HarvardX: PH125.9x Professional Certificate in Data Science - Choose Your Own Project

Stefan Prangenberg

6/15/2019

Introduction

The goal of this project is to predict if mushrooms are edible or poisonous - based on their numerous attributes such as cap color ornumber of rings. Since we want to avoid eating a poisonous mushroom, we strive for 100% accuracy.

This project and the source core are available online at https://github.com/prangberg/MushroomClassification (https://github.com/prangberg/MushroomClassification)

This report is part of the final course of the HarvardX Professional Certificate in Data Science. The goal is to apply machine learning techniques that go beyond standard linear regression.

I downloaded the data from Kaggle https://www.kaggle.com/uciml/mushroom-classification (https://www.kaggle.com/uciml/mushroom-classification) and saved it as a .csv in a local folder.

This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like “leaflets three, let it be” for Poisonous Oak and Ivy.

Analysis

We use the folloring libraries for our analysis:

```
library(tidyverse)
library(caret)
library(randomForest)
library(ggplot2)
```

First, we ingest the CSV

```
mushrooms <- read.csv("mushrooms.csv", colClasses = "character")
```

Note: The method read.csv worked better for me than read_csv, since the later tried to convert ‘bruises’ and ‘gill-attachment’ to a logical variable, which caused problems.

```
dim(mushrooms)
```

```
## [1] 8124   23
```

The dataset has 8124 entries with 23 columns.

```
glimpse(mushrooms)
```

```
## Observations: 8,124
## Variables: 23
## $ class                    <chr> "p", "e", "e", "p", "e", "e", "e", "e...
## $ cap.shape                <chr> "x", "x", "b", "x", "x", "x", "b", "b...
## $ cap.surface              <chr> "s", "s", "s", "y", "s", "y", "s", "y...
## $ cap.color                <chr> "n", "y", "w", "w", "g", "y", "w", "w...
## $ bruises                  <chr> "t", "t", "t", "t", "f", "t", "t", "t...
## $ odor                     <chr> "p", "a", "l", "p", "n", "a", "a", "l...
## $ gill.attachment          <chr> "f", "f", "f", "f", "f", "f", "f", "f...
## $ gill.spacing             <chr> "c", "c", "c", "c", "w", "c", "c", "c...
## $ gill.size                <chr> "n", "b", "b", "n", "b", "b", "b", "b...
## $ gill.color               <chr> "k", "k", "n", "n", "k", "n", "g", "n...
## $ stalk.shape              <chr> "e", "e", "e", "e", "t", "e", "e", "e...
## $ stalk.root               <chr> "e", "c", "c", "e", "e", "c", "c", "c...
## $ stalk.surface.above.ring <chr> "s", "s", "s", "s", "s", "s", "s", "s...
## $ stalk.surface.below.ring <chr> "s", "s", "s", "s", "s", "s", "s", "s...
## $ stalk.color.above.ring   <chr> "w", "w", "w", "w", "w", "w", "w", "w...
## $ stalk.color.below.ring   <chr> "w", "w", "w", "w", "w", "w", "w", "w...
## $ veil.type                <chr> "p", "p", "p", "p", "p", "p", "p", "p...
## $ veil.color               <chr> "w", "w", "w", "w", "w", "w", "w", "w...
## $ ring.number              <chr> "o", "o", "o", "o", "o", "o", "o", "o...
## $ ring.type                <chr> "p", "p", "p", "p", "e", "p", "p", "p...
## $ spore.print.color        <chr> "k", "n", "n", "k", "n", "k", "k", "n...
## $ population               <chr> "s", "n", "n", "s", "a", "n", "n", "s...
## $ habitat                  <chr> "u", "g", "m", "u", "g", "g", "m", "m...
```

The values are single letter. We can find the meaning in the definition on kaggle.com: The columns are already labeled, but the single letter abbreviations are not very meaningful.

Definition of columns from kaggle

- classes: edible=e, poisonous=p
- cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
- cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
- cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y
- bruises: bruises=t,no=f
- odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s
- gill-attachment: attached=a,descending=d,free=f,notched=n
- gill-spacing: close=c,crowded=w,distant=d
- gill-size: broad=b,narrow=n
- gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g,green=r,orange=o,pink=p,purple=u,red=e,white=w,yellow=y
- stalk-shape: enlarging=e,tapering=t
- stalk-root: bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=?
- stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
- stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
- stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- veil-type: partial=p,universal=u
- veil-color: brown=n,orange=o,white=w,yellow=y
- ring-number: none=n,one=o,two=t
- ring-type: cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z
- spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y
- population: abundant=a,clustered=c,numerous=n,scattered=s,several=v,solitary=y
- habitat: grasses=g,leaves=l,meadows=m,paths=p,urban=u,waste=w,woods=d

Data Cleaning

Labeling the columns: We'll create human-friendly names for each category. First we map the data to a factor:

```
mushrooms <- mushrooms %>% map_df(function(.x) as.factor(.x))
str(mushrooms)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    8124 obs. of  23 variables:
##  $ class                   : Factor w/ 2 levels "e","p": 2 1 1 2 1 1 1 1 2 1 ...
##  $ cap.shape               : Factor w/ 6 levels "b","c","f","k",..: 6 6 1 6 6 6 1 1 6
1 ...
##  $ cap.surface             : Factor w/ 4 levels "f","g","s","y": 3 3 3 4 3 4 3 4 4 3
...
##  $ cap.color               : Factor w/ 10 levels "b","c","e","g",..: 5 10 9 9 4 10 9
9 9 10 ...
##  $ bruises                 : Factor w/ 2 levels "f","t": 2 2 2 2 1 2 2 2 2 2 ...
##  $ odor                    : Factor w/ 9 levels "a","c","f","l",..: 7 1 4 7 6 1 1 4 7
1 ...
##  $ gill.attachment         : Factor w/ 2 levels "a","f": 2 2 2 2 2 2 2 2 2 2 ...
##  $ gill.spacing            : Factor w/ 2 levels "c","w": 1 1 1 1 2 1 1 1 1 1 ...
##  $ gill.size               : Factor w/ 2 levels "b","n": 2 1 1 2 1 1 1 1 2 1 ...
##  $ gill.color              : Factor w/ 12 levels "b","e","g","h",..: 5 5 6 6 5 6 3 6
8 3 ...
##  $ stalk.shape             : Factor w/ 2 levels "e","t": 1 1 1 1 2 1 1 1 1 1 ...
##  $ stalk.root              : Factor w/ 5 levels "?","b","c","e",..: 4 3 3 4 4 3 3 3 4
3 ...
##  $ stalk.surface.above.ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3 3 3 3 3 3
...
##  $ stalk.surface.below.ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3 3 3 3 3 3
...
##  $ stalk.color.above.ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8 8 8 8 8 8 8
8 ...
##  $ stalk.color.below.ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8 8 8 8 8 8 8
8 ...
##  $ veil.type               : Factor w/ 1 level "p": 1 1 1 1 1 1 1 1 1 1 ...
##  $ veil.color              : Factor w/ 4 levels "n","o","w","y": 3 3 3 3 3 3 3 3 3 3
...
##  $ ring.number             : Factor w/ 3 levels "n","o","t": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ring.type               : Factor w/ 5 levels "e","f","l","n",..: 5 5 5 5 1 5 5 5 5
5 ...
##  $ spore.print.color       : Factor w/ 9 levels "b","h","k","n",..: 3 4 4 3 4 3 3 4 3
3 ...
##  $ population              : Factor w/ 6 levels "a","c","n","s",..: 4 3 3 4 1 3 3 4 5
4 ...
##  $ habitat                 : Factor w/ 7 levels "d","g","l","m",..: 6 2 4 6 2 2 4 4 2
4 ...
```

Notice that “veil.type” only has one single level, so this variable is fairly useless for any analysis and we can ignore it. In a larger dataset it would make sense to remove it in order to reduce complexity and improve performance, but in this small dataset I omit this step. Every other variabel has 2-12 different levels.

