Which Mushroom will kill you?!

Alex Martinez

2024-08-05

The structure below is one possible setup for a data analysis project (including the course project). For a manuscript, adjust as needed. You don’t need to have exactly these sections, but the content covering those sections should be addressed.

This uses MS Word as output format. [See here](https://quarto.org/docs/output-formats/ms-word.html) for more information. You can switch to other formats, like html or pdf. See [the Quarto documentation](https://quarto.org/) for other formats.

Warning: package 'here' was built under R version 4.3.3

# 1. Summary/Abstract

**Briefly describe what the data is, how it was collected, where you will get (or got) it from. How many observations do you have, what was measured? Anything else important to report about the data?**

The dataset I will be using is from Kaggle called [Muchroom Dataset](https://www.kaggle.com/datasets/prishasawhney/mushroom-dataset/data). The kaggle version is a cleaned version of the original [Mushroom Dataset for Binary Classification](https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset), available at UCI Library. The data set contains 9 columns and 54036 rows. There is one variable that is binary which indicates if the mushroom is edible or not. The following 8 variables are different features of the mushroom. Here are those variables.

Cap Diameter

Cap Shape

Gill Attachment

Gill Color

Stem Height

Stem Width

Stem Color Season

Target Class - Is it edible or not?

**At this stage you are not required to already have and show the data, but if you do, even better. Then add a few lines of code which load the data and using some of the commands you learned about, provide summary descriptions of the data.**

I am still working on importing and creating summary descriptions of my dataset.

**Explain the question you want to answer using the data. What will be your outcome(s) of interest (if any)? What (if any) specific predictors will you focus on? What relations/patterns are you looking for in the data?**

The main quesiton I want to answer is the following. *Determine what is the best model to predict if a mushroom is edible or not using the following 8 variables are your predictors.*

Cap Diameter

Cap Shape

Gill Attachment

Gill Color

Stem Height

Stem Width

Stem Color Season

Target Class - Is it edible or not?

I will focus on each predictor variable to see what is most significant in predicting if a mushroom is edible or not.

**As much as you know, suggest how you will analyze it. At this stage in the course, we haven’t covered analysis approaches yet, so you can keep things vague and non-technical here.**

I will run multiple different regression models to determine which is best at predicting the predictor varible. Some models I will consider are, linear/multilinear regression, logistic regression, decision tree models, etc.

**You are allowed, but not yet required, to provide background information for the question you plan to answer. For instance you can describe why it’s an interesting question, who else has done similar analyses, how your analysis will be new/different, etc. Similar to what you read in an introduction to a research paper. For the final report, you’ll need these parts. For part 1, they are not required, but you are welcome to already write down some of that.**

I want to enhance my modeling skills by trying multiple models to obtain the largest accuracy. Something unique that I will be doing compared to everything Ive seen on Kaggle for this data set is that I will be using R while everyone else used Python.

**Eventually, for your final report, what you write for this part will go into different sections of the full report. Some will go into the introduction, some in the methods section. You can already place these items there, or for now just write them as a single section.**

Ok thanks.

### 1.0.1 BELOW IS WHAT WILL GET DONE THROUGHOUT THE SEMESTER (:

# 2. Introduction

## 2.1 General Background Information

*Provide enough background on your topic that others can understand the why and how of your analysis*

## 2.2 Description of data and data source

This dataset is a cleaned version of the original [Mushroom Dataset for Binary Classification](https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset) Available at UCI Library. This dataset was cleaned using various techniques such as Modal imputation, one-hot encoding, z-score normalization, and feature selection. It contains 9 columns:

1. Cap Diameter
2. Cap Shape
3. Gill Attachment
4. Gill Color
5. Stem Height
6. Stem Width
7. Stem Color
8. Season
9. Target Class - Is it edible or not?

The Target Class contains two values - 0 or 1 - where 0 refers to edible and 1 refers to poisonous.

## 2.3 Questions/Hypotheses to be addressed

*Using the Variables listed above, what is the best predictive model to determine if a mushroom is edible or not.*

# 3. Methods

*Describe your methods. That should describe the data, the cleaning processes, and the analysis approaches. You might want to provide a shorter description here and all the details in the supplement.*

The modeling methods I will be using are the following

A random forest model is trained using 100 trees. A decision tree is just one simple decision tree that is evaluated. A lighGBM model is efficient for large datasets but can still work on smaller ones. GBM uses cross-validation to find the optimal number of trees.

