Identifying Poisonous Mushrooms Using Logistic Regression

Betty Wu, Jiaman

November 22, 2020

Summary

In the project, I use the logistic regression to model the odds of a mushroom being poisonous on a range of characteristics associated with the mushroom. The goal of the project is to identify important characteristics that are associated with poisonous and edible mushrooms and quantify the relationships. From the results, I find most variables in the dataset have an impact on the odds. Specifically, variables such as spore print color, gill color, gill size, and stalk surface above ring have noticeably large effects. However, there are a number of caveats which concern with the validity of the inferences.

Introduction

Misclassifying poisonous mushrooms as edible can lead to mushroom poisoning. The symptoms of mushroom poisoning can range from slight gastrointestinal discomfort to death. According to American Association of Poison Control Center (AAPCC), there are 7,312 reported cases of plants and mushrooms poisoning in 2016. Among the reported cases of poisoning, 13 cases resulted in death[1]. Therefore, a reliable way of identifying edible mushrooms from poisonous ones is needed for the general public who enjoy foraging and mushroom hunting.

In this project, I am interested in finding characteristics that are important in determining whether a mushroom is edible or poisonous using the logistic regression and quantify the effects. For example, whether a specific cap texture or cap color is predictive of the toxicity of a mushroom and what is the magnitude of the impacts.

Data

1. Data Preparation The dataset is obtained from UCI Machine Learning Repository[2]. It contains 8124 observations on 22 variables. The response variable is a binary variable indicating whether a mushroom is poisonous or edible. Other variables are categorical variables describing other characteristics of mushrooms. For example, the cap size, cap color, gill attachment, habitat, odor, and ring number. This dataset contains no missing value. Overall, it is a balanced dataset with 4208 edible and 3916 poisonous mushrooms.

Upon initial inspection, there are several issues with this dataset that need to be addressed in order to proceed to further analysis. The first issue with the dataset is that there are some variables and groups of variables that perfectly separate poisonous and edible mushrooms. For example, odor is a strong indicator of poisonous and edible mushrooms. All mushrooms with almond or anise smells are edible. A mushroom has 96.6% probability of being edible if it has no odor. All other types of smells indicate poisonous mushrooms. The problem that certain categories in a variables can separate poisonous mushrooms from edible ones exit in most variables. Take the variable gill color for example. It has 12 categories indicating different color values this variable can take. While most categories have observations in both poisonous and edible class, all mushrooms with red or orange gill color are edible and all green gill mushrooms are poisonous. This issue leads to overfitting. In fact, the logistic regression on all predictors result in perfect accuracy and AUC score.

The second issue is that most variables have high cardinality. This means that these variables have many unique levels. Gill color, for example, has 12 levels; habitat, a variable describes where the mushroom is found, has seven categories. In fact, out of the 21 predictors, 11 variables has more than five unique levels. This is problematic for two reasons. First, too many levels tend to result in fewer observations in each level. For example, there are only 24 observations for green gill mushrooms; there are only four observations for mushrooms with the conical cap shape. Too few observation can affect the validity of model inferences. Second, this issue ties back to the first issue that is discussed above. Detailed breakdown of each variables could lead to higher probability of a category only exclusively contain a certain class of mushrooms. For instance, all four observations for conical shaped mushroom are poisonous.

The third issue is that predictors tend to be correlated with one another. For example, all mushrooms with sunken cap shape live in the urban habitat. Furthermore, variables that describe different aspects of the same part tend to be highly correlated. For instance, ring number and ring type have high correlation. Stalk color above ring is also highly correlated with stalk color below ring. This is reasonable because certain mushroom species tend to have certain attributes. However, this is a problem for the logistic regression model. High correlation among predictors leads to inaccurate standard errors. This, in turn, results in inaccurate model inferences.