For each variable we define meaningful (and easy to understand) variable values, baszed on the definitions on Kaggle.

```
levels(mushrooms$class) <- c("edible", "poisonous")
levels(mushrooms$cap.shape) <- c("bell", "conical", "flat", "knobbed", "sunken", "conve
x")
levels(mushrooms$cap.color) <- c("buff", "cinnamon", "red", "gray", "brown", "pink", "gr
een", "purple", "white", "yellow")
levels(mushrooms$cap.surface) <- c("fibrous", "grooves", "scaly", "smooth")
levels(mushrooms$bruises) <- c("no", "yes")
levels(mushrooms$odor) <- c("almond", "creosote", "foul", "anise", "musty", "none", "pun
gent", "spicy", "fishy")
levels(mushrooms$gill.attachment) <- c("attached", "free")
levels(mushrooms$gill.spacing) <- c("close", "crowded")
levels(mushrooms$gill.size) <- c("broad", "narrow")
levels(mushrooms$gill.color) <- c("buff", "red", "gray", "chocolate", "black", "brown",
"orange", "pink", "green", "purple", "white", "yellow")
levels(mushrooms$stalk.shape) <- c("enlarging", "tapering")
levels(mushrooms$stalk.root) <- c("missing", "bulbous", "club", "equal", "rooted")
levels(mushrooms$stalk.surface.above.ring) <- c("fibrous", "silky", "smooth", "scaly")
levels(mushrooms$stalk.surface.below.ring) <- c("fibrous", "silky", "smooth", "scaly")
levels(mushrooms$stalk.color.above.ring) <- c("buff", "cinnamon", "red", "gray", "brown"
, "pink", "green", "purple", "white", "yellow")
levels(mushrooms$stalk.color.below.ring) <- c("buff", "cinnamon", "red", "gray", "brown"
, "pink", "green", "purple", "white", "yellow")
levels(mushrooms$veil.type) <- "partial"
levels(mushrooms$veil.color) <- c("brown", "orange", "white", "yellow")
levels(mushrooms$ring.number) <- c("none", "one", "two")
levels(mushrooms$ring.type) <- c("evanescent", "flaring", "large", "none", "pendant")
levels(mushrooms$spore.print.color) <- c("buff", "chocolate", "black", "brown", "orange"
, "green", "purple", "white", "yellow")
levels(mushrooms$population) <- c("abundant", "clustered", "numerous", "scattered", "sev
eral", "solitary")
levels(mushrooms$habitat) <- c("wood", "grasses", "leaves", "meadows", "paths", "urban",
"waste")

str(mushrooms)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    8124 obs. of  23 variables:
##  $ class                   : Factor w/ 2 levels "edible","poisonous": 2 1 1 2 1 1 1 1
2 1 ...
##  $ cap.shape               : Factor w/ 6 levels "bell","conical",..: 6 6 1 6 6 6 1 1
6 1 ...
##  $ cap.surface             : Factor w/ 4 levels "fibrous","grooves",..: 3 3 3 4 3 4 3
4 4 3 ...
##  $ cap.color               : Factor w/ 10 levels "buff","cinnamon",..: 5 10 9 9 4 10
9 9 9 10 ...
##  $ bruises                 : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
##  $ odor                    : Factor w/ 9 levels "almond","creosote",..: 7 1 4 7 6 1 1
4 7 1 ...
##  $ gill.attachment         : Factor w/ 2 levels "attached","free": 2 2 2 2 2 2 2 2 2
2 ...
##  $ gill.spacing            : Factor w/ 2 levels "close","crowded": 1 1 1 1 2 1 1 1 1
1 ...
##  $ gill.size               : Factor w/ 2 levels "broad","narrow": 2 1 1 2 1 1 1 1 2 1
...
##  $ gill.color              : Factor w/ 12 levels "buff","red","gray",..: 5 5 6 6 5 6
3 6 8 3 ...
##  $ stalk.shape             : Factor w/ 2 levels "enlarging","tapering": 1 1 1 1 2 1 1
1 1 1 ...
##  $ stalk.root              : Factor w/ 5 levels "missing","bulbous",..: 4 3 3 4 4 3 3
3 4 3 ...
##  $ stalk.surface.above.ring: Factor w/ 4 levels "fibrous","silky",..: 3 3 3 3 3 3 3 3
3 3 ...
##  $ stalk.surface.below.ring: Factor w/ 4 levels "fibrous","silky",..: 3 3 3 3 3 3 3 3
3 3 ...
##  $ stalk.color.above.ring  : Factor w/ 10 levels "buff","cinnamon",..: 8 8 8 8 8 8 8
8 8 8 ...
##  $ stalk.color.below.ring  : Factor w/ 10 levels "buff","cinnamon",..: 8 8 8 8 8 8 8
8 8 8 ...
##  $ veil.type               : Factor w/ 1 level "partial": 1 1 1 1 1 1 1 1 1 1 ...
##  $ veil.color              : Factor w/ 4 levels "brown","orange",..: 3 3 3 3 3 3 3 3
3 3 ...
##  $ ring.number             : Factor w/ 3 levels "none","one","two": 2 2 2 2 2 2 2 2 2
2 ...
##  $ ring.type               : Factor w/ 5 levels "evanescent","flaring",..: 5 5 5 5 1
5 5 5 5 5 ...
##  $ spore.print.color       : Factor w/ 9 levels "buff","chocolate",..: 3 4 4 3 4 3 3
4 3 3 ...
##  $ population              : Factor w/ 6 levels "abundant","clustered",..: 4 3 3 4 1
3 3 4 5 4 ...
##  $ habitat                 : Factor w/ 7 levels "wood","grasses",..: 6 2 4 6 2 2 4 4
2 4 ...
```

The data is looking good - we're ready to explore it

Data Exploration and Visualization

The most important information for the person finding a mushroom is weather it is ebible or poisonous.

```
## [1] 8124   23
```

```
##           x freq
## 1    edible 4208
## 2 poisonous 3916
```