## 3.1 Data aquisition

*The data is from the Kaggle website that has been listed above. It is a CSV file with 9 variables and ~50,000 rows.*

## 3.2 Data import and cleaning

The data is pretty cleaned already espescially since it is only 9 variables and ~50,000 rows. Here is the import

library(tidyverse)

Warning: package 'tidyverse' was built under R version 4.3.3

Warning: package 'ggplot2' was built under R version 4.3.3

Warning: package 'tibble' was built under R version 4.3.3

Warning: package 'tidyr' was built under R version 4.3.3

Warning: package 'readr' was built under R version 4.3.3

Warning: package 'purrr' was built under R version 4.3.3

Warning: package 'dplyr' was built under R version 4.3.3

Warning: package 'stringr' was built under R version 4.3.3

Warning: package 'forcats' was built under R version 4.3.3

Warning: package 'lubridate' was built under R version 4.3.3

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

getwd()

[1] "C:/Users/alexm/Downloads/Pract 2/AlexMartinez-Analysis-Pract2/products/manuscript"

mushrooms <- read.csv('C:/Users/alexm/Downloads/Pract 2/AlexMartinez-Analysis-Pract2/data/mushroom\_cleaned.csv')  
  
  
## Lets Explore the data  
head(mushrooms)

cap.diameter cap.shape gill.attachment gill.color stem.height stem.width  
1 1372 2 2 10 3.807467 1545  
2 1461 2 2 10 3.807467 1557  
3 1371 2 2 10 3.612496 1566  
4 1261 6 2 10 3.787572 1566  
5 1305 6 2 10 3.711971 1464  
6 1337 6 2 10 3.775635 1520  
 stem.color season class  
1 11 1.8042727 1  
2 11 1.8042727 1  
3 11 1.8042727 1  
4 11 1.8042727 1  
5 11 0.9431946 1  
6 11 0.9431946 1

str(mushrooms)

'data.frame': 54035 obs. of 9 variables:  
 $ cap.diameter : int 1372 1461 1371 1261 1305 1337 1300 1354 1222 1085 ...  
 $ cap.shape : int 2 2 2 6 6 6 2 6 6 6 ...  
 $ gill.attachment: int 2 2 2 2 2 2 2 2 2 2 ...  
 $ gill.color : int 10 10 10 10 10 10 10 10 10 10 ...  
 $ stem.height : num 3.81 3.81 3.61 3.79 3.71 ...  
 $ stem.width : int 1545 1557 1566 1566 1464 1520 1563 1532 1476 1581 ...  
 $ stem.color : int 11 11 11 11 11 11 11 11 11 11 ...  
 $ season : num 1.804 1.804 1.804 1.804 0.943 ...  
 $ class : int 1 1 1 1 1 1 1 1 1 1 ...

summary(mushrooms)

cap.diameter cap.shape gill.attachment gill.color   
 Min. : 0.0 Min. :0 Min. :0.000 Min. : 0.00   
 1st Qu.: 289.0 1st Qu.:2 1st Qu.:0.000 1st Qu.: 5.00   
 Median : 525.0 Median :5 Median :1.000 Median : 8.00   
 Mean : 567.3 Mean :4 Mean :2.142 Mean : 7.33   
 3rd Qu.: 781.0 3rd Qu.:6 3rd Qu.:4.000 3rd Qu.:10.00   
 Max. :1891.0 Max. :6 Max. :6.000 Max. :11.00   
 stem.height stem.width stem.color season   
 Min. :0.000426 Min. : 0 Min. : 0.000 Min. :0.02737   
 1st Qu.:0.270997 1st Qu.: 421 1st Qu.: 6.000 1st Qu.:0.88845   
 Median :0.593295 Median : 923 Median :11.000 Median :0.94319   
 Mean :0.759110 Mean :1051 Mean : 8.418 Mean :0.95216   
 3rd Qu.:1.054858 3rd Qu.:1523 3rd Qu.:11.000 3rd Qu.:0.94319   
 Max. :3.835320 Max. :3569 Max. :12.000 Max. :1.80427   
 class   
 Min. :0.0000   
 1st Qu.:0.0000   
 Median :1.0000   
 Mean :0.5492   
 3rd Qu.:1.0000   
 Max. :1.0000

