STOR565 Group 15 Final Project Report

To Eat or Not to Eat: ML-Driven Mushroom Classification

University of North Carolina at Chapel Hill
STOR 565
Yufeng Liu
Yubin Hu, Yao Ma, Xiuhan Chen

I. Summary

In this project, we used classic machine learning methods, such as Ordinary Least Square Logistic Regression, Ridge and LASSO Regression, Decision Tree (Classification Tree), Boosting, and Random Forest, to determine or predict whether a mushroom is edible or poisonous. We collected our data from the UCI Machine Learning Repository. There are 61068 mushrooms with caps based on 173 species (353 mushrooms per species). Such a mushroom dataset has 20 key attributes describing the mushrooms' appearance, colors, dimensions, habitats, etc. For consistency and analytical purposes, we encoded "Edible" as 0 and "Poisonous" as 1 in the classification categories. Our research addresses two critical objectives in mushroom classification: identifying definitive physical characteristics that determine edibility and developing robust methods to minimize dangerous misclassifications of poisonous specimens as edible.

We first introduced the Logistic Regression method along with Ridge and LASSO regression to better understand and predict mushrooms' edibility based on their physical features. We've encountered some restrictions and limitations, so we have extended our methods to the Decision Tree. We developed two Decision Tree models: one using the original data and another using weighted data to focus on reducing the false negative rate in our predictions. Subsequently, we employed Boosting and Random Forest methods to further evaluate and enhance our predictive models. Ultimately, we assessed the performance and limitations of each model, providing an informed response to our objectives.

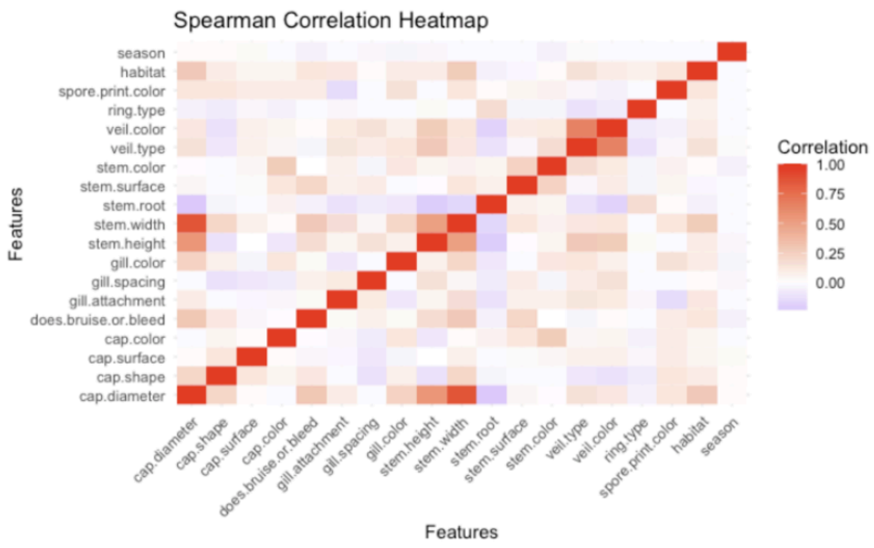
II. AI Usage Disclaimer & Member Contributions

Usage	Tool Used (e.g., ChatGPT-4)	How you edited the output, if at all	Conversation Link (If available)
Topic selection	No AI used		
Brainstorming and idea generation	ChatGPT-4	use AI to help brainstorm which models to use	
Research	No AI used		
Source validation	No AI used		
Outlining/planning	No AI used		
Drafting	No AI used		
Media creation	No AI used		
Peer review	No AI used		
Revising	ChatGPT-4	Debug R code	
Polishing	ChatGPT-4, Claude AI	Polished our writing and check grammar mistakes	
Other			

All team members—Xiuhan Chen, Yubin Hu, and Yao Ma—contributed equally to this research through collaborative data analysis, model development, presentation, and final report, with each member providing essential insights across all project phases.

