HarvardX: PH125.9x Data Science Beer Recipe Project

Le Minh Thinh

May 17, 2019

Contents

1 Introduction

Beer seem to be one of the most favorite drink of all time. Recently, there are many different types of beer occurred in the Vietnam market, and we seem to be interested in challenging each other to correctly name those beers without seeing their labels. In fact, the project is conducted because of the curiosity about beer and the urge to find answer for the challenge.

The project arms to predict the 10 most popular types of beer using the "Brewer's Friend Beer Recipes" dataset in Kaggle. It is a dataset of 75,000 homemade brewed beers with over 176 different styles. Beer records were reported individually by each user, and those beers were classified according to one of the 176 different styles.

The report is expected to follow part of the structure which is recommended by Dr. Roger D.Peng in his book called "Report Writing for Data Science in R". Thus, those steps are.

1. Defining the question
2. Obtaining the data
3. Cleaning the data
4. Exploratory data analysis
5. Statistical prediction/modelling
6. Interpretation of results
7. Conclusion

2 Defining the question

There are two main questions the project expects to answer. The first one is what model will be the good one to produce the highest accuracy. The second one is what is the most important characteristic of beers that help us in choicing the right beer style.

3 Data Cleaning

3.1 Remove and free up working space

```r
# Remove and free up working space
rm(list = ls(all.names = TRUE))
gc()
```

```
##           used (Mb) gc trigger (Mb) max used (Mb)
## Ncells  520927 27.9    1183787 63.3   609151 32.6
## Vcells 1024494  7.9    8388608 64.0  1597850 12.2
```

3.2 Load necessary packages for exploration and wrangling purposes

```r
# To import, tidy, wrangle, visualize, model and communicate the data
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.r-project.org")

# To create summary statistics of variables
if(!require(skimr)) install.packages("skimr", repos = "http://cran.r-project.org")

# To create grid display for graphs
if(!require(gridExtra)) install.packages("gridExtra", repos = "http://cran.r-project.org")
```

3.3 Load the raw dataset downloaded on Kaggle website

```r
# Load the raw dataset
recipedata <- read_csv("dataset/recipeData.csv")
```

3.4 Explore the raw dataset

3.4.1 Check the names of all columns in the raw dataset

Those names seems to be easily read and written while we recall them in different steps of the project, except "Size(L)". In fact, "Size(L)" will be renamed in the next step.

```
# Names of variables in the raw dataset
names(recipedata)
```

```
##  [1] "BeerID"        "Name"          "URL"           "Style"
##  [5] "StyleID"       "Size(L)"       "OG"            "FG"
##  [9] "ABV"           "IBU"           "Color"         "BoilSize"
## [13] "BoilTime"      "BoilGravity"   "Efficiency"    "MashThickness"
## [17] "SugarScale"    "BrewMethod"    "PitchRate"     "PrimaryTemp"
## [21] "PrimingMethod" "PrimingAmount" "UserId"
```

3.4.2 Define the variables

- The "names" function had shown that the dataset had 23 variables, and their definitions are described below.

1. BeerID : Record ID of each user
2. Name: A beer name made by an user
3. URL: Location of recipe webpage
4. Style: Beer Style
5. StyleID: Numeric ID of a beer style
6. Size(L): the batch size of the listed recipe in liter (L)
7. OG: the original gravity of wort before fermentation in Degree Plato (P) or Specific Gravity (SG)
8. FG: the final gravity of wort after fermentation in Degree Plato (P) or Specific Gravity (SG)
9. ABV: Alcohol By Volume in percentage (%)
10. IBU: International Bittering Units (IBU)
11. Color: Standard Reference Method (SRM)
12. BoilSize: Fluid at beginning of boil in liter (L)
13. BoilTime: time to boil the wort in minutes (min)
14. BoilGravity: the gravity of wort before the boil in Degree Plato (P) or Specific Gravity (SG)
15. Efficiency: Beer mash extraction efficiency in percentage (%)
16. MashThickness: Amount of water per pound of grain in liters per kilogram (L/kg)
17. SugarScale: Scale to determine the concentration of dissolved solids in wort support in both Degree Plato (P) and Specific Gravity (SG)
18. BrewMethod: Various techniques for brewing
19. PitchRate: Yeast added to the fermentor per gravity unit (million cells/ml/P)
20. PrimaryTemp: Temperature at the fermenting stage (C)
21. PrimingMethod: to add other ingredients
22. PrimingAmount: Amount of priming ingredients used (g)
23. UserId: ID of an user

- The definition of those variables are supported by the owner of the dataset called "Brewer's Friend Beer Recipes", and other websites which are listed below.