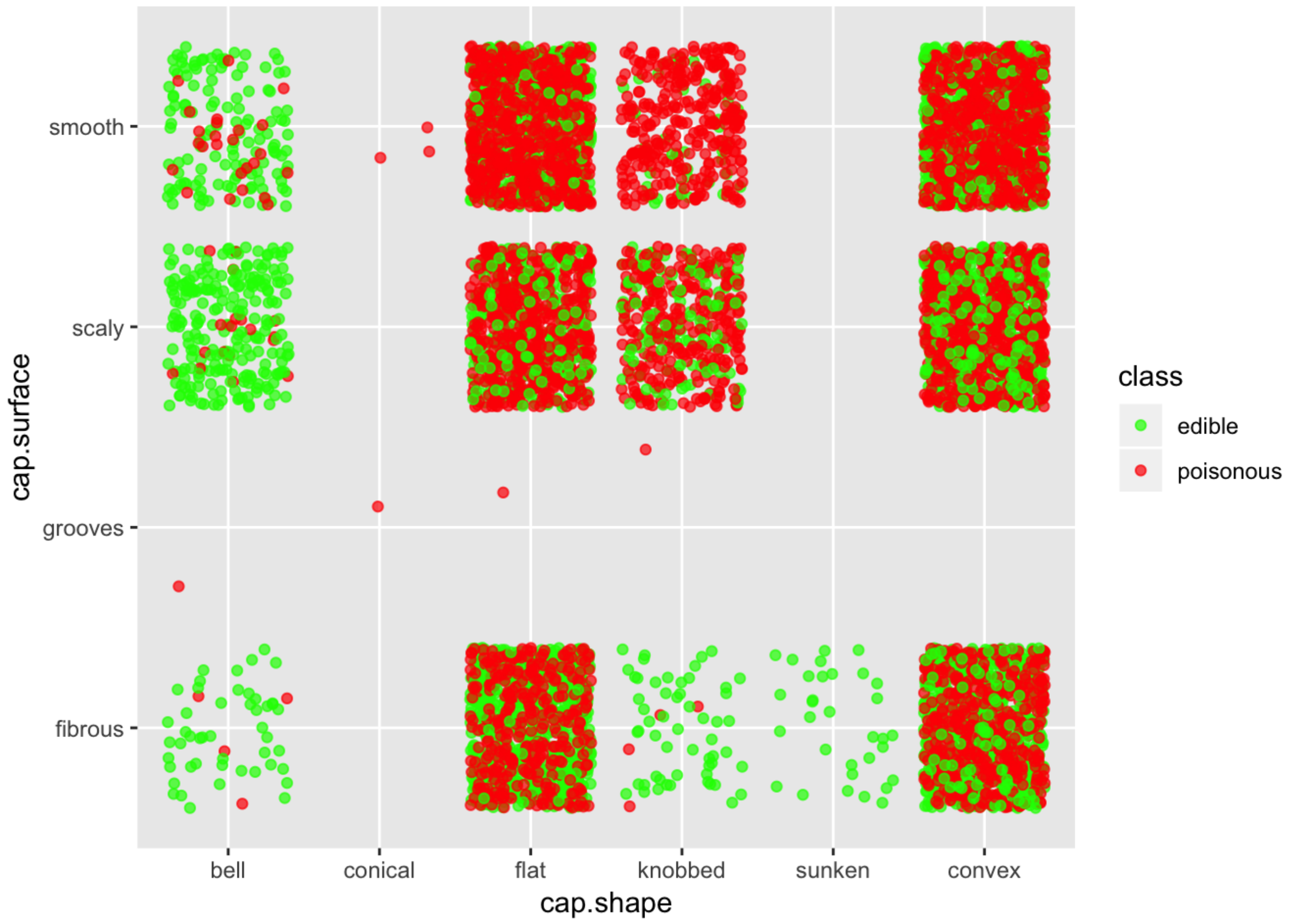
Almost half the mushrooms (48.2%) are poisonous.

We'll use ggplot2 to create some visualizations of various attributes and how the relate to edibility.

Checking out the first two attributes:

Cap Shape and CapSurface

```
ggplot(mushrooms, aes(x = cap.shape, y = cap.surface, col = class)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red")) +
  geom_jitter(alpha = 0.7)
```

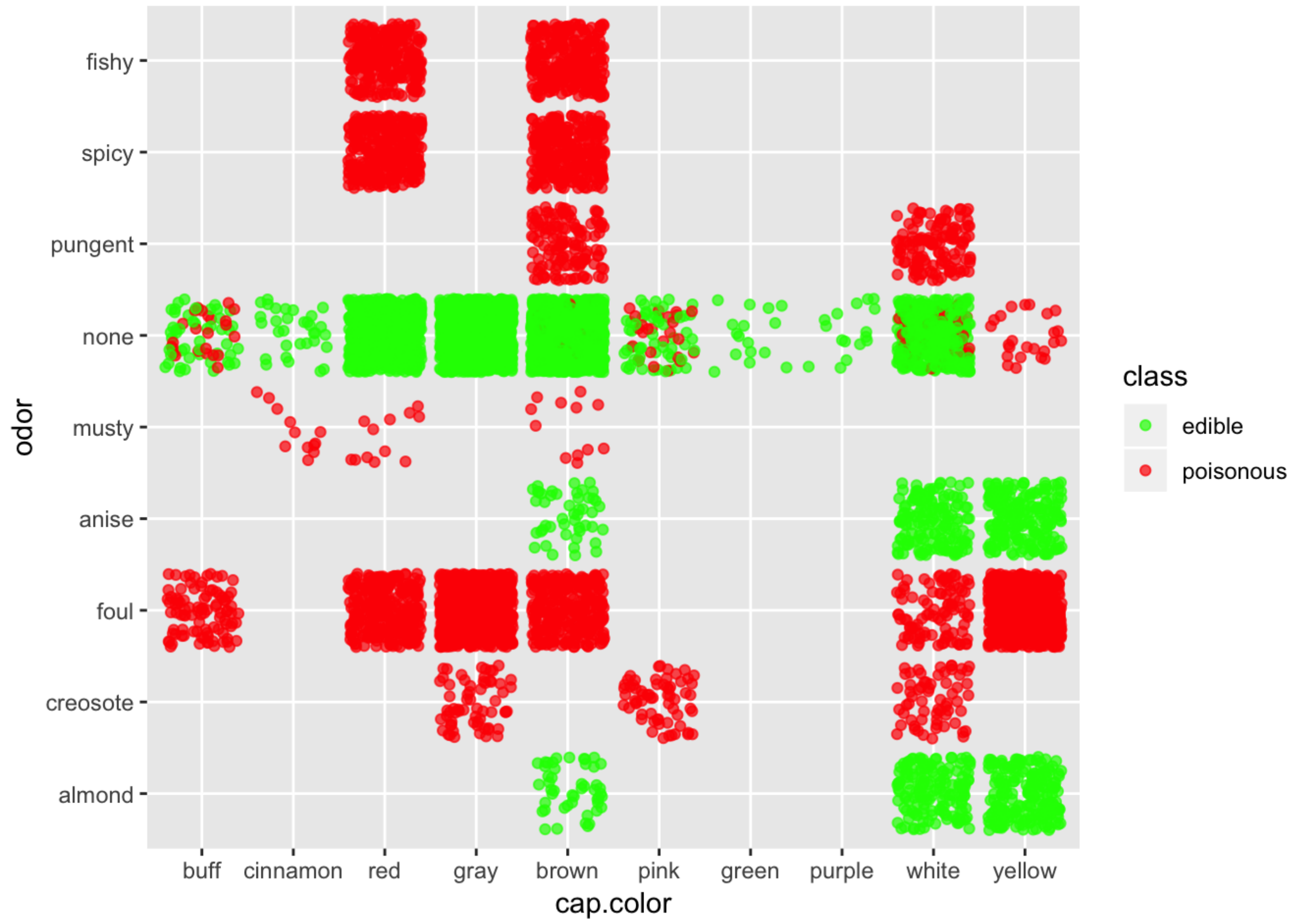


We can see a couple of things in this plot: There is a very low (one-digit) number of mushrooms that have cap surfaces with grooves or conical cap shapes. All of those are poisonous Mushrooms with bell shaped caps are mostly edible. Those with flat or convex cap shapes are mostly poisonous.