III. Data Preparation & Exploration

To improve dataset quality and model reliability, we excluded more than 300 rows with various missing values labeled "unknown" or "empathy" among the 20 variables. We also removed the "has_ring" column to eliminate redundancy with "ring_type." Mushrooms with a "true" value under the "has_ring" column already have a specific "ring_type," while mushrooms with "has_ring" = "false" are fully represented by "ring_type" = "none." This would represent a strong multicollinearity.



IV. Modeling Approaches

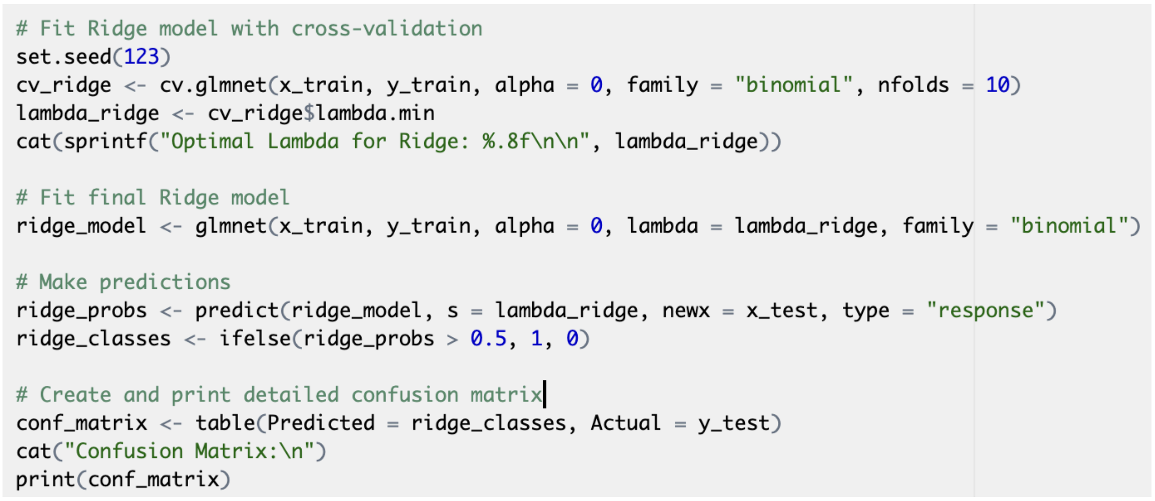
Model 1: Logistic Regression

Our investigation into mushroom classification began with logistic regression as our foundational model, given the binary nature of our classification problem (edible vs. poisonous). After excluding target variables and implementing a train-test split, we developed our initial logistic regression model to establish baseline performance metrics.

```
# Step 3: Fit and display model
logistic_model <- glm(formula, data = mushroom_train, family = binomial)
```

The logistic regression model demonstrated promising overall performance, achieving 87.5% accuracy in mushroom classification. The model showed particular strength in sensitivity, correctly identifying 88.23% of poisonous mushrooms while maintaining 86.58% specificity for edible mushroom identification. An Area Under the Curve (AUC) of 0.9423 further confirmed the model's robust discriminative capabilities. However, a critical safety concern emerged: the misclassification of 1,330 poisonous mushrooms as edible, corresponding to an 11.77% false negative rate. Given the potentially lethal consequences of such misclassifications in real-world applications, this error rate necessitated the exploration of alternative modeling approaches.

Model 2 + 3: Regularization - Ridge Regression & Lasso Regression
To potentially improve prediction accuracy and address possible multicollinearity in our feature set, we implemented two regularization techniques: ridge regression (L2) and lasso regression (L1). Ridge regression, implemented with an optimal lambda value of 0.01069948, unexpectedly resulted in decreased model performance. The accuracy declined to 85.81%, representing a 1.69% decrease from our logistic regression baseline. More concerning was the increased false negative rate of 13.16%, resulting in 156 additional dangerous misclassifications. This performance degradation suggests that the L2 regularization may have overly constrained important mushroom characteristics in the feature set.



```r
# Fit Ridge model with cross-validation
set.seed(123)
cv_ridge <- cv.glmnet(x_train, y_train, alpha = 0, family = "binomial", nfolds = 10)
lambda_ridge <- cv_ridge$lambda.min
cat(sprintf("Optimal Lambda for Ridge: %.8f\n\n", lambda_ridge))

# Fit final Ridge model
ridge_model <- glmnet(x_train, y_train, alpha = 0, lambda = lambda_ridge, family = "binomial")

# Make predictions
ridge_probs <- predict(ridge_model, s = lambda_ridge, newx = x_test, type = "response")
ridge_classes <- ifelse(ridge_probs > 0.5, 1, 0)

# Create and print detailed confusion matrix
conf_matrix <- table(Predicted = ridge_classes, Actual = y_test)
cat("Confusion Matrix:\n")
print(conf_matrix)
```