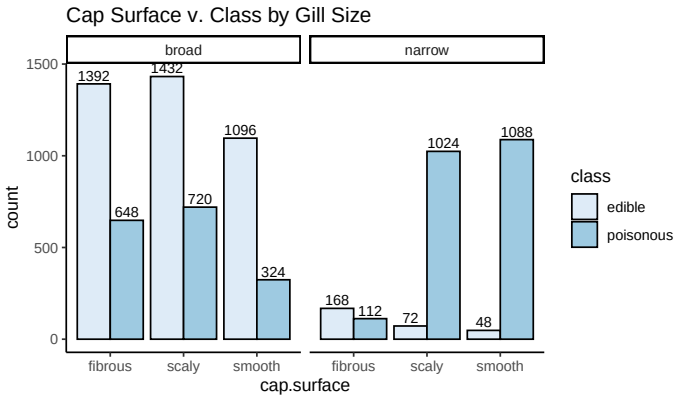
[1] https://eattheplanet.org/foraging-fatality-statistics-2016/
[2] https://archive.ics.uci.edu/ml/datasets/Mushroom

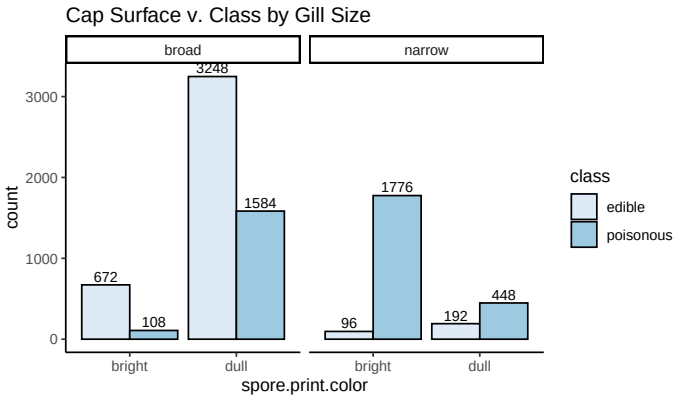
To deal with the above mentioned issues, I implemented mainly the following two solutions. First, for variables with high cardinality, I combined some categories based on both exploratory data analysis and external research on fungal toxicity. Take the `population` variable for example. It indicates the population patterns, and can take six values: `abundant`, `clustered`, `numerous`, `scattered`, `several`, `several`, and `solitary`. This variable is collapsed based on scientific knowledge to contain only two levels: `many` and `few`. The `many` level contains `abundant`, `clustered`, and `numerous`. The `few` level contains the rest of the levels. The re-leveled variable now indicates whether a mushroom tend to grow in close vicinity with others or it tends to grow away from groups. This re-leveling process is done on most variables. After re-leveling, I dropped variables that still can separate the two classes of mushrooms with high accuracy individually. For instance, after re-leveling, odor contains only two levels `odor` and `no odor`. However, it can still predict the toxicity of mushrooms with high accuracy by itself. Therefore, it was dropped as a predictor. Highly correlated variables were also dropped based on exploratory data analysis and scientific importance. For example, `stalk.surface.below.ring` was dropped in favor of `stalk.surface.above.ring`; and `stalk.color.below.ring` was dropped in favor of `stalk.color.above.ring`. These variables are highly correlated. After this process, the predictors are reduced from 21 to 13 variables.

2. Exploratory Data Analysis The reduced dataset has 11 variables that are binary, cap shape and cap surface have three levels, and habitat has five levels. These predictors describe colors, texture, and shape of different parts of a mushrooms. In general, most observations in this dataset have round (4108) or flat (3152) cap shape as opposed to other shapes (864). Roughly 40% (3248) of the mushrooms in the dataset have scaly cap surface, 31% (2556)have smooth surface, and the remaining (2320) have fibrous surface. Mushrooms in this sample have roughly equal divide between bright (4000) and dull (4124) cap color. Most mushrooms have broad gill size and bright gill color. The majority of mushrooms in the sample have white veil color (7924). In terms of habitat, the least amount of mushrooms live in meadows(292), while most observations live in woods (3148).

Due to the nature of the dataset, most predictors have interesting interaction effects that are worth exploring. The graphs below show a selected sample of potential interaction effects between predictors. The plot "Cap Surface v. Class by Gill Size" shows that the relationship between the probability of poisonous mushrooms and the texture of cap surface differ depending on the whether gill size is broad or narrow. Specifically, a broad gill mushroom with scaly cap surface is more likely to be edible, however, it is more likely to be poisonous had its gill was narrow. The plot "Cap Surface v. Class by Gill Size" shows the relationship between mushroom class and spore color differs depending on gill size. In the plot, the probability of a poisonous mushroom by spore print color is completely reversed depending on the gill size.