Checking out the next attributes:

Cap Color and Odor

```
ggplot(mushrooms, aes(x = cap.color, y = odor, col = class)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red")) +
  geom_jitter(alpha = 0.7)
```

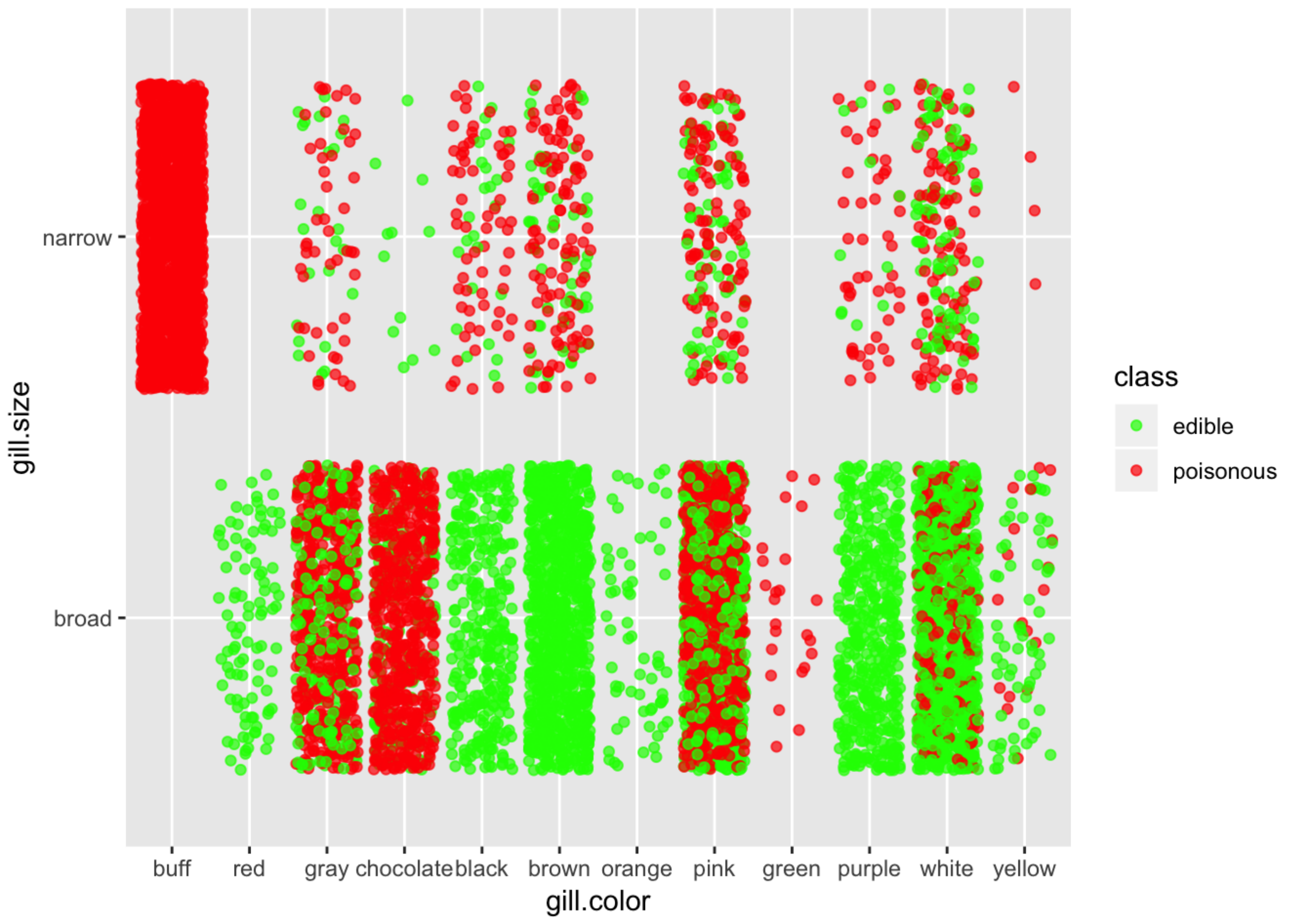


You definitely want to avoid any odor that is fishy, spicy, pungent, musty, foul or creosote. Anise and almond seem to be very safe. Mushrooms with no odor are mostly safe, but some are poisonous.

Checking out more attributes:

Gill Color and Gill Size

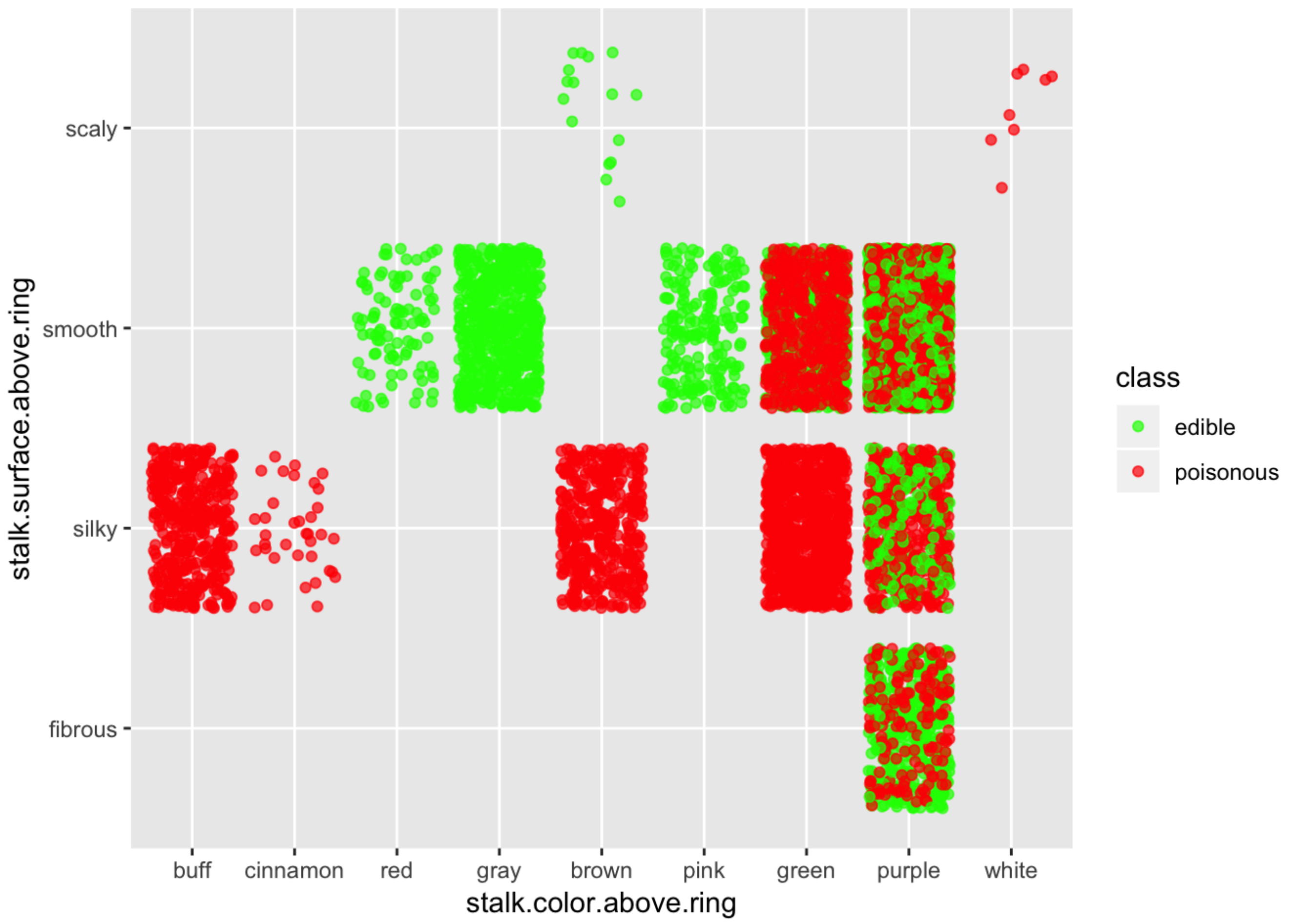
```
ggplot(mushrooms, aes(x = gill.color, y = gill.size, col = class)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red")) +
  geom_jitter(alpha = 0.7)
```



The Gill Colors red and orange are safe. Green is always poisonous. If the Gill size is groad the following colors are safe too: black,brown, orange, purple.