Lasso regression, with its L1 regularization approach, offered marginally better results. Using an optimal lambda of 0.00002252, the model achieved 87.53% accuracy while maintaining the same sensitivity (88.23%) as our original logistic regression. The extremely small lambda value indicates minimal feature selection was necessary, suggesting our initial feature set was appropriately chosen. However, with 1,329 poisonous mushrooms still misclassified as edible - only one fewer than logistic regression - the improvement proved insufficient for practical application.

```r
# Fit LASSO model with cross-validation
set.seed(123)
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 1, family = "binomial", nfolds = 10)
lambda_lasso <- cv_lasso$lambda.min
cat(sprintf("Optimal Lambda for LASSO: %.8f\n\n", lambda_lasso))

# Fit final LASSO model
lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = lambda_lasso, family = "binomial")

# Make predictions
lasso_probs <- predict(lasso_model, s = lambda_lasso, newx = x_test, type = "response")
lasso_classes <- ifelse(lasso_probs > 0.5, 1, 0)

# Create and print detailed confusion matrix
conf_matrix <- table(Predicted = lasso_classes, Actual = y_test)
cat("Confusion Matrix:\n")
print(conf_matrix)
```

Our comparative analysis revealed that while LASSO and logistic regression performed similarly and superior to ridge regression, even our best model now which is Lasso or logistic models shared a fundamental limitation: approximately one in eight poisonous mushrooms was misclassified as edible. This consistent pattern suggests inherent limitations in linear classification approaches for this particular problem. The models' limitations manifested in two key areas: structural constraints and interpretability challenges. Structurally, all three models were limited by their linear decision boundaries and assumptions of feature independence, struggling to capture complex interactions between mushroom characteristics. Interpretability posed additional challenges, particularly after regularization, making it difficult to translate model insights into practical identification guidelines for mushroom safety.

Model 4: Decision Tree

The data preparation phase began with a stratified 70-30 train-test split to maintain class distribution, ensuring reproducibility through a systematic seed setting. Our initial decision tree model was constructed using the part algorithm, incorporating all available features as predictors. The model's structure was visualized to provide insights into the decision-making process, revealing key splitting points based on mushroom characteristics. Performance evaluation utilized a comprehensive set of metrics, with predictions generated using a standard 0.5 probability threshold for binary classification. In such process, Gini Impurity is a key metric to determine the best way to split the data at each decision node. Gini Impurity quantifies the probability of misclassifying a randomly selected sample from a subset of the classification based on the distribution of classes within that subset (Pramod, 2023). At each node, the algorithm evaluates possible splits by calculating the Gini Impurity of the resulting child nodes. The formula for Gini Impurity is given by:

$$Gini = 1 - \sum_{k=1}^{K} p_k^2$$

p_k is the proportion of samples belonging to class k in the subset

K is the number of classes (in this case, edible and poisonous)

```
# Build a classification tree on the training set
tree_model <- rpart(class ~ ., data = train_data, method = "class")
```

```
# Plot the classification tree
rpart.plot(tree_model, type = 4, extra = 102, fallen.leaves = TRUE, cex = 0.8)
```

```
# Make predictions on the test set
pred_probs <- predict(tree_model, test_data, type = "prob")[,2]
pred_class <- ifelse(pred_probs > 0.5, 1, 0)
```

We then evaluated the model's performance by creating a confusion matrix with the ***confusionMatrix*** function. The model demonstrated strong overall performance with 88.38% accuracy (95% CI: 87.91% - 88.84%) and an AUC of 0.915, indicating excellent discriminative ability. Sensitivity and specificity measures of 87.34% and 89.21% respectively suggested balanced performance across both classes. The Kappa statistic of 0.7649 confirmed substantial agreement between predicted and actual classifications.

