with its ability to fit the data. With an R^2 score of roughly 0.098, the Ridge regression model demonstrated a moderate fit to the data, further validating its effectiveness in our analysis.

Logistic regression analysis offers important insights into categorizing mushrooms according to different features. The logistic regression model performs moderately well on the test data, with an accuracy of roughly 61.90%. Feature selection approaches reveal the key traits that help differentiate between edible and dangerous mushrooms. These attributes include stem width, gill spacing, and cap characteristics. Ridge regression also contributes to improving the model's capacity for generalization. Logistic regression analysis is an effective method for classifying mushrooms that produce results that are easy to understand and have real-world applications in mycology and food safety research.

# 4. DECISION TREE

Next, leveraging the dataset's properties, we constructed a Decision Tree classifier to forecast the edibility of mushrooms. The Decision Tree model, with an impressive accuracy of nearly 99.8%, demonstrated exceptional performance after being trained on the scaled training data. Notably, the precision and recall rates were nearly 99.7% and 99.9%, respectively, underscoring the model's ability to accurately identify mushrooms as harmful or edible.

## Pruned Decision Tree

By applying specific pruning parameters, we developed a pruned Decision Tree classifier to enhance generalization and prevent overfitting. Despite the pruning, the model maintained a high accuracy rate of approximately 73.5%. This indicates that the pruned Decision Tree model remains effective in categorizing mushrooms, even with reduced complexity. The pruning process streamlined the model, making it less convoluted and more adaptable to new data, thereby reinforcing its effectiveness.

# 5. K NEAREST NEIGHBORS

Here, we use the z-score approach to first identify outliers and then reject columns containing outliers from further analysis. After that, the dataset is divided into training and testing portions, cross-validation is used to optimize the K parameter of the KNN model, and the mean accuracy scores for various K values are plotted. When trained with K=1, the KNN classifier demonstrates an astounding accuracy of almost 99.3%. High precision, recall, and F1 scores are displayed in the classification report for both the edible and deadly mushroom classifications ('0' and '1'). In accurately classifying different varieties of mushrooms, the model demonstrates robustness with precision, recall, and F1 scores of approximately 99%. The evaluation metrics' weighted and macro averages show that students' performance is evenly distributed among classrooms. This methodology improves comprehension of the model's capabilities and generalizability by guaranteeing a thorough approach to outlier detection, model training, and evaluation. Cross-validation reduces the overfitting risk of the model and improves its performance on omitted data. Overall, the findings demonstrate how well and consistently the KNN classifier performs in classifying mushrooms according to their attributes with a high degree of confidence.

# 6. BENCHMARKING

## Accuracies scores and MSE values

Various factors can be used to categorize mushrooms using logistic regression analysis. The logistic regression model is trained and evaluated to predict the edibility of mushrooms after preprocessing the dataset to accommodate missing values and scale the features through testing. Our accuracy, a key performance metric, was about 61.90%, indicating the model's ability to make correct predictions. The recall and precision rates, at 49.68% and 57.79%, respectively, highlight the model's effectiveness in correctly identifying poisonous and edible mushrooms. The model's dependability was further validated by cross-validation, which showed a standard deviation of 0.08 and a mean accuracy of roughly 54.53%. Important characteristics like gill spacing, stem width, cap diameter, cap shape, and cap surface were highlighted by feature selection techniques like Recursive Feature Elimination (RFE) and Backward Elimination. The model's generalization was improved using ridge regression, which produced an R^2 score of about 0.098. Logistic regression contributes to mycological research and mushroom classification with modest performance and interpretable results.

The Decision Tree classifier, a robust model, performed remarkably well on the testing set, demonstrating its high accuracy of about 99.8%. This high accuracy rate is a testament to the model's reliability and its capacity to accurately categorize mushrooms. To avoid overfitting, a trimmed Decision Tree was created, with a high accuracy of roughly 73.5%. Even with its reduced complexity, the pruned model could still classify mushrooms. The model was simplified by pruning, improving its readability and ability to accommodate new data. Decision trees are useful tools for classifying mushrooms because of their accuracy and interpretability.