## 3.3 Statistical analysis

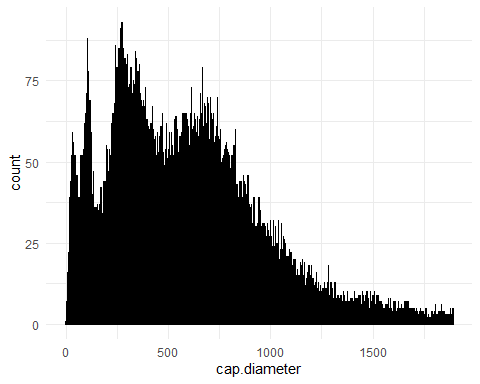
*Explain anything related to your statistical analyses.*

# 4. Results

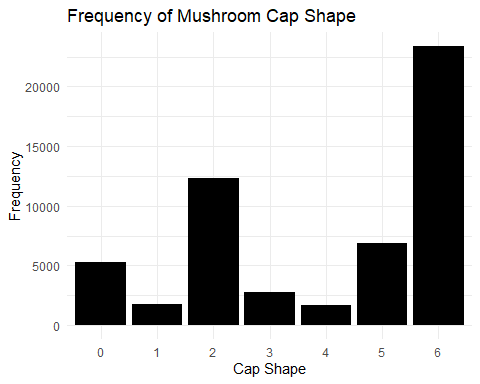
## 4.1 Exploratory/Descriptive analysis

*Use a combination of text/tables/figures to explore and describe your data. Show the most important descriptive results here. Additional ones should go in the supplement. Even more can be in the R and Quarto files that are part of your project.*

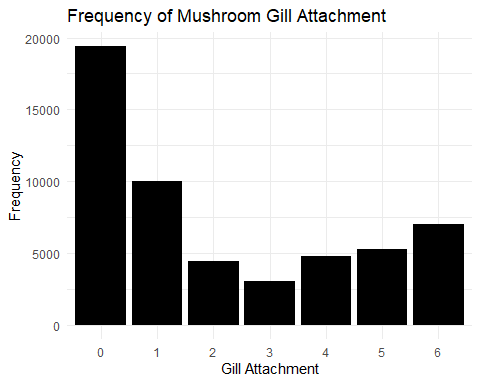
#Exploring all varibles with some visualizations  
mushrooms %>% ggplot(aes(x = cap.diameter)) +  
 geom\_histogram(binwidth = 1, fill = "blue", color = "black") +  
 theme\_minimal()



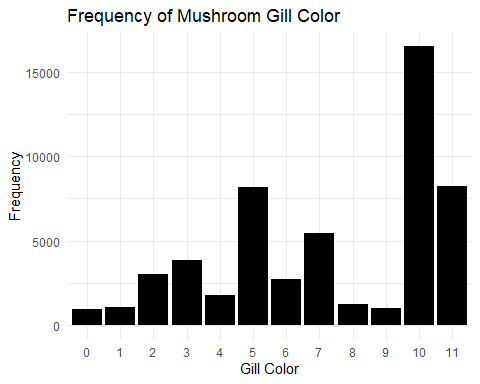
ggplot(mushrooms, aes(x = factor(cap.shape))) +  
 geom\_bar(fill = 'black') +  
 labs(x = "Cap Shape", y = "Frequency", title = "Frequency of Mushroom Cap Shape") +  
 theme\_minimal()



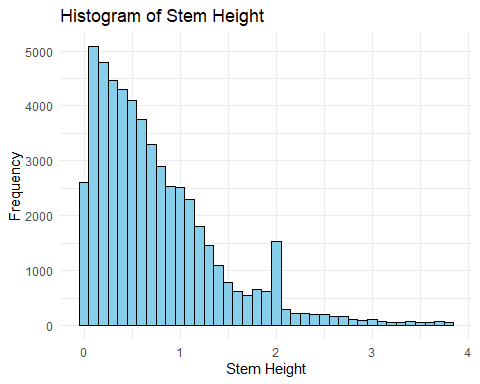
ggplot(mushrooms, aes(x = factor(gill.attachment))) +  
 geom\_bar(fill = 'black') +  
 labs(x = "Gill Attachment", y = "Frequency", title = "Frequency of Mushroom Gill Attachment") +  
 theme\_minimal()