1. Frequently Asked Questions in Brewer' Friend Beer Recipes
2. Box Brew Kits
3. Craft Beer and Brewing Magazine

3.4.3 Look for missing values in the raw dataset

Using "glimpse" function would provide general information of the dataset, especially what we expect about the types of variables and how missing values were represented in the dataset.

```
# Have a look at the raw dataset
glimpse(recipedata)
```

```
## Observations: 73,861
## Variables: 23
## $ BeerID        <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
## $ Name          <chr> "Vanilla Cream Ale", "Southern Tier Pumking clon...
## $ URL           <chr> "/homebrew/recipe/view/1633/vanilla-cream-ale", ...
## $ Style         <chr> "Cream Ale", "Holiday/Winter Special Spiced Beer...
## $ StyleID       <dbl> 45, 85, 7, 7, 20, 10, 86, 45, 129, 86, 7, 7, 7, ...
## $ `Size(L)`     <dbl> 21.77, 20.82, 18.93, 22.71, 50.00, 24.61, 22.71,...
## $ OG            <dbl> 1.055, 1.083, 1.063, 1.061, 1.060, 1.055, 1.072,...
## $ FG            <dbl> 1.013, 1.021, 1.018, 1.017, 1.010, 1.013, 1.018,...
## $ ABV           <dbl> 5.48, 8.16, 5.91, 5.80, 6.48, 5.58, 7.09, 5.36, ...
## $ IBU           <dbl> 17.65, 60.65, 59.25, 54.48, 17.84, 40.12, 268.71...
## $ Color         <dbl> 4.83, 15.64, 8.98, 8.50, 4.57, 8.00, 6.33, 5.94,...
## $ BoilSize      <dbl> 28.39, 24.61, 22.71, 26.50, 60.00, 29.34, 30.28,...
## $ BoilTime      <dbl> 75, 60, 60, 60, 90, 70, 90, 75, 75, 60, 90, 90, ...
## $ BoilGravity   <chr> "1.038", "1.07", "N/A", "N/A", "1.05", "1.047", ...
## $ Efficiency    <dbl> 70, 70, 70, 70, 72, 79, 75, 70, 73, 70, 74, 70, ...
## $ MashThickness <chr> "N/A", "N/A", "N/A", "N/A", "N/A", "N/A", "N/A",...
## $ SugarScale    <chr> "Specific Gravity", "Specific Gravity", "Specifi...
## $ BrewMethod    <chr> "All Grain", "All Grain", "extract", "All Grain"...
## $ PitchRate     <chr> "N/A", "N/A", "N/A", "N/A", "N/A", "1", "N/A", "...
## $ PrimaryTemp   <chr> "17.78", "N/A", "N/A", "N/A", "19", "N/A", "N/A"...
## $ PrimingMethod <chr> "corn sugar", "N/A", "N/A", "N/A", "Sukkerlake",...
## $ PrimingAmount <chr> "4.5 oz", "N/A", "N/A", "N/A", "6-7 g sukker/l",...
## $ UserId        <dbl> 116, 955, NA, NA, 18325, 5889, 1051, 116, 116, N...
```

3.4.4 Re-load the raw dataset with descriptions

- Add “N/A” to the description of missing values in the function “read_csv”
- Specify variable types

```
# Load dataset with an update in description
recipedata <- read_csv("dataset/recipeData.csv",
                       na = c("", "NA", "N/A"),
                       col_types = cols(Style = col_factor(levels = NULL),
                                        `Size(L)` = col_number(),
                                        OG = col_number(),
                                        FG = col_number(),
                                        ABV = col_number(),
                                        IBU = col_number(),
                                        Color = col_number(),
                                        BoilSize = col_number(),
                                        BoilTime = col_number(),
                                        BoilGravity = col_number(),
                                        Efficiency = col_number(),
                                        MashThickness = col_number(),
                                        SugarScale = col_factor(levels = NULL),
                                        BrewMethod = col_factor(levels = NULL),
                                        PitchRate = col_number(),
                                        PrimaryTemp = col_number(),
                                        PrimingMethod = col_factor(levels = NULL),
                                        PrimingAmount = col_factor(levels = NULL),
                                           UserId = col_number()))
```

- In addition, variable “`Size(L)`” was named in an odd way which might create confusion in the modelling steps, so it will be changed to “sizeL”.

```
# Rename Size(L) to sizeL
recipedata <- recipedata %>% rename(sizeL = `Size(L)`)
```