Stark color above and below the rind

```
ggplot(mushrooms, aes(x = stalk.color.above.ring , y = stalk.surface.above.ring, col = c
lass)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red")) +
  geom_jitter(alpha = 0.7)
```

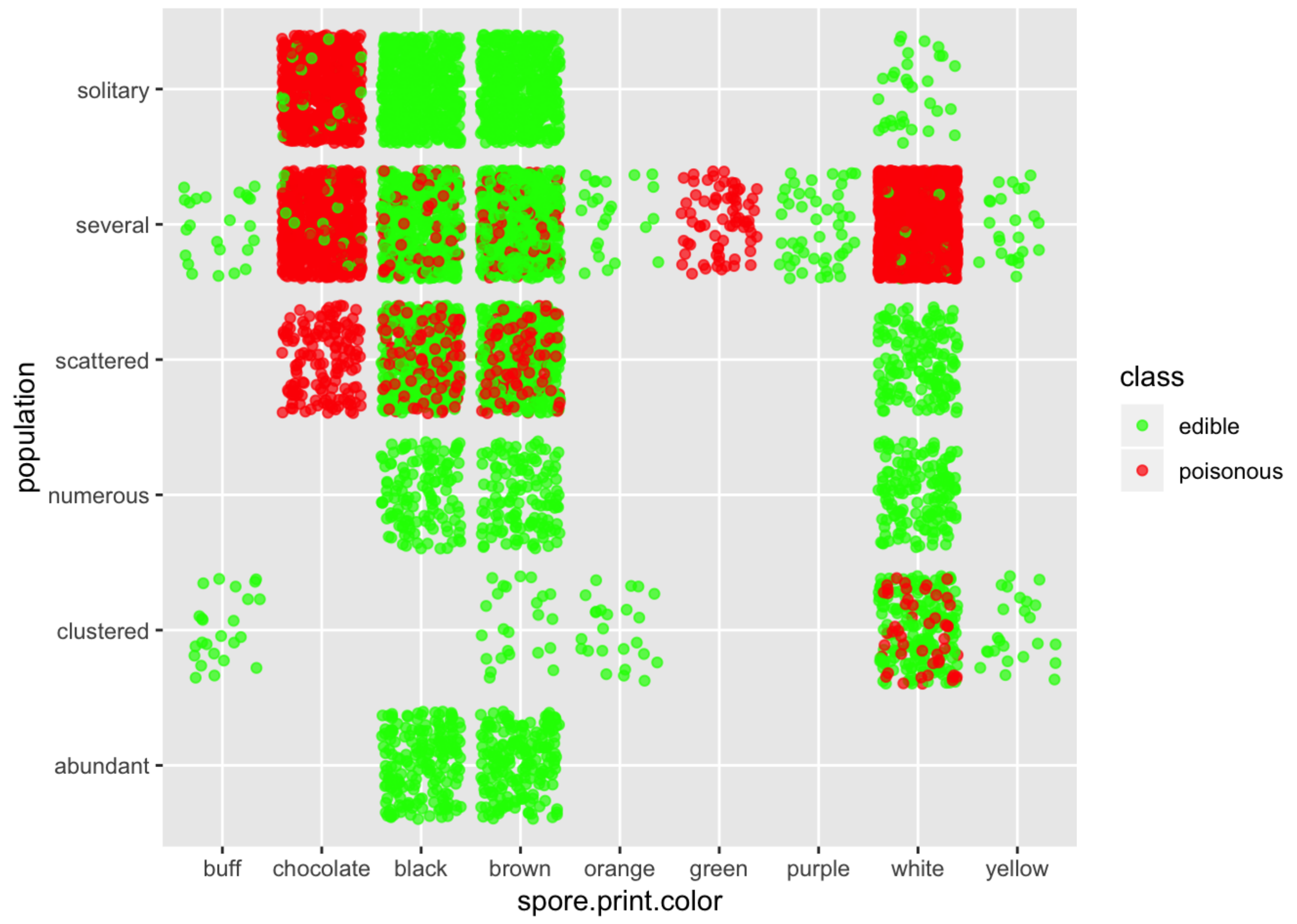


We can see many combinations which are defintelly to avoid (buff/green, brow) and a few that are safe (smooth + red/gray/pink)

Finally we take a look at:

Population and spore print color

```
ggplot(mushrooms, aes(x = spore.print.color , y = population, col = class)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red")) +
  geom_jitter(alpha = 0.7)
```