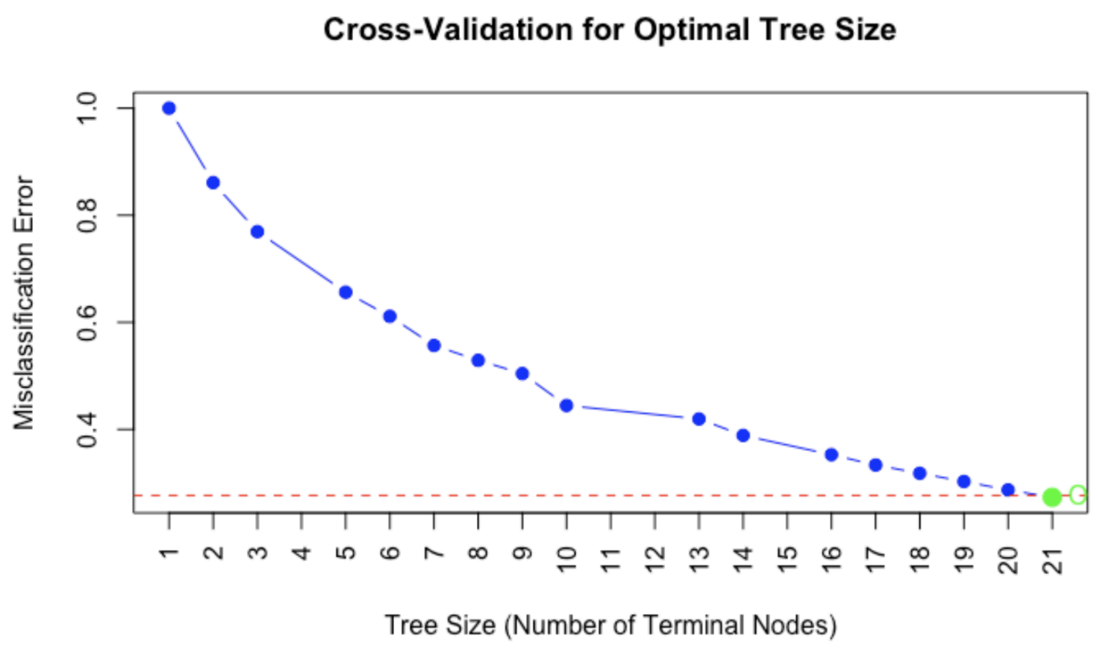
The complexity of the decision tree is assessed by counting the number of terminal nodes (leaves) using the sum function on the model's frame where the variable is ***"<leaf>"***. A higher number of terminal nodes may indicate a more complex model prone to overfitting.

```
# Count the number of terminal nodes in the tree
num_terminal_nodes <- sum(tree_model$frame$var == "<leaf>")
cat("Number of Terminal Nodes:", num_terminal_nodes, "\n")
```

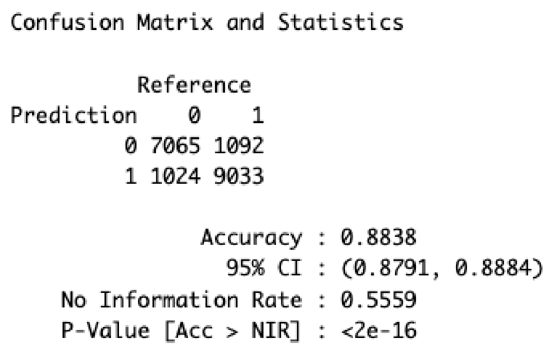
To further evaluate the model's discriminatory power, a Receiver Operating Characteristic (ROC) curve is plotted using the roc function from the ***pROC*** package. The ROC curve illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) across different threshold levels. The Area Under the Curve (AUC) is calculated to quantify the model's ability to distinguish between poisonous and edible mushrooms, with values closer to 1 indicating excellent discrimination.

```
# ROC Curve and AUC Calculation
roc_curve <- roc(test_data$class, pred_probs)

# Plot the ROC Curve
plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)
abline(a = 0, b = 1, col = "gray", lty = 2) # Add diagonal line

# Calculate and print AUC
auc_value <- auc(roc_curve)
cat("AUC:", auc_value, "\n")
```