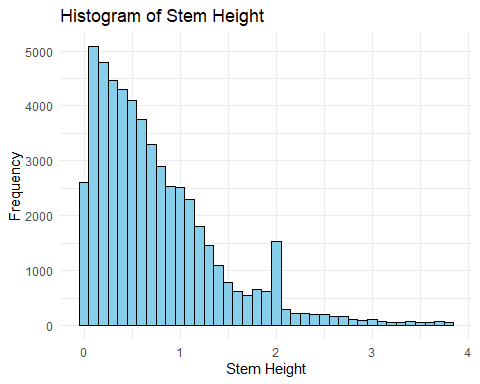
ggplot(mushrooms, aes(x = factor(gill.color))) +  
 geom\_bar(fill = 'black') +  
 labs(x = "Gill Color", y = "Frequency", title = "Frequency of Mushroom Gill Color") +  
 theme\_minimal()



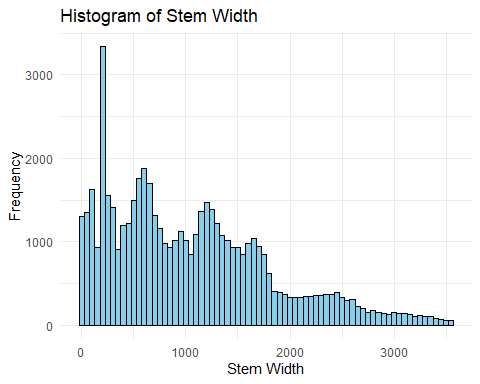
ggplot(mushrooms, aes(x = stem.height)) +  
 geom\_histogram(binwidth = 0.1, fill = "skyblue", color = "black") +  
 labs(x = "Stem Height", y = "Frequency", title = "Histogram of Stem Height") +  
 theme\_minimal()



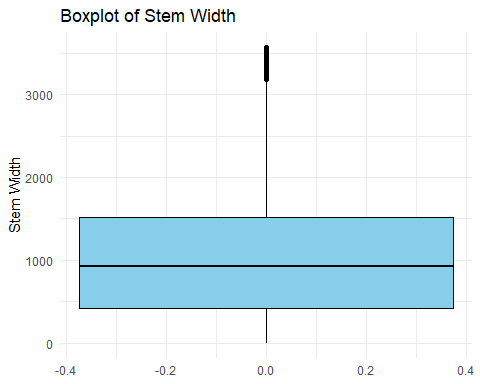
ggplot(mushrooms, aes(x = stem.height)) +  
 geom\_histogram(binwidth = 0.1, fill = "skyblue", color = "black") +  
 labs(x = "Stem Height", y = "Frequency", title = "Histogram of Stem Height") +  
 theme\_minimal()



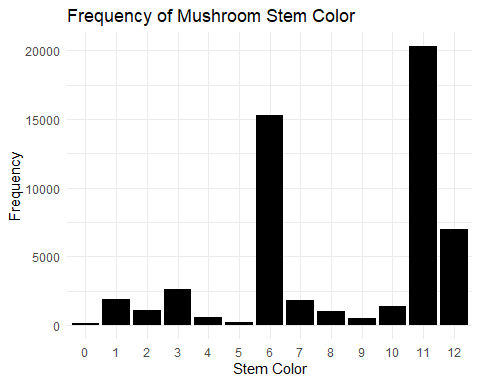
ggplot(mushrooms, aes(x = stem.width)) +  
 geom\_histogram(binwidth = 50, fill = "skyblue", color = "black") +  
 labs(x = "Stem Width", y = "Frequency", title = "Histogram of Stem Width") +  
 theme\_minimal()