It is important to note that the plots shown above only display two instances of potential interaction effects. In fact, most of the predictors show varying degrees of differing relationship with the outcome variable by a third variable. In the proceeding model building section, most of the potential interactions are included in the full model. However, because some combinations of variables result in limited numbers of observations or zero observations, it is not possible to explore all interaction effects. Only interactions with reasonable numbers of observations in each combination of variables are included in the full model.





Model

The initial model is given as the following:

$$y_i \mid x_i \sim \ Bernoulli(\pi_i) \log(\frac{\pi_i}{1-\pi_i}) = \boldsymbol{x_i}\boldsymbol{\beta},$$

where y_i is the binary variable indicating whether a mushroom is edible or poisonous. $\boldsymbol{x_i}$ includes all predictors as main effects.

The binned residual plot shows that residuals are roughly scattered around zero. However, most points lie outside the 95% confidence interval. In addition, residuals tend to be positive for low predicted probabilities, and negative for high predicted probabilities. To ameliorate this situation, interactions terms are included in the model. The following interactions are included based on exploratory data analysis:

- `gill.size` * `cap.surface`
- `gill.size` * `spore.print.color`
- `spore.print.color` * `stalk.shape`
- `bruises`*`gill.color`
- `stalk.shape` * `stalk.surface.above.ring`
- `spore.print.color` * `stalk.surface.above.ring`
- `cap.color` * `stalk.surface.above.ring`
- `cap.color` * `gill.color`.

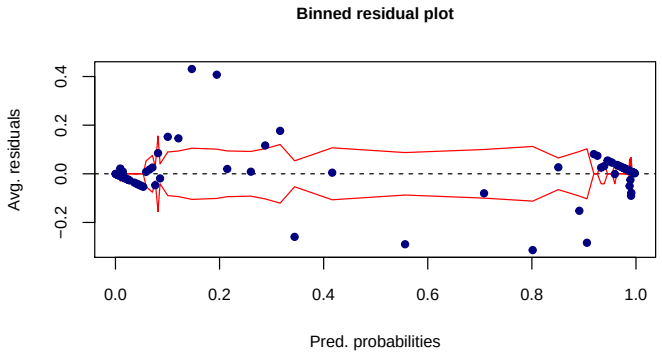
The full model results in improvement in the model fit (better AIC), but only improves the binned residual plot slightly. After using different model selection algorithm with both AIC and BIC criterion, it is found that backward selection using AIC criterion produced a model that has reasonable size and passes the F-test. F-test on the resulting model and the full model produced a large p-value. This means that the dropped variables are not statistically different from zero. They are dropped for the final model.

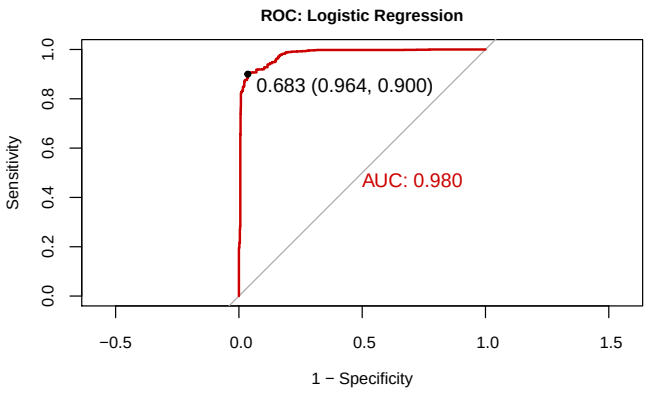
The final model contains all main effects, and the following interactions:

- `gill.size` * `cap.surface`
- `stalk.surface.above.ring` * `stalk.shape`
- `bruises` * `gill.color`
- `spore.print.color` * `stalk.shape`
- `spore.print.color` * `stalk.surface.above.ring`

The binned residual plot for the final model also shows similar patterns as the full model. Specifically, residuals are mostly scattered around zero; residuals tend to be positive for lower predicted probabilities, and negative for higher predicted probabilities. These patterns implies the logistic regression assumption are potentially violated. VIF scores are calculated to assess multicolinearity issues. While most predictors have relatively low scores, the VIF scoare among interactions are large (ranging from 6 to 15). This suggests that there is potential multicolinearity issue.