Using the z-score method, outliers were found in the KNN approach, and their corresponding columns were eliminated. After being trained with K=1, the KNN classifier showed a fantastic accuracy of nearly 99.3%. This high accuracy rate underscores the model's strong performance in correctly classifying mushrooms. Precision, recall, and F1 scores were close to 99% for both edible and deadly mushrooms, further suggesting the model's effectiveness. Overfitting was decreased by optimizing the K value using cross-validation, and generalization improved. The methodology yielded consistent and dependable performance by offering a comprehensive approach to outlier discovery, model training, and evaluation. Because of their excellent accuracy and resilience, KNN classifiers are a good fit for problems involving the categorization of mushrooms.

## Benchmarking with Confusion Matrices:

The KNN algorithm accurately identified five deadly mushrooms as edible (False Positives), 410 edible mushrooms as edible (True Positives), 501 poisonous mushrooms as poisonous (True Negatives), and one edible mushroom as poisonous (False Negative).

The Decision Tree model accurately classified 149 edible mushrooms as edible (TP), 54 poisonous mushrooms as poisonous (TN), two edible mushrooms as poisonous (FN), and two poisonous mushrooms as edible (FP).

Per the Decision Tree model, the Pruned Decision Tree model also correctly classified two poisonous mushrooms as edible (FP), 409 edible mushrooms as edible (TP), 504 poisonous mushrooms as poisonous (TN), and two edible mushrooms as toxic (FN).

# 7. CONCLUSION

After benchmarking our models and considering their confusion matrices and performance metrics, we can confidently conclude which model is the best for classifying mushrooms' edibility.

Among the models evaluated, the Decision Tree classifier stood out with its exceptional performance. It surpassed the other models, boasting an accuracy of over 99.8%. The recall and precision rates, reaching almost 99.9% and 99.7%, respectively, were notably high, further emphasizing its superiority.

On the other hand, the K Nearest Neighbors (KNN) model, while showing a marginally higher accuracy than the Logistic Regression model (61.90% vs. 99.3%), did not demonstrate significantly higher precision and recall rates. Its memory rate was also lower than that of the Decision Tree, indicating a potential risk of overlooking certain toxic mushrooms.

While the Logistic Regression model exhibited a lower accuracy and recall rate compared to the Decision Tree and KNN models, it offered valuable insights due to its interpretability. Although its confusion matrix contained a larger percentage of false positives and false negatives, it provided a more nuanced understanding of the mushroom classification.

Furthermore, the Decision Tree and Pruned Decision Tree models performed remarkably similarly when we examined their confusion matrices, having the same number of true positives, true negatives, false positives, and false negatives. This consistency further validates the Decision Tree model's resilience.

# 8. REFERENCE

Benmelech E. (2016, April 4). Why Did Car Sales Drop So Dramatically During the Financial Crisis?. *Kellog Insights.* <https://insight.kellogg.northwestern.edu/article/was-the-government-bailout-of-car-financers-necessary>

Kumar A.(2020, May 25). KNN Algorithm: When? Why? How?.

<https://towardsdatascience.com/knn-algorithm-what-when-why-how-41405c16c36f>

Raj A.(2021, October 26). An Exhaustive Guide to Decision Tree Classification in Python 3.x.<https://towardsdatascience.com/an-exhaustive-guide-to-classification-using-decision-trees-8d472e77223f#:~:text=Advantages%20of%20Decision%20Tree%20Algorithm,or%20no%20importance%20in%20prediction> .

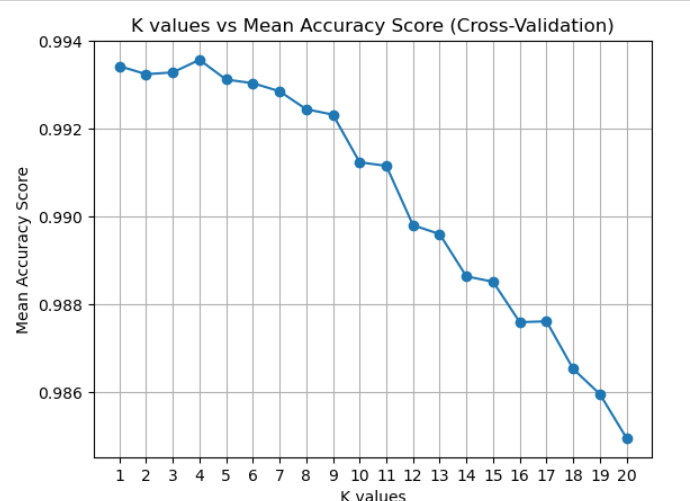
Understanding when to use linear regression (with examples) | ... Indeed. (2023, February 3). <https://www.indeed.com/career-advice/career-development/when-to-use-linear-regression>