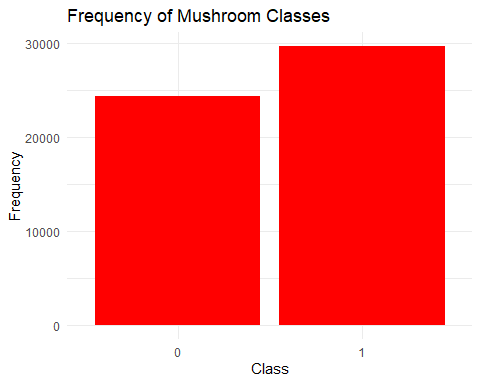
ggplot(mushrooms, aes(y = stem.width)) +  
 geom\_boxplot(fill = "skyblue", color = "black") +  
 labs(y = "Stem Width", title = "Boxplot of Stem Width") +  
 theme\_minimal()



ggplot(mushrooms, aes(x = factor(stem.color))) +  
 geom\_bar(fill = 'black') +  
 labs(x = "Stem Color", y = "Frequency", title = "Frequency of Mushroom Stem Color") +  
 theme\_minimal()



ggplot(mushrooms, aes(x = factor(class))) +  
 geom\_bar(fill = 'red') +  
 labs(x = "Class", y = "Frequency", title = "Frequency of Mushroom Classes") +  
 theme\_minimal()



#lets view a correlation matrix of all the variables  
library(corrplot)

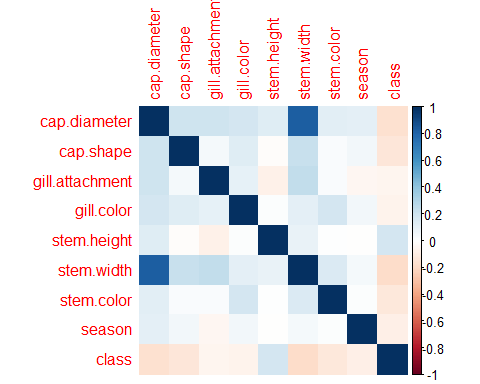
Warning: package 'corrplot' was built under R version 4.3.3

corrplot 0.92 loaded

correlation\_matrix <- cor(mushrooms)  
print(correlation\_matrix)

cap.diameter cap.shape gill.attachment gill.color  
cap.diameter 1.0000000 0.20401127 0.20048148 0.18637715  
cap.shape 0.2040113 1.00000000 0.04306624 0.13138708  
gill.attachment 0.2004815 0.04306624 1.00000000 0.10027559  
gill.color 0.1863772 0.13138708 0.10027559 1.00000000  
stem.height 0.1356524 -0.01039312 -0.07528418 0.01505654  
stem.width 0.8284693 0.22249376 0.24529991 0.11028297  
stem.color 0.1218561 0.02903463 0.02007325 0.18609020  
season 0.1133345 0.05544203 -0.04031458 0.05996487  
class -0.1656763 -0.13333831 -0.05254126 -0.06394667  
 stem.height stem.width stem.color season  
cap.diameter 0.1356524038 0.82846931 0.121856112 0.1133344523  
cap.shape -0.0103931195 0.22249376 0.029034633 0.0554420257  
gill.attachment -0.0752841798 0.24529991 0.020073247 -0.0403145793  
gill.color 0.0150565412 0.11028297 0.186090200 0.0599648711  
stem.height 1.0000000000 0.09809506 0.002623783 -0.0002919205  
stem.width 0.0980950645 1.00000000 0.157394266 0.0406794021  
stem.color 0.0026237827 0.15739427 1.000000000 0.0107499434  
season -0.0002919205 0.04067940 0.010749943 1.0000000000  
class 0.1833538889 -0.18285619 -0.128339344 -0.0829189342  
 class  
cap.diameter -0.16567630  
cap.shape -0.13333831  
gill.attachment -0.05254126  
gill.color -0.06394667  
stem.height 0.18335389  
stem.width -0.18285619  
stem.color -0.12833934  
season -0.08291893  
class 1.00000000

# correlation matrix heatmap  
corrplot(correlation\_matrix, method = "color")



## 4.2 Basic statistical analysis

*To get some further insight into your data, if reasonable you could compute simple statistics (e.g. simple models with 1 predictor) to look for associations between your outcome(s) and each individual predictor variable. Though note that unless you pre-specified the outcome and main exposure, any “p<0.05 means statistical significance” interpretation is not valid.*