However, the model preforms reasonably well in terms of prediction. Using 0.5 as the cut-off threshold, a mushroom is predicted to be poisonous if the predicted probability is greater than or equal 0.5, otherwise, edible. The model achieves 0.92 accuracy, 0.91 sensitivity, and 0.94 specificity. This means that the model predicted 92% of the data correctly. 0.91 sensitivity means that given a mushroom is poisonous, the model has 91% probability of predicting it is poisonous. 0.94 specificity means that given a mushroom is edible, the model has 94% probability of predicting it is edible. In addition, the model also achieves the AUC score at 0.978.





Results

The logistic regression results are shown in the table below.

Predictors	Odds Ratios	CI	p
(Intercept)	0.12	0.03 – 0.40	**0.001**
habitat [woods]	0.28	0.20 – 0.40	**<0.001**
habitat [grasses]	0.39	0.28 – 0.55	**<0.001**
habitat [leaves]	0.42	0.23 – 0.80	**0.007**
habitat [meadows]	0.62	0.37 – 1.05	0.077
population [few]	9.90	6.80 – 14.71	**<0.001**
spore.print.color [dull]	44.57	26.23 – 76.62	**<0.001**
stalk.color.above.ring [white or yellow]	0.28	0.19 – 0.39	**<0.001**
stalk.surface.above.ring [smooth]	0.02	0.01 – 0.03	**<0.001**
gill.size [narrow]	7.40	3.11 – 17.86	**<0.001**
gill.color [dull]	0.40	0.29 – 0.56	**<0.001**
stalk.shape [tapering]	0.92	0.44 – 1.91	0.831
cap.shape [other]	0.66	0.46 – 0.96	**0.031**
cap.shape [flat]	0.94	0.75 – 1.17	0.569
cap.surface [scaly]	3.60	2.37 – 5.62	**<0.001**
cap.surface [smooth]	18.94	12.10 – 30.50	**<0.001**
cap.color [dull]	0.62	0.50 – 0.77	**<0.001**
bruises [t]	2.53	1.72 – 3.77	**<0.001**
veil.color [white]	3.20	1.17 – 9.47	**0.028**
gill.size [narrow] * cap.surface [scaly]	2.03	1.08 – 3.78	**0.027**
gill.size [narrow] * cap.surface [smooth]	0.58	0.30 – 1.10	0.096
spore.print.color [dull] * gill.size [narrow]	17.58	7.57 – 40.83	**<0.001**
spore.print.color [dull] * stalk.shape [tapering]	0.01	0.00 – 0.01	**<0.001**
spore.print.color [dull] * stalk.surface.above.ring [smooth]	0.02	0.01 – 0.03	**<0.001**
gill.color [dull] * bruises [t]	2.04	1.30 – 3.22	**0.002**
stalk.surface.above.ring [smooth] * stalk.shape [tapering]	98.48	55.34 – 177.23	**<0.001**
Observations	8124		
R^2 Tjur	0.790		

All but four variables are significant at 5% significant level. For conciseness, in this report, only the most interesting results are interpreted. I find that all else equal, compared to mushrooms grow in artificial habitats, mushrooms grow in natural habitats (except for meadows) have lower odds of being poisonous. Specifically, compared to artificial habitats, the odds of a mushroom being poisonous is multiplied by 0.28, 0.39, and 0.42 for mushrooms grow in the woods, grasses, and leaves, respectively. Furthermore, compared to mushrooms that tend to grow in groups, the odds of a mushroom is poisonous if it tends to grow away from others is multiplied by 9.9.

Spore print color is a highly significant variable with large coefficients. The main effect for spore print color is 44.57. This means that holding other variables constant, for a mushroom with dull spore print color, broad gill, enlarging stalk shape, and unsmooth stalk surface above ring, the odds of it being poisonous is multiplied by 44.57 compared to that of the bright spore print color mushrooms.

It is interesting to find that although stalk shape is not significant by itself, but the interaction effect with spore print color is highly significant. Specifically, given a mushroom has dull spore print color, the odds of it being poisonous decreases by 0.59 if it has tapering stalk shape compared to that of enlarging stalk shape mushrooms.

Given a mushrooms has enlarging stalk shape, compared unsmooth stalk surface above ring, the odds of a smooth stalk surface above ring mushroom being poisonous decreases by 0.98 all else the same. However, if it has both smooth stalk surface above ring and tapering stalk shape, the odds of it being poisonous is multiplied by 1.81(0.02*0.92*98.48) compared to that of unsmooth stalk surface above ring and enlarging stalk shape (note, the mushroom should also has bright spore print color).