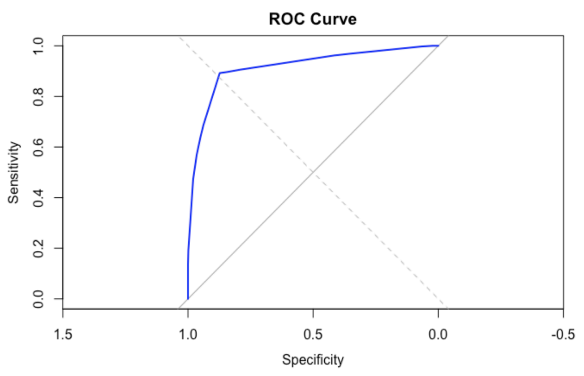
After building the model, the code extracts the complexity parameter table (cp_table) from the decision tree object. This table contains information about the tree at each pruning step, including the number of splits ***(nsplit)*** and the cross-validated error rate ***(xerror)***. The total number of terminal nodes ***(tree_size)*** at each stage is calculated by adding one to the number of splits. The code then identifies the optimal tree size by finding the index where the misclassification error is minimized ***(which.min(misclassification_error))*** and retrieves the corresponding tree size ***(optimal_tree_size)***. After doing 10-fold cross-validation, we found the optimal tree size is 21.



The decision tree model achieved strong performance metrics with 88.38% accuracy (95% CI: 87.91% - 88.84%). The model demonstrated balanced classification capabilities with 87.34% sensitivity and 89.21% specificity, supported by an AUC of 0.915 and a Kappa statistic of 0.7649. These metrics, along with the McNemar's test p-value of 0.1452, indicate consistent and reliable performance across both edible and poisonous mushroom classifications. Overall, the model is well-suited for the task, combining high accuracy, strong sensitivity and specificity, and robust discriminative ability, as reflected in the AUC value. This makes it a reliable tool for identifying edible and poisonous mushrooms in practical applications.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7065 1092
         1 1024 9033

               Accuracy : 0.8838
                 95% CI : (0.8791, 0.8884)
    No Information Rate : 0.5559
    P-Value [Acc > NIR] : <2e-16
```



The ROC curve's proximity to the top-left corner confirms the model's robust discriminative ability between classes, making it a reliable tool for mushroom classification. However, despite these strong statistical measures, the model's practical application must be considered carefully given the critical nature of accurate poisonous mushroom identification. There are still a considerable amount of false negatives in the data (about 1092) mistakenly predicted as edible while they really represent poisonous mushrooms. The false negative rate is about 10.79%. We want to figure out a way to reduce the number.

Model 5: The Weighted Decision Tree

Creating a weighted loss matrix would help solve this problem. The loss matrix assigns higher penalties to false negatives compared to false positives, reflecting the real-world importance of correctly identifying poisonous mushrooms. Specifically, the penalty for predicting a poisonous mushroom as edible is set to 1, while the penalty for predicting an edible mushroom as poisonous is set to 10. These weights are incorporated into the decision tree model using the ***parms = list(loss = loss_matrix)*** parameter. By doing so, the model prioritizes reducing false negatives during training, even if it results in a slightly higher number of false positives. When building the tree, the algorithm evaluates potential splits at each decision node to minimize the **expected misclassification cost**.

```r
# Create a weighted loss matrix
# Loss matrix: rows = actual, columns = predicted
# 0 = edible, 1 = poisonous
loss_matrix <- matrix(c(0, 10, 1, 0), ncol = 2)

# Build a classification tree with class weights
set.seed(123)
weighted_tree <- rpart(class ~ ., data = train_data, method = "class",
                       parms = list(loss = loss_matrix), xval = 10, cp = 0.01)
```

The weighted decision tree model significantly reduced false negatives from 1,092 to 59 cases by implementing a loss matrix that heavily penalized the misclassification of poisonous mushrooms. While this increased false positives from 1,024 to 3,119 and decreased overall accuracy from 88.38% to 82.55%, the model achieved its primary safety objective with 99.42% specificity.