## 4.3 Full analysis

*Use one or several suitable statistical/machine learning methods to analyze your data and to produce meaningful figures, tables, etc. This might again be code that is best placed in one or several separate R scripts that need to be well documented. You want the code to produce figures and data ready for display as tables, and save those. Then you load them here.*

#libraries needed  
library(tidyverse)  
library(caret)

Warning: package 'caret' was built under R version 4.3.3

Loading required package: lattice

Attaching package: 'caret'

The following object is masked from 'package:purrr':  
  
 lift

library(randomForest)

Warning: package 'randomForest' was built under R version 4.3.3

randomForest 4.7-1.1

Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

The following object is masked from 'package:dplyr':  
  
 combine

The following object is masked from 'package:ggplot2':  
  
 margin

library(rpart)

Warning: package 'rpart' was built under R version 4.3.3

library(lightgbm)

Warning: package 'lightgbm' was built under R version 4.3.3

library(gbm)

Warning: package 'gbm' was built under R version 4.3.3

Loaded gbm 2.1.9

This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.com/gbm-developers/gbm3

data <- mushrooms  
  
# Convert class to factor  
data$class <- as.factor(data$class)  
  
# Scale the specified columns  
scaled\_data <- data %>%  
 mutate(across(c(cap.diameter, stem.width, season), scale))  
  
# Convert necessary columns to factors  
scaled\_data <- scaled\_data %>%  
 mutate(across(c(cap.shape, gill.attachment, gill.color, stem.color), as.factor))  
  
# Split data into features and target variable  
X <- scaled\_data %>%  
 select(-class)  
y <- scaled\_data$class  
  
# Set seed for reproducibility  
set.seed(123)  
  
# Create training and testing datasets  
trainIndex <- createDataPartition(y, p = 0.9, list = FALSE)  
X\_train <- X[trainIndex, ]  
X\_test <- X[-trainIndex, ]  
y\_train <- y[trainIndex]  
y\_test <- y[-trainIndex]  
  
# Train Random Forest Classifier  
rf\_model <- randomForest(x = X\_train, y = y\_train, ntree = 100)  
rf\_pred <- predict(rf\_model, X\_test)  
rf\_accuracy <- mean(rf\_pred == y\_test)  
print(paste("Random Forest Accuracy:", rf\_accuracy))

[1] "Random Forest Accuracy: 0.988709975939293"

# Train Decision Tree Classifier  
dt\_model <- rpart(class ~ ., data = data.frame(X\_train, class = y\_train))  
dt\_pred <- predict(dt\_model, X\_test, type = "class")  
dt\_accuracy <- mean(dt\_pred == y\_test)  
print(paste("Decision Tree Accuracy:", dt\_accuracy))

[1] "Decision Tree Accuracy: 0.820099944475292"

# Convert data to matrix format for LightGBM  
X\_train\_matrix <- as.matrix(X\_train)  
X\_test\_matrix <- as.matrix(X\_test)  
  
# Train LightGBM Classifier  
lgb\_train <- lgb.Dataset(data = X\_train\_matrix, label = as.numeric(y\_train) - 1)  
params <- list(objective = "multiclass", num\_class = length(levels(y\_train)))  
lgb\_model <- lgb.train(params, lgb\_train, 100)

[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.002772 seconds.  
You can set `force\_col\_wise=true` to remove the overhead.  
[LightGBM] [Info] Total Bins 809  
[LightGBM] [Info] Number of data points in the train set: 48632, number of used features: 8  
[LightGBM] [Info] Start training from score -0.796700  
[LightGBM] [Info] Start training from score -0.599319

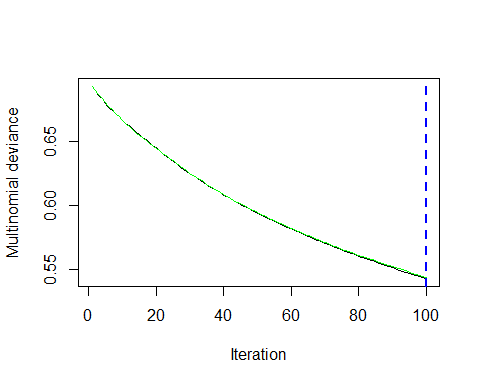
lgb\_pred <- predict(lgb\_model, X\_test\_matrix)  
lgb\_pred\_class <- factor(max.col(lgb\_pred) - 1, levels = 0:(length(levels(y\_train)) - 1))  
lgb\_accuracy <- mean(lgb\_pred\_class == y\_test)  
print(paste("LightGBM Accuracy:", lgb\_accuracy))