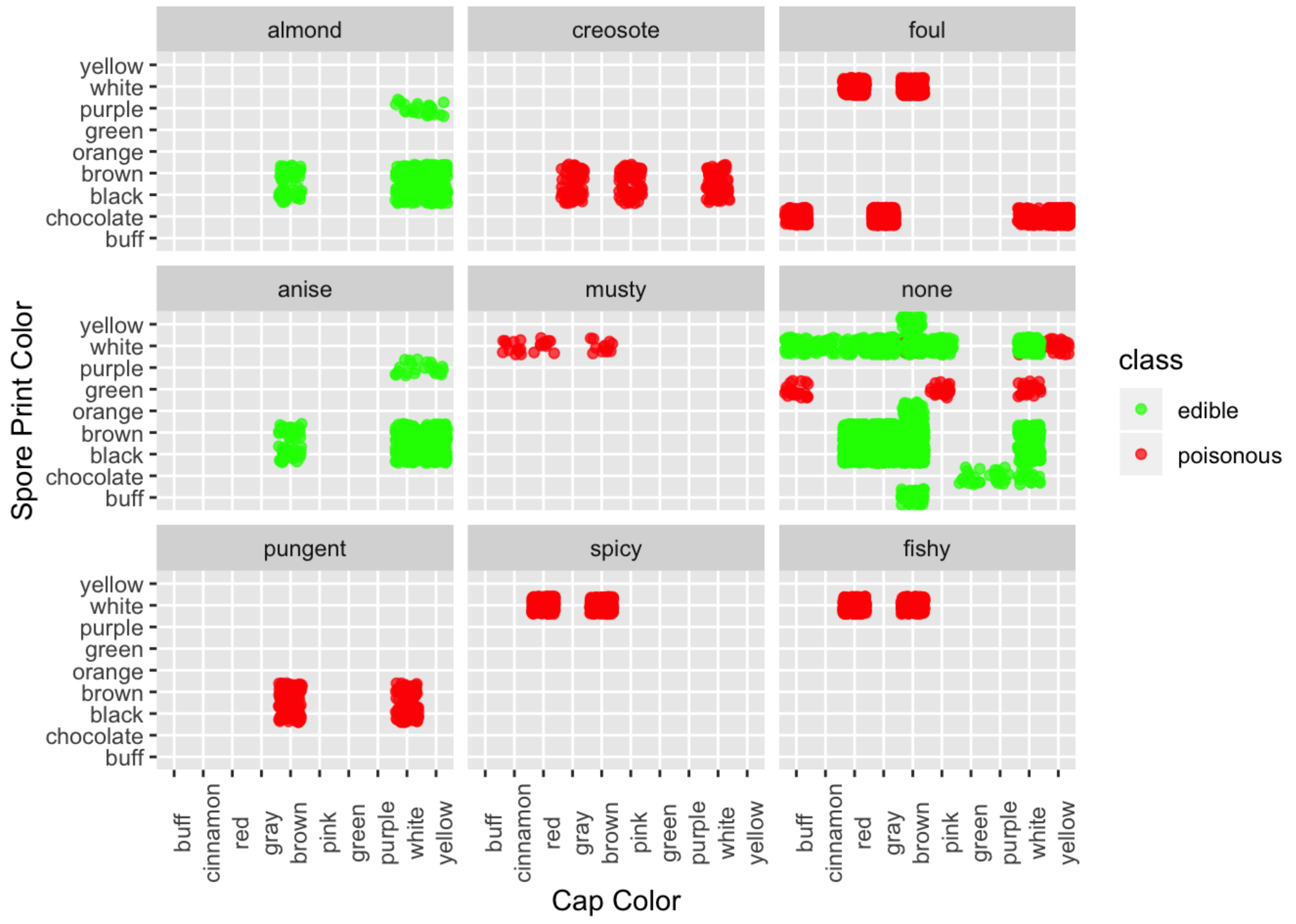


Again, we see several combinations which are safe, incl. any mushrooms with a numerous population or buff, orange , purple, or yellow spore prints.

To finalize our initial data exploration, let's look at three attributes that seem to play a big role in prediting edibility:

Cap Color, Sport Print Color and Odor

```
ggplot(mushrooms, aes(x = mushrooms$cap.color  , y = mushrooms$spore.print.color, col =
class)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red")) +
  geom_jitter(alpha = 0.7) + facet_wrap(mushrooms$odor) +
  xlab("Cap Color") +
  ylab("Spore Print Color") +
  theme(axis.text.x = element_text(angle = 90))
```



There are clearly pattern, so let’s try to find a model to predict if a mushroom is edible or not.

It looks like we might be able to make a good prediction using a regression model, but we already used that in the previous MovieLens exercise (https://github.com/prangberg/harvard-data-science (https://github.com/prangberg/harvard-data-science)) and want to use a ‘real’ machine learning model this time.

Modeling Approach

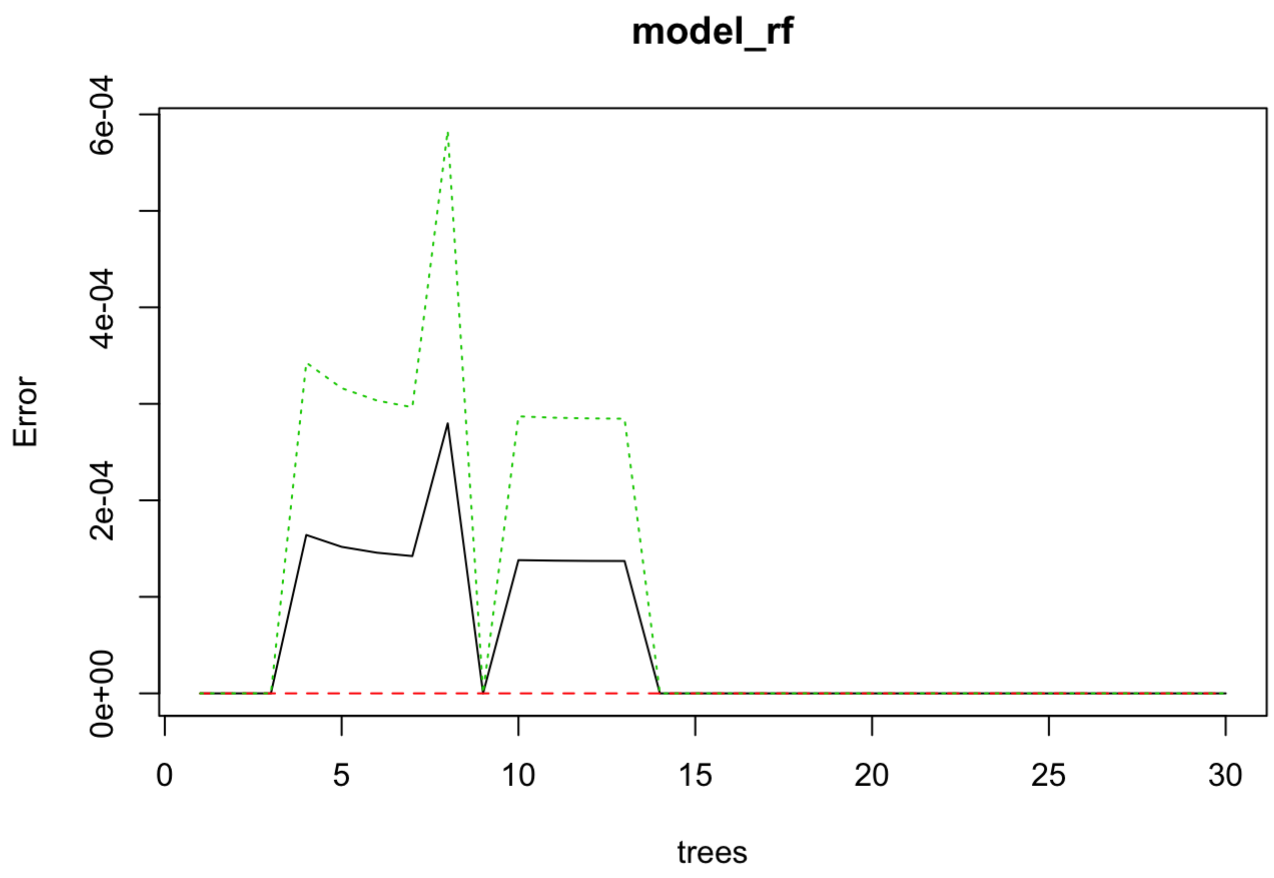
Split the data into training and a 10% validation set:

```
set.seed(1)
# Validation set will be 10%
test_index <-
  createDataPartition(
    y = mushrooms$class,
    times = 1,
    p = 0.1,
    list = FALSE
  )
edx <- mushrooms[-test_index, ]
validation <- mushrooms[test_index, ]
```

Random Forest

We'll use the Random Forest algorithm

```
set.seed(1)
model_rf <- randomForest(class ~ ., ntree = 30, data = edx)
plot(model_rf)
```



The plot shows that above 15 trees, the error isn't decreasing anymore and is very close to 0.

We use the model to create a prediction for our training data set.