```
Weighted Tree Confusion Matrix:
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0  4970    59
         1  3119 10066

               Accuracy : 0.8255
                 95% CI : (0.8199, 0.831)
    No Information Rate : 0.5559
    P-Value [Acc > NIR] : < 2.2e-16
```

The trade-off resulted in lower sensitivity (61.44%) but maintained a high Positive Predictive Value (98.83%), ensuring reliable identification of safe mushrooms. The model's AUC of 0.8736 and balanced accuracy of 80.43% demonstrate effective performance, with gill attachment emerging as the most significant decision split. This safety-first approach prioritizes preventing dangerous misclassifications, accepting reduced overall accuracy as a reasonable trade-off.

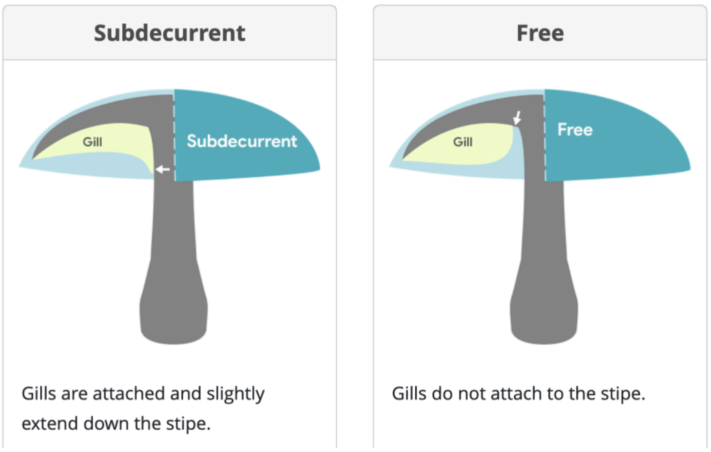

Subdecurrent
Gill
Subdecurrent
Gills are attached and slightly extend down the stipe.
Free
Gill
Free
Gills do not attach to the stipe.

Figure: Two Types of Mushroom Gill-Attachment

Model 6: Gradient Boosting Machines (GBM) + Random Forest

Building upon the foundational framework of decision trees, Gradient Boosting Machines (GBM) were employed as an advanced ensemble technique to improve prediction accuracy for binary classification. Instead of constructing a single model, GBM begins with an initial model and continually fits new models by minimizing a loss function to achieve the most accurate prediction. (He et al., 2019). The objective was to classify mushrooms as either poisonous or non-poisonous using a decision threshold of 0.5. A mushroom was classified as poisonous if the model predicted a probability of 0.5 or higher.

```r
# Train a Gradient Boosting Model
set.seed(123)
gbm_model <- gbm(
  formula = class01 ~ .,
  data = mushroom_train %>% select(-class),
  distribution = "bernoulli",
  n.trees = 100,
  interaction.depth = 3,
  shrinkage = 0.1,
  n.minobsinnode = 10
)

# Make predictions on the test set
gbm_predictions <- predict(gbm_model, mushroom_test, n.trees = 100, type = "response")
gbm_class <- ifelse(gbm_predictions > 0.5, 1, 0)
```

The GBM model achieved an overall accuracy of 93.12%. The sensitivity, representing the proportion of correctly identified poisonous mushrooms, was 92.55%. While the model performed well, 727 poisonous mushrooms were misclassified as non-poisonous, indicating a potential safety concern in practical applications.

	var	rel.inf
	<chr>	<dbl>
cap.surface	cap.surface	15.5534614
stem.color	stem.color	12.8054697
gill.attachment	gill.attachment	12.6498512
stem.root	stem.root	8.3681622

Feature analysis revealed the most influential attributes contributing to the classification process. The top three features were Cap Surface (relative importance of 15.55), Stem Color (relative importance of 12.81), and Gill Attachment (relative importance of 12.65). These attributes significantly influenced the model's predictions. This analysis provides insight into the key variables driving classification and highlights areas for potential refinement to improve sensitivity and reduce misclassification risks.

While Gradient Boosting Machines delivered impressive results with an accuracy of 93.12% and valuable insights into key features driving the predictions, it also highlighted some limitations, such as the misclassification of 727 poisonous mushrooms. The Random Forest algorithm was employed to further optimize classification accuracy and address these shortcomings. Recognized for its robustness and capability to handle complex datasets, Random Forest provides an alternative ensemble learning approach.