Conclusion

In this project, I use logistic regression to identify characteristics that are important in determining whether a mushroom is poisonous or edible. From regression results, it is find that almost all predictors are significant at 5% significance level. It seems like factors ranging from habitat, population pattern to spore print color, surface texture, colors all play

a role in determining the toxicity of mushrooms. For `bruises` and `stalk.shape`, even though the main effects are not significant, their interactions with other variables are significant.

There are a few variables and interactions that have particularly large effect on the odds of a mushroom being poisonous. For example, the main effect for spore print color is 44.57. The coefficient of the interaction `stalk.surface.above.ring` and `stalk.shape` is 98.48. And the main effect for `stalk.surface.above.ring` is 0.02. These extreme coefficients are problematic. The predicted odds will be extreme as well as the result of being multiplied by these extreme multipliers.

Overall, the model has high accuracy at 0.927 and high AUC at 0.98. Despite the excellent predictive accuracy, the model suffers from a few major drawbacks. The data that is used to train the model is problematic. As mentioned in the Data section, there are variables or groups of variables that *can* perfectly separate the two classes of mushrooms. With information from EDA and external research, I find that in most cases, the toxicity of mushrooms can be determined based on physical appearances. Therefore, this dataset would be more informative and interesting if it is used for data visualization purposes. Other limitations of the dataset are variables tend to be correlated with each other. Despite my effort to drop variables that are highly correlated, it is inevitable that some variables are still highly correlated due to the nature of the variables. Therefore, the model potentially suffer from multicolinearity as well. In addition, the categorical nature of all predictors inhibit the possibility of exploring some of the potential interactions due the lack of data in some combination of categories.

The consequences of these limitations are manifested in different ways. First, the residual plot shows many points are outside of the 95% confidence interval bounds and potential patterns that could not be ameliorated. Second, relatively high VIF scores for some variables and especially for interactions. Third, most estimated coefficients are statistically significant. And they tend to take extreme values. It is likely that extremely small coefficients are compensating for extremely large coefficients, and vice versa. These suggest that the inferences using this model could be dubious.

Potential future work for this project could be applying regularization to the logistic model. This could "shrink" the effects on variables that are "unimportant." By doing this, we could have a model that is more informative for making inferences.

Appendix

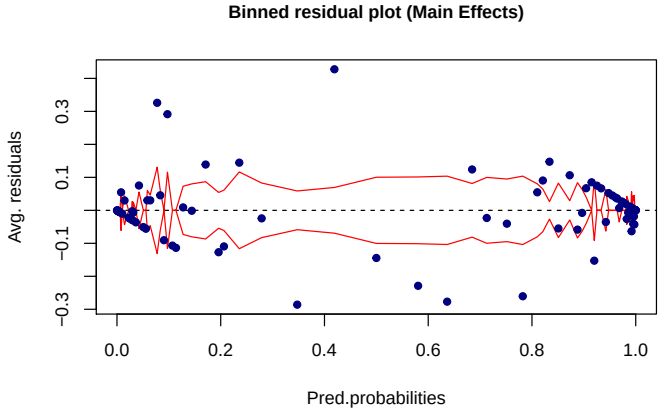
Github repo: https://github.com/JiamanBettyWu/mushroom-classfication-702

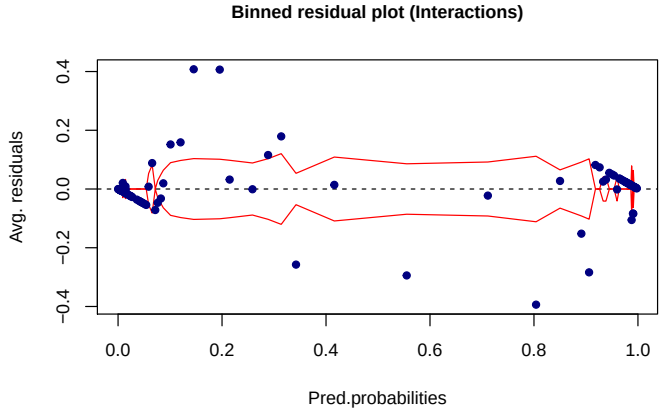
Summary Statistics

	cap.shape	cap.surface	cap.color	bruises	gill.size	gill.color	stalk.shape	stalk.surface.above.ring
	round:4108	fibrous:2320	bright:4000	f:4748	broad :5612	bright:5184	enlarging:3516	unsmooth:2924
	other: 864	scaly :3248	dull :4124	t:3376	narrow:2512	dull :2940	tapering :4608	smooth :5200
	flat :3152	smooth :2556						

	stalk.color.above.ring	veil.color	spore.print.color	population	habitat
	other :3652	nonwhite: 200	bright:2652	many:1124	artificial:1704
	white or yellow:4472	white :7924	dull :5472	few :7000	woods :3148
					grasses :2148
					leaves : 832
					meadows : 292