[1] "LightGBM Accuracy: 0.97427355173052"

# Train GBM Classifier using gbm package  
gbm\_model <- gbm(class ~ ., data = data.frame(X\_train, class = y\_train), distribution = "multinomial", n.trees = 100, cv.folds = 5, interaction.depth = 3, shrinkage = 0.01, n.minobsinnode = 10)

Warning: Setting `distribution = "multinomial"` is ill-advised as it is  
currently broken. It exists only for backwards compatibility. Use at your own  
risk.

gbm\_pred <- predict(gbm\_model, newdata = X\_test, n.trees = gbm.perf(gbm\_model, method = "cv"), type = "response")



gbm\_pred\_class <- factor(max.col(gbm\_pred), levels = 1:length(levels(y\_train)) - 1)  
gbm\_accuracy <- mean(gbm\_pred\_class == y\_test)  
print(paste("GBM Accuracy:", gbm\_accuracy))

[1] "GBM Accuracy: 0.549139367018323"

Best Performance

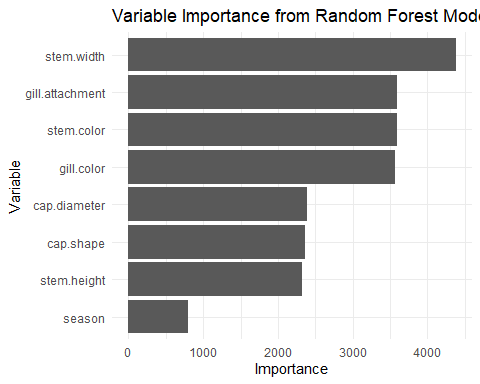
#best variables  
print(paste("Random Forest Accuracy:", rf\_accuracy))

[1] "Random Forest Accuracy: 0.988709975939293"

var\_importance <- importance(rf\_model)  
var\_importance\_df <- data.frame(Variable = rownames(var\_importance),   
 Importance = var\_importance[, 'MeanDecreaseGini'])  
  
# Print the variable importance  
print(var\_importance\_df)

Variable Importance  
cap.diameter cap.diameter 2385.4351  
cap.shape cap.shape 2364.3550  
gill.attachment gill.attachment 3589.0687  
gill.color gill.color 3562.2552  
stem.height stem.height 2326.6286  
stem.width stem.width 4380.7891  
stem.color stem.color 3587.2115  
season season 792.6014

# Plot variable importance  
library(ggplot2)  
ggplot(var\_importance\_df, aes(x = reorder(Variable, Importance), y = Importance)) +  
 geom\_bar(stat = "identity") +  
 coord\_flip() +  
 xlab("Variable") +  
 ylab("Importance") +  
 ggtitle("Variable Importance from Random Forest Model") +  
 theme\_minimal()



# 5. Discussion

## 5.1 Summary and Interpretation

The four classification models we used to help determine what mushroom is edible or not is the Random Forest, Decision Tree, LightBGM, and GBM classifier. We will determine which one is the best by looking at its performance on our test dataset.

A random forest model is trained using 100 trees. A decision tree is just one simple decision tree that is evaluated. A lighGBM model is efficient for large datasets but can still work on smaller ones. GBM uses cross-validation to find the optimal number of trees.

## 5.2 Conclusions

Our best model is the random forest model. It has an accuracy of 98.9% of predicting which mushrooms are edible using the following factors: Stem Width, Gill Attachment, Stem Color, Gill Color, Cap Shape, Stem Height, Cap Diameter and Season. From these factors, the top four that are best at predicting if a mushroom is edible or not base on that factor alone are Stem Width, Gill Attachment, Stem Color, Gill Color ordered from most importance to least importance.