```r
# Train a Random Forest model
set.seed(123)  # For reproducibility
rf_model <- randomForest(class01 ~ .-class,
                         data = mushroom_train,
                         ntree = 100,
                         importance = TRUE)
summary(rf_model)

# Make predictions on the test set
rf_predictions <- predict(rf_model, mushroom_test)

# Evaluate the model's performance
confusion_matrix <- confusionMatrix(rf_predictions, mushroom_test$class01)
```

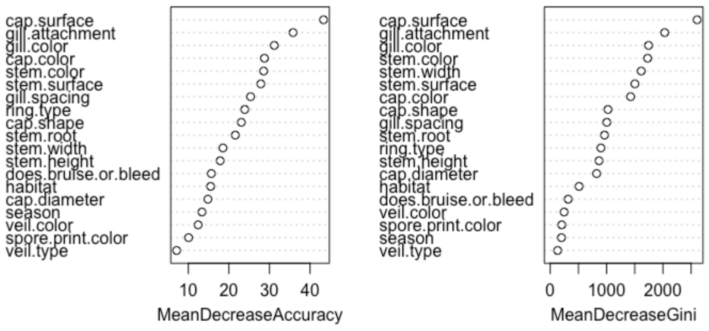
The Random Forest model achieved **perfect accuracy (100%)**, with all mushrooms correctly classified as either edible or poisonous. The confusion matrix reveals no misclassifications, as evidenced by the absence of false positives and false negatives. Both sensitivity and specificity are 1.0000, confirming that the model accurately identifies all edible mushrooms and poisonous mushrooms without error. Additionally, positive and negative predictive values are also perfect, demonstrating the reliability of the predictions.

```
          Reference
Prediction     0     1
         0  8942     0
         1     0 11296

               Accuracy : 1
                 95% CI : (0.9998, 1)
    No Information Rate : 0.5582
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA
```

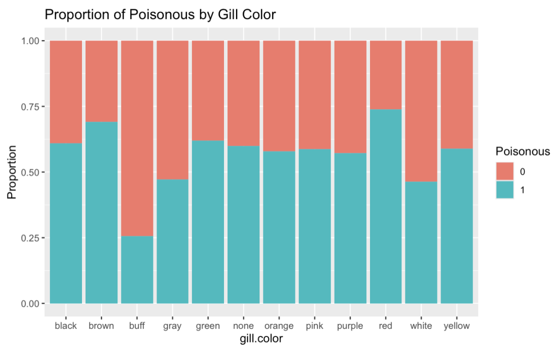
```
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.4418
         Detection Rate : 0.4418
   Detection Prevalence : 0.4418
      Balanced Accuracy : 1.0000