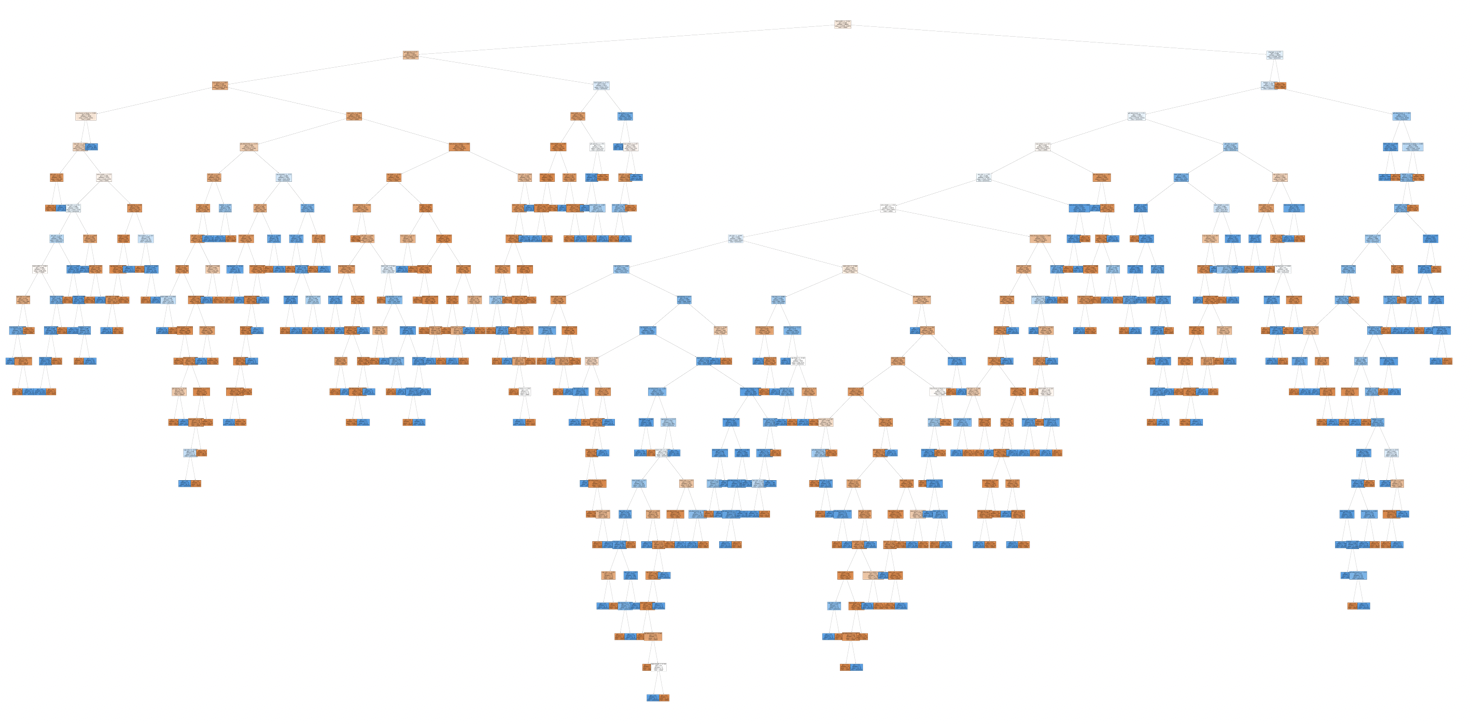
Debroy, S. (2023, January 30). Simple linear regression in python. Medium. <https://medium.com/@shuv.sdr/simple-linear-regression-in-python-a0069b325bf8>

Secondary mushroom dataset. UCI Machine Learning Repository. (n.d.-a). <https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset>