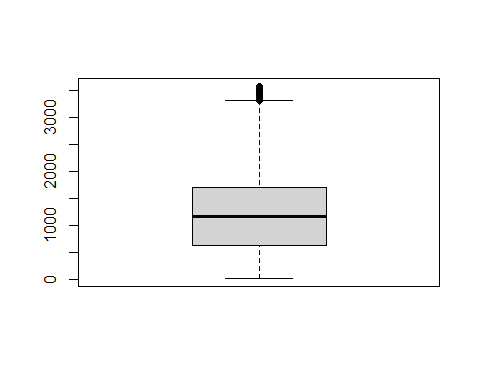
# Extract the top 4 important variables  
top4\_vars <- head(var\_importance\_df[order(-var\_importance\_df$Importance), ], 4)  
print(top4\_vars)

Variable Importance  
stem.width stem.width 4380.789  
gill.attachment gill.attachment 3589.069  
stem.color stem.color 3587.211  
gill.color gill.color 3562.255

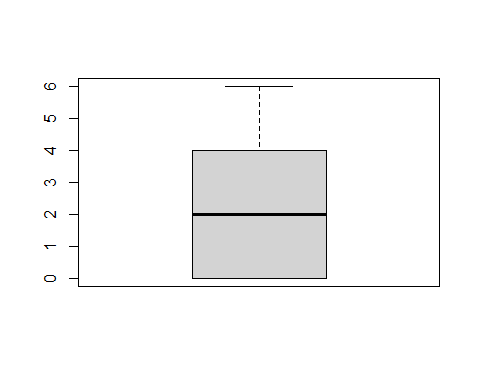
poisonous\_mushrooms <- subset(mushrooms, class == 0)  
summary(poisonous\_mushrooms[, top4\_vars$Variable])

stem.width gill.attachment stem.color gill.color   
 Min. : 21 Min. :0.000 Min. : 0.00 Min. : 0.000   
 1st Qu.: 632 1st Qu.:0.000 1st Qu.: 6.00 1st Qu.: 5.000   
 Median :1157 Median :2.000 Median :11.00 Median :10.000   
 Mean :1209 Mean :2.271 Mean : 8.88 Mean : 7.555   
 3rd Qu.:1705 3rd Qu.:4.000 3rd Qu.:11.00 3rd Qu.:10.000   
 Max. :3569 Max. :6.000 Max. :12.00 Max. :11.000

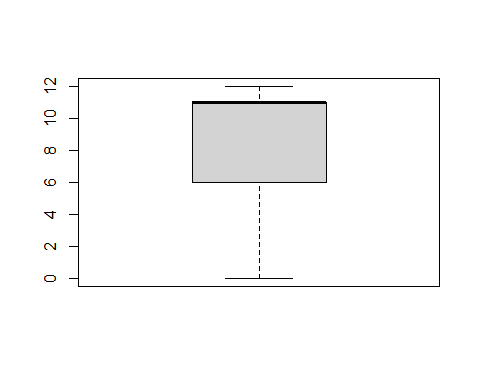
boxplot(poisonous\_mushrooms$stem.width)



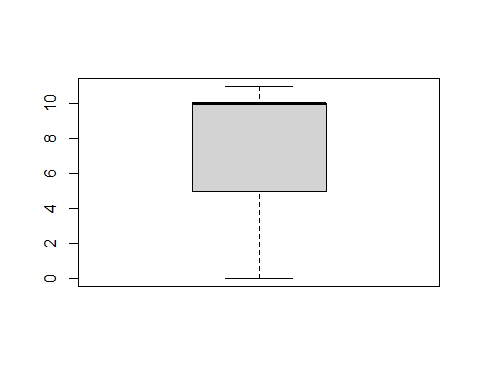
boxplot(poisonous\_mushrooms$gill.attachment)



boxplot(poisonous\_mushrooms$stem.color)



boxplot(poisonous\_mushrooms$gill.color)



We see from the top 4 most important variables that help determine if a mushroom is edible or not have the following in common. A stem width of 1157, gill attachment of 2, stem color of code 11 and gill color of code 10. So if you have a mushroom with any of these features, be safe!

# 6. References

[Dataset](https://www.kaggle.com/datasets/prishasawhney/mushroom-dataset/data). The kaggle version is a cleaned version of the original [Mushroom Dataset for Binary Classification](https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset), available at UCI Library.