       'Positive' Class : 0
```

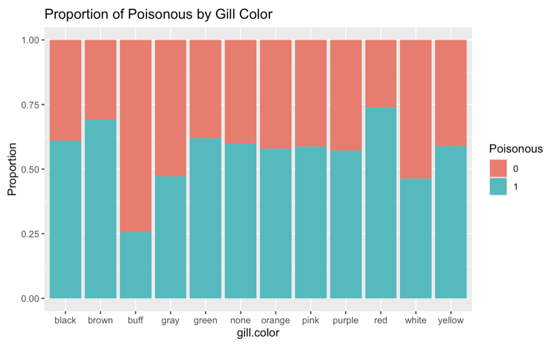
The feature importance plots from the Random Forest model provide key insights into the variables driving the predictions. Two metrics are used. The first one is Mean Decrease in Accuracy: This measures the impact of each feature on the model's predictive performance. Features such as **cap.surface**, **gill.attachment**, and **gill.color** significantly influences the classification, as their removal results in a substantial decrease in accuracy. Mean Decrease in Gini evaluates the purity achieved by splitting on a given feature within the decision trees.

Again, **cap.surface**, **gill.attachment**, and **gill.color** is the most critical feature for distinguishing between edible and poisonous mushrooms.



Based on the key features identified through the Random Forest model, we conducted a proportional analysis to better understand how specific mushroom attributes influence their classification as poisonous or edible. The analysis focuses on two critical features: **cap surface** and **gill color**.





The left chart analyzes the relationship between the surface texture of the mushroom cap (Cap.surface) and its likelihood of being poisonous. Certain surface types, such as **silky** and **fibrous**, show a higher proportion of poisonous mushrooms (blue bars). In contrast, other textures like **smooth** and **shiny** exhibit a more balanced or predominantly edible proportion (red bars).

The right chart examines the gill.color feature, showing a strong association between certain gill colors and the likelihood of being poisonous. Colors like **brown** and **red** are predominantly poisonous, as evidenced by the high proportion of blue bars. Conversely, **buff** and **white** gill colors exhibit a more balanced or predominantly edible composition.

V. Results & Discussion

Our project demonstrated the effectiveness of machine learning in classifying mushrooms as edible or poisonous by analyzing physical features such as cap surface, gill attachment, and gill color. Advanced models like Random Forest and Gradient Boosting achieved exceptional results, with Random Forest attaining perfect accuracy and Gradient Boosting delivering 93.12% accuracy. These models leveraged key features identified through importance analysis, such as shiny cap surfaces and vivid gill colors, to enhance classification reliability, highlighting the practical applicability of machine learning for such tasks. Our weighted decision tree approach successfully reduced dangerous misclassifications from 1,092 to 59 cases, though at the cost of increased false positives.

However, the study revealed significant limitations that must be addressed. Simpler models like LASSO and Logistic Regression consistently performed around 87% accuracy and misclassified about one in eight poisonous mushrooms as edible, making them unsuitable for real-world applications. Additionally, the dataset used was biased toward specific habitats, such as woods, raising concerns about the model's ability to generalize to mushrooms found in other environments. Feature dependence, particularly on variables like cap surface and gill attachment, poses another challenge, as missing or misclassified features could significantly impact the model's accuracy. Furthermore, complex models like Random Forest, while highly accurate, remain difficult to interpret for non-technical stakeholders.

Future improvements should address these limitations by expanding datasets to include diverse habitats and species, balancing class representation, and exploring strategies to handle missing data. Enhancing model interpretability through techniques like decision tree visualizations or Shapley Additive Explanations (SHAP) (Lundberg et al., 2017) would also improve their usability for broader audiences. While machine learning shows immense potential for mushroom classification, tackling these challenges will be critical for developing robust, generalizable, and practical models for real-world use.

Reference

Data Mining Lab 4: New Tree Data Set and loss matrices. (n.d.). https://www.louisaslett.com/Courses/Data_Mining_09-10/ST4003-Lab4-New_Tree_Data_Set_and_Loss_Matrices.pdf

Encyclopædia Britannica, inc. (2024, October 22). *Expected value*. Encyclopædia Britannica. https://www.britannica.com/topic/expected-value

He, Z., Lin, D., Lau, T., & Wu, M. (2019, August 19). *Gradient boosting machine: A survey*. arXiv.org. https://arxiv.org/abs/1908.06951

Lundberg, S. M., Lee, S.-I., Scott M. LundbergPaul G. Allen School of Computer Science, U. of W., & Su-In LeePaul G. Allen School of Computer Science, D. of G. S. (2017, December 4). *A unified approach to interpreting model predictions: Proceedings of the 31st International Conference on Neural Information Processing Systems*. Guide Proceedings. https://dl.acm.org/doi/10.5555/3295222.3295230

Types of gill attachments - mushroom identification. Mycology Start. (n.d.). https://mycologyst.art/identification/mushroom-morphology/types-of-gill-attachments/

Pramod, O. (2023, January 29). *Decision trees*. Medium. https://medium.com/@ompramod9921/decision-trees-91530198a5a5