Binned Residual Plots





R Appendix

```
knitr::opts_chunk$set(echo = TRUE)
# opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
library(tinytex)
library(knitr)
library(gridExtra)
# library(kableExtra)
library(ggplot2)
library(sjPlot)
library(caret)
library(pROC)
library(rms)
library(arm)
library(stargazer)
library(dplyr)
# read original data

dt = read.csv("~/Desktop/ids_702/assignment/mushroom-classfication-702/Data/mushrooms.csv", stringsAsFactors
# remove columns
dt = dt[, !colnames(dt) %in% "veil.type"]
# veil.type is removed because every row has the same category

## DATA CLEANING

dt = dt[, !colnames(dt) %in%  "odor"]
# odor remove because perfect separation


dt$class = factor(dt$class, levels = c("e", "p"), labels = c("0", "1")) # convert class to 0, 1 variable

dt$population = factor(dt$population,
       levels = c("a", "c", "n", "s", "v", "y"),
       labels = c("many", "many", "many", "few", "few", "few"))

dt$ring.number = factor(dt$ring.number,
       levels = c("n", "o", "t"),
       labels = c("less than 2", "less than 2", "two"))

dt$stalk.color.above.ring = factor(dt$stalk.color.above.ring,
                         levels = c("b", "c", "e", "g", "n", "o", "p", "w", "y"),
                         labels = c("other","other","other","other","other","other","other", "white or yellow

dt$stalk.surface.above.ring = factor(dt$stalk.surface.above.ring,
       levels = levels(dt$stalk.surface.above.ring),
       labels = c("unsmooth", "unsmooth", "smooth", "smooth"))

dt$cap.color = factor(dt$cap.color,
       levels = levels(dt$cap.color),
       labels = c("bright", "bright","bright","dull","dull", "bright", "bright","bright","bright", "bright"

dt$cap.surface = factor(dt$cap.surface,
       levels = levels(dt$cap.surface),
       labels = c("fibrous", "scaly","smooth","scaly"))

dt$cap.shape = factor(dt$cap.shape,
       levels = levels(dt$cap.shape),
```

```
       labels = c("bell", "conical","flat","conical", "sunken", "convex"))

dt$cap.shape = factor(dt$cap.shape,
       levels = levels(dt$cap.shape),
       labels = c("round", "other","flat","other", "round"))

dt$habitat = factor(dt$habitat,
       levels = levels(dt$habitat),
       labels = c("woods", "grasses","leaves","meadows", "artificial", "artificial", "artificial"))
dt$habitat = relevel(dt$habitat, ref = "artificial")

dt$spore.print.color = factor(dt$spore.print.color,
       levels = levels(dt$spore.print.color),
       labels = c("bright", "dull","dull","dull", "bright", "bright", "bright", "bright", "bright"))


dt$gill.color = factor(dt$gill.color,
       levels = levels(dt$gill.color),
       labels = c("bright", "bright","dull","dull", "dull", "dull", "bright", "bright", "bright", "bright",

dt$veil.color = factor(dt$veil.color,
       levels = levels(dt$veil.color),
       labels = c("nonwhite", "nonwhite", "white", "nonwhite"))

dt$gill.size = factor(dt$gill.size,
                      levels = levels(dt$gill.size),
                      labels = c("broad", "narrow"))

dt$stalk.shape = factor(dt$stalk.shape,
                      levels = levels(dt$stalk.shape),
                      labels = c("enlarging", "tapering"))
variables = c("class", "habitat" , "population", "spore.print.color",
  "stalk.color.above.ring", "stalk.surface.above.ring",
  "gill.size", "gill.color", "stalk.shape",
  "cap.shape", "cap.surface", "cap.color", "bruises", "veil.color")

dt_r = dt[,colnames(dt) %in%  variables]

# options(knitr.kable.NA = '')
# kable(summary(dt_r[,2:9]))
# kable(summary(dt_r[,10:ncol(dt_r)]))
# kable(summary(dt_r))%>% column_spec(1, width = "10em")
p1 = ggplot(dt, aes(cap.surface, group = class, fill = class)) +