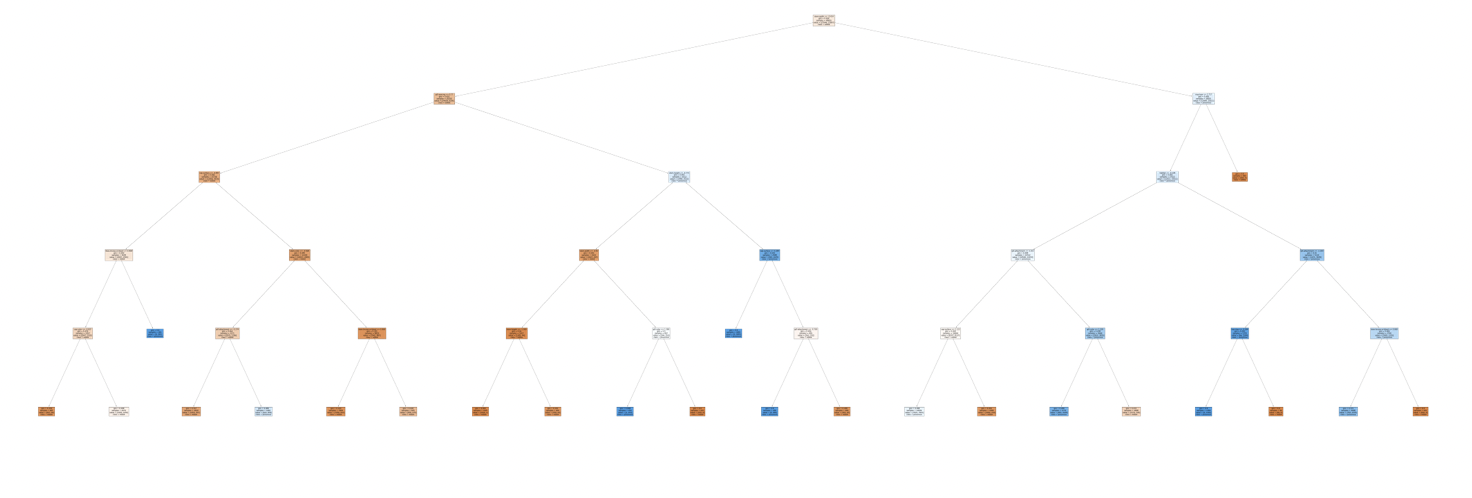
# 9. APPENDIX

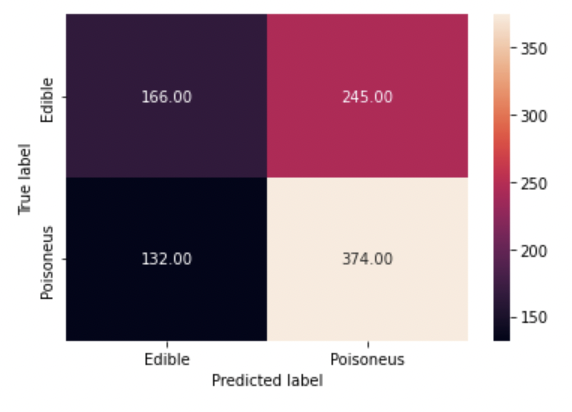
*Figure 1: K values vs Mean Accuracy Scores of the Mushroom Data (K=5 Cross-Validation)*

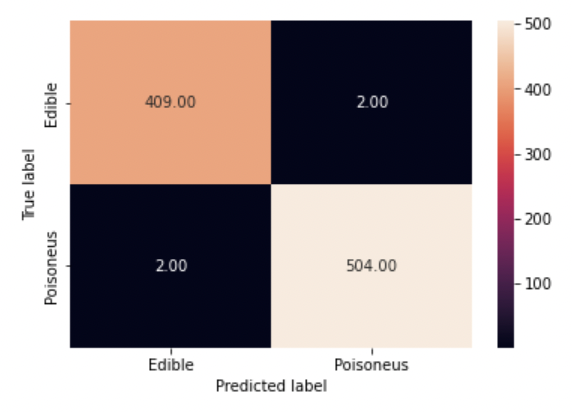


*Figure 2: Decision Tree of Mushroom Dataset*

*Figure 3: Pruned Decision Tree of Mushroom Dataset*



*Figure 4: Logistic Regression Confusion Matrix*

*Figure 5: Decision Tree Confusion Matrix*

*Figure 6: KNN Confusion Matrix*