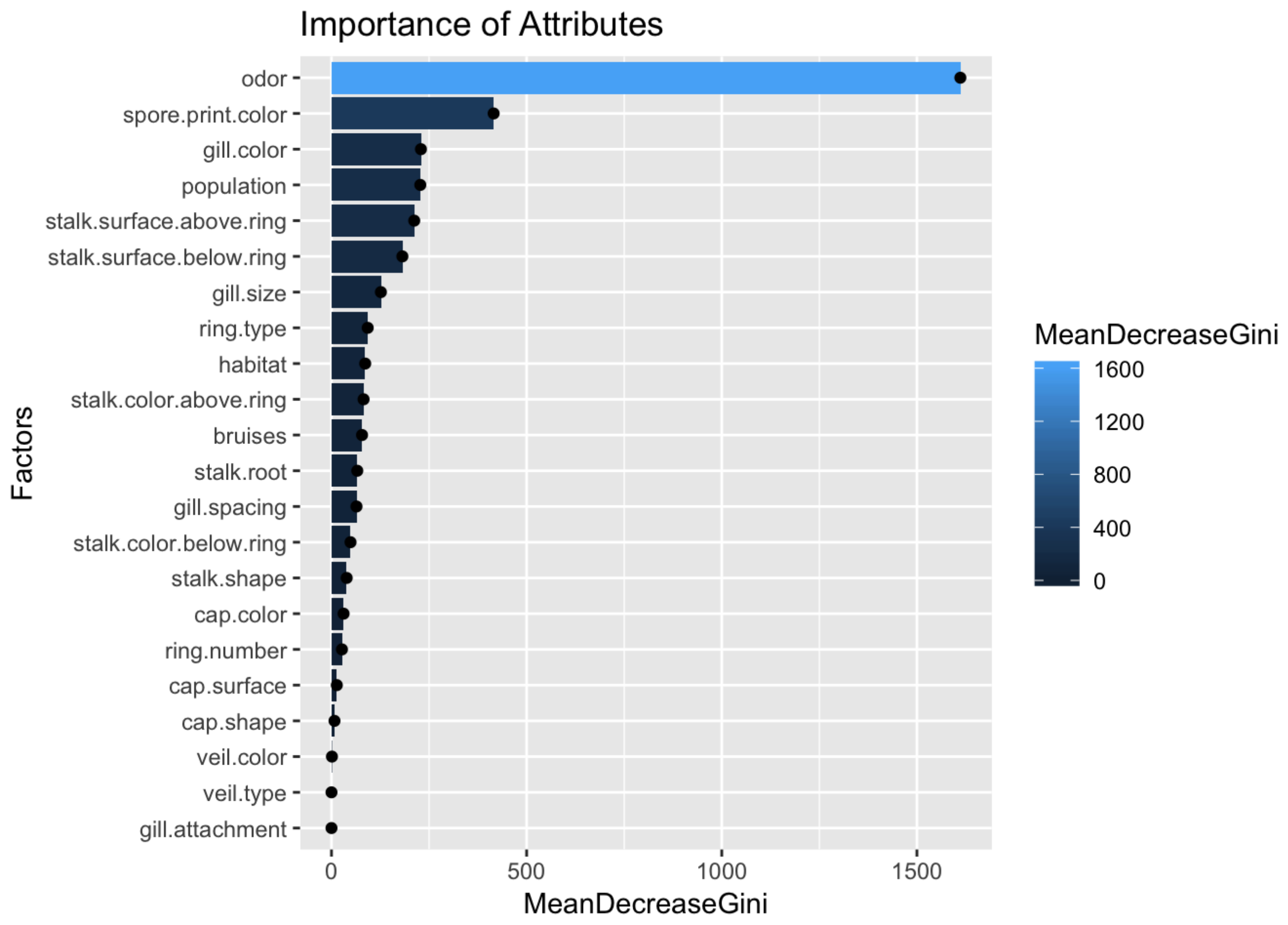
```
edx$predicted <- predict(model_rf ,edx)
confusionMatrix(data = edx$predicted, reference = edx$class ,
                positive = "edible")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  edible poisonous
##   edible      3787         0
##   poisonous      0      3524
##
##                Accuracy : 1
##                  95% CI : (0.9995, 1)
##     No Information Rate : 0.518
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.000
##             Specificity : 1.000
##          Pos Pred Value : 1.000
##          Neg Pred Value : 1.000
##              Prevalence : 0.518
##          Detection Rate : 0.518
##    Detection Prevalence : 0.518
##       Balanced Accuracy : 1.000
##
##        'Positive' Class : edible
##
```

Using this model, there are no errors for the training set: This is a perfect prediction of which mushrooms are edible or poisonous.

Let’s take a quick look at which of the attributes of a mushroom play the most important role in predicting its edibility:

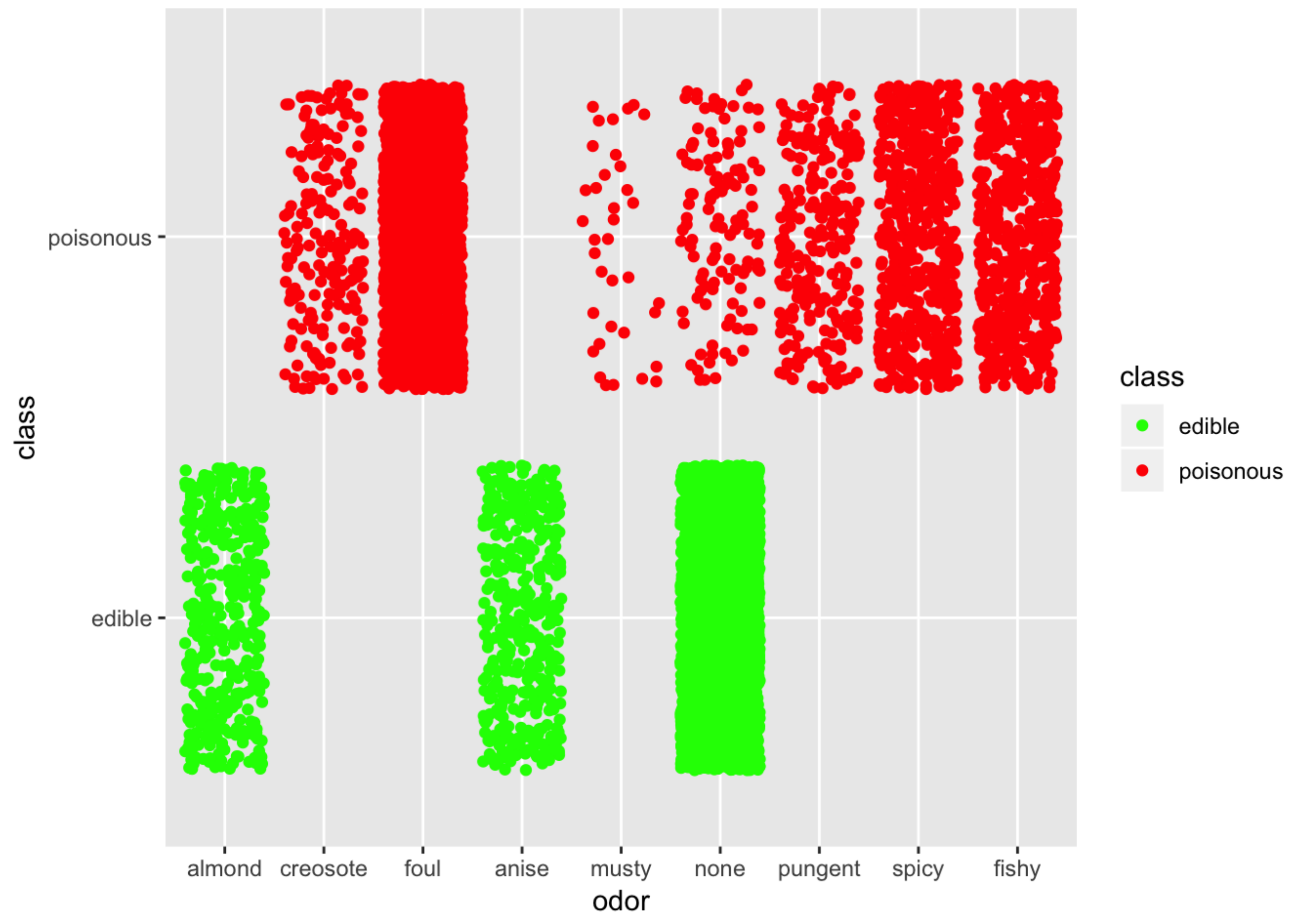
```
var_imp <-importance(model_rf) %>% data.frame() %>%
  rownames_to_column(var = "Variable") %>%
  arrange(desc(MeanDecreaseGini))

#Importance of attributes
ggplot(var_imp, aes(x=reorder(Variable, MeanDecreaseGini), y=MeanDecreaseGini, fill=Mean
DecreaseGini)) +
  geom_bar(stat = 'identity') +
  geom_point() +
  coord_flip() +
  xlab("Factors") +
  ggtitle("Importance of Attributes")
```