  geom_bar(color = "black", position = "dodge") +
  geom_text(aes(label = ..count..), stat = "count", vjust = -0.3, position = position_dodge(0.9), size = 3)
  facet_wrap(~gill.size) +
  scale_fill_brewer(palette="Blues", labels = c("edible", "poisonous")) +
  theme_classic(base_size = 11) +
  labs(title = "Cap Surface v. Class by Gill Size")


p2 = ggplot(dt, aes(spore.print.color, group = class, fill = class)) +
  geom_bar(color = "black", position = "dodge") +
  geom_text(aes(label = ..count..), stat = "count", vjust = -0.3, position = position_dodge(0.9), size = 3)
  facet_wrap(~gill.size) +
  scale_fill_brewer(palette="Blues", labels = c("edible", "poisonous")) +
  theme_classic(base_size = 11) +
```

```
  labs(title = "Cap Surface v. Class by Gill Size")

grid.arrange(p1, p2, ncol = 2)


formulam = class ~habitat + population  + spore.print.color +
  stalk.color.above.ring + stalk.surface.above.ring +
  gill.size + gill.color + stalk.shape +
  cap.shape + cap.surface + cap.color + bruises + veil.color

modelm = glm(formulam, data = dt, family = binomial)

formulaf = class ~habitat + population  + spore.print.color +
  stalk.color.above.ring + stalk.surface.above.ring +
  gill.size + gill.color + stalk.shape +
  cap.shape + cap.surface + cap.color + bruises + veil.color +
  gill.size * cap.surface +
  gill.size * spore.print.color +
  spore.print.color * stalk.shape +
  bruises*gill.color +
  stalk.shape * stalk.surface.above.ring +
  spore.print.color * stalk.surface.above.ring +
  cap.color * stalk.surface.above.ring +
  cap.color * gill.color

modelf = glm(formulaf, data = dt, family = binomial)

NullModel = glm(class ~ 1, data = dt, family = binomial)

# Model_forward = step(NullModel, scope = formula(modelf),direction="forward",trace=0)
# Model_forward$formula

Model_backward = step(modelf,direction="backward",trace=0)
#Model_backward$formula


model_bic = glm(Model_backward$formula, data = dt, family = binomial)

# model_back = glm(Model_backward$formula, dt, family = binomial)
# summary(model_bic)
# anova(modelf, model_back, test = "Chisq")
par(mfrow=c(1,2))

arm::binnedplot(fitted(model_bic),residuals(model_bic,"resp"),
                xlab="Pred. probabilities",col.int="red",
                ylab="Avg. residuals",
                main="Binned residual plot",
                col.pts="navy",cex.lab = .8,cex.axis = .8,cex.main = .8,cex.sub = .8)

Conf_mat <- confusionMatrix(as.factor(ifelse(fitted(model_bic) >= 0.5, "1","0")),
                            dt$class,positive = "1")
# Conf_mat$table
# Conf_mat$overall["Accuracy"]
# Conf_mat$byClass[c("Sensitivity","Specificity")]
roc(dt$class, fitted(model_bic),plot=T,print.thres="best",legacy.axes=T,
    print.auc =T,col="red3", main = "ROC: Logistic Regression", cex.lab = .8,cex.axis = .8,cex.main = .8,cex
```

```r
options(knitr.kable.NA = '')
kable(summary(dt_r[,2:9]))
kable(summary(dt_r[,10:ncol(dt_r)]))
# summary(model)
par(mfrow=c(1,2))
arm::binnedplot(fitted(modelm),residuals(modelm,"resp"),
                xlab="Pred.probabilities",col.int="red",
                ylab="Avg. residuals",
                main="Binned residual plot (Main Effects)",
                col.pts="navy",cex.lab = .9,cex.axis = .9,cex.main = .9,cex.sub = .9)
# summary(model)

arm::binnedplot(fitted(modelf),residuals(modelf,"resp"),
                xlab="Pred.probabilities",col.int="red",
                ylab="Avg. residuals",
                main="Binned residual plot (Interactions)",
                col.pts="navy",cex.lab = .9,cex.axis = .9,cex.main = .9,cex.sub = .9)
```