A higher descrease in Gini means that a particular predictor variable plays a greater role in partitioning the data into the defined classes. The plot indicates that **Odor** is the most predicive variable in determining edibility.

We can confirm the importance of odor when looking at this plot:

```
ggplot(mushrooms, aes(x = odor, y = class, col = class)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red"))+ geom_point(position='jitter')
```

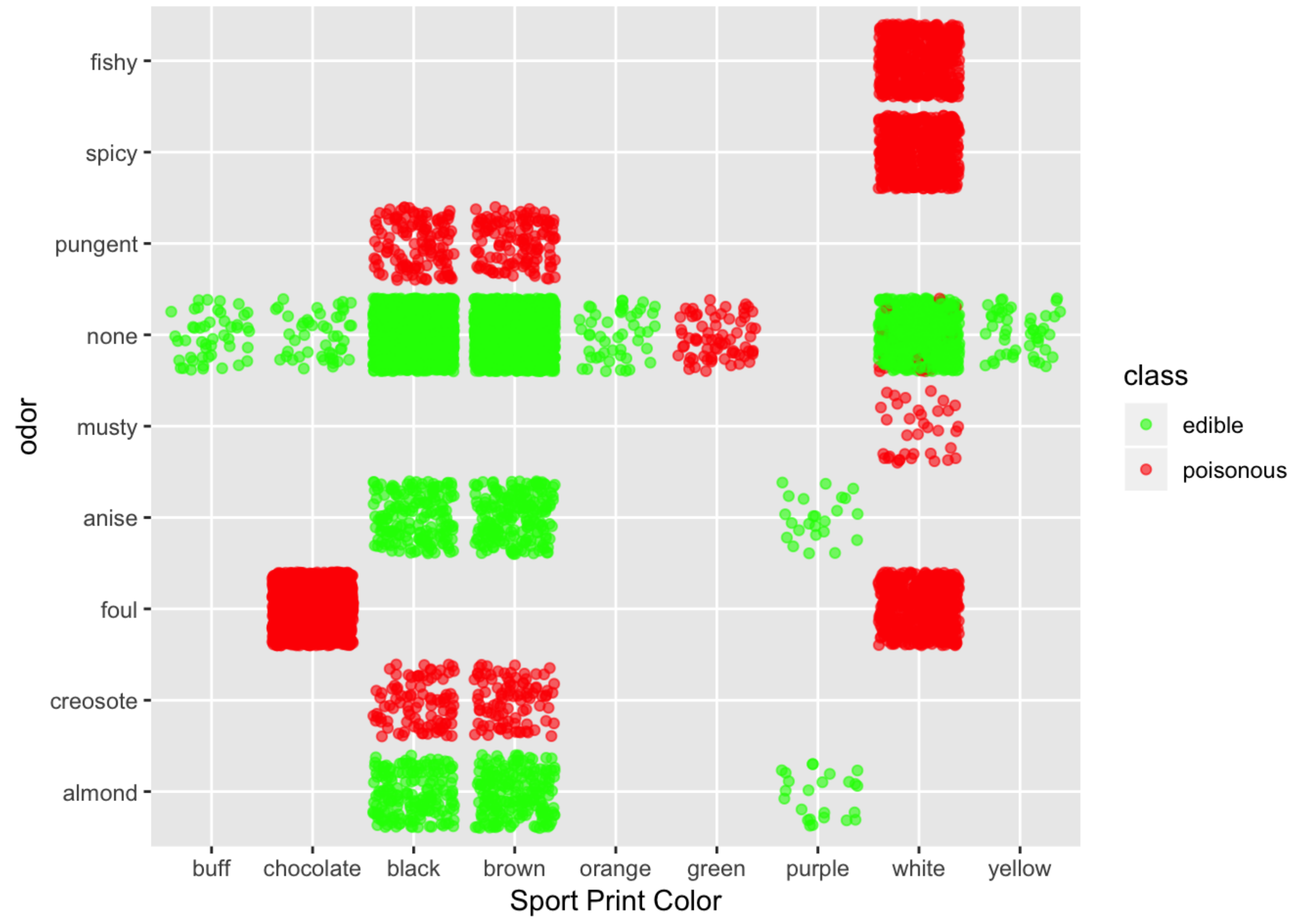


Actually any mushroom that does have an odor (i.e. not 'none') can be predicted: Almond or Anise are always safe. Creosote, foul, pungent, spicy or fishy are always poisonous. Only those mushrooms without odor cannot be predicted, even though it seems that most are edible.

-> So if there was only *ONE* single attribute you can consider when you find a mushroom, it should be odor.

Taking a quick look at the second most important factor *Sport Print Color*, in combination with odor:

```
ggplot(mushrooms, aes(x = spore.print.color, y = odor, col = class)) +
  scale_color_manual(breaks = c("edible", "poisonous"),
                     values = c("green", "red")) +
  geom_jitter(alpha = 0.6) +
  xlab("Sport Print Color")
```



This plot helps reduce the large uncertainty we had with odorless (odor=none) mushrooms. We can eat any odorless mushroom exceopt those with green or white spore print color. We could continue this exercise and would probably come up with a very precise regression model, but let's focus on our Random Tree prediction.

Validation

Let's apply the Random Forest model to our validation set and see how good the predictions are:

```
validation$predicted <- predict(model_rf ,validation)

confusionMatrix(data = validation$predicted, reference = validation$class , positive =
"edible")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  edible poisonous
##   edible       421         0
##   poisonous      0       392
##
##                Accuracy : 1
##                  95% CI : (0.9955, 1)
##     No Information Rate : 0.5178
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5178
##          Detection Rate : 0.5178
##    Detection Prevalence : 0.5178
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : edible
##
```

This is a perfect prediction - Accuracy, Sensitivity and Specificity are 1.00.

Results

The Model using Random Forests gives a 100% accurate prediction if a mushroom is edible or not - based on its attributes. Above 15 trees, the error isn’t decreasing anymore and is equal to 0.

Conclusion

Odor is the most predicive variable in determining edibility. Almond or Anise are always safe. Creosote, foul, pungent, spicy or fishy are always poisonous. So if there was only ONE single attribute you can consider when you find a mushroom, it should be odor.

Using the other attributes in a Random Forest model, we can predict with 100% accuracy, if a mushroom is edible or not.

It would be interesting to apply this model to a larger dataset which containts more than 23 species of gilled mushrooms in the Agaricus and Lepiota Family. This might require a higher number of Random Trees or a better fitted model.