3.4.5 Summary statistics of the updated dataset

```
# Summarize the dataset
skim_with(numeric = list(hist = NULL))
skim(recipedata)
```

```
## Skim summary statistics
##  n obs: 73861
##  n variables: 23
##
## -- Variable type:character -------------------------------------------------------------------------
##  variable missing complete     n min max empty n_unique
##      Name       1    73860 73861   1  83     0    59140
##       URL       0    73861 73861  26 118     0    73861
##
## -- Variable type:factor ----------------------------------------------------------------------------
##       variable missing complete     n n_unique
##     BrewMethod       0    73861 73861        4
##  PrimingAmount   69085     4776 73861     1892
##  PrimingMethod   67095     6766 73861      871
##          Style     596    73265 73861      175
##     SugarScale       0    73861 73861        2
##                                    top_counts ordered
##  All: 49692, BIA: 12016, ext: 8626, Par: 3527   FALSE
##       NA: 69085, 5 o: 205, 3/4: 110, 4 o: 106   FALSE
##       NA: 67095, Cor: 717, Dex: 503, cor: 360   FALSE
##   Ame: 11940, Ame: 7581, Sai: 2617, Ame: 2277   FALSE
##                  Spe: 71959, Pla: 1902, NA: 0   FALSE
##
## -- Variable type:numeric ---------------------------------------------------------------------------
##       variable missing complete     n     mean       sd      p0      p25
##            ABV       0    73861 73861     6.14     1.88   0         5.08
##         BeerID       0    73861 73861 36931    21321.98   1     18466
##    BoilGravity    2990    70871 73861     1.35     1.93   0         1.04
##       BoilSize       0    73861 73861    49.72   193.25   1        20.82
##       BoilTime       0    73861 73861    65.07    15.02   0        60
##          Color       0    73861 73861    13.4     11.94   0         5.17
##     Efficiency       0    73861 73861    66.35    14.09   0        65
##             FG       0    73861 73861     1.08     0.43  -0.003     1.01
##            IBU       0    73861 73861    44.28    42.95   0        23.37
##  MashThickness   29864    43997 73861     2.13     1.68   0         1.5
##             OG       0    73861 73861     1.41     2.2    1         1.05
##      PitchRate   39252    34609 73861     0.75     0.39   0         0.35
##    PrimaryTemp   22662    51199 73861    19.18     4.22 -17.78      18
##          sizeL       0    73861 73861    43.93   180.37   1        18.93
##        StyleID       0    73861 73861    60.18    56.81   1        10
##         UserId   50490    23371 73861 43078.07 27734.25  49     20984
##       p50      p75      p100
##      5.79      6.83     54.72
##  36931    55396     73861
##      1.05      1.06     52.6
##     27.44    30       9700
##     60       60        240
##      8.44     16.79    186
##     70       75        100
##      1.01      1.02     23.42
##     35.77     56.38   3409.3
##      1.5       3        100
##      1.06      1.07     34.03
##      0.75      1          2
##     20       20        114
##     20.82    23.66    9200
##     35      111        176
##  42897    57841    134362
```

3.4.6 Standardizing the measurement unit

- The site Brewer's Friend allowed users to fill their beer's recipes in to different scale "Degree Plato" and "Specific Gravity", and it could be seen in "SugarScale" variable. The project will convert all values recored in "Degree Plato" to "Specific Gravity".
- Convert all Plato units to specific gravity (SG) The function could be found in the dataset owner website

SG = 1+ (plato / (258.6 – ((plato/258.2) x 227.1)))

```r
# Set a function to convert Plato to Specific Gravity
plato_to_sg <- function(x) {
  1 + (x / (258.6 - ((x/258.2) * 227.1)))
  }

# Nest the dataset by SugarScale
recipedata_nest_sugarscale <- recipedata %>%
  nest(-SugarScale)

# Apply the plato_to_sg function to the nested dataset
recipedata_nest_sugarscale$data[[2]] <- recipedata_nest_sugarscale$data[[2]] %>%
  mutate_at(vars(OG, FG, BoilGravity), plato_to_sg)

# Unnest the datasets
recipedata_sg_scale <- recipedata_nest_sugarscale %>%
  unnest(data) %>%
  mutate(SugarScale = as.factor("Specific_Gravity"))
```

- "SugarScale" indicated what measurement was used by users, so it should be dropped after all variables were calculated in "Specific Gravity".

```r
# Drop SugarScale
recipedata_sg_scale <- recipedata_sg_scale %>% select(- SugarScale)
```

3.4.7 Handling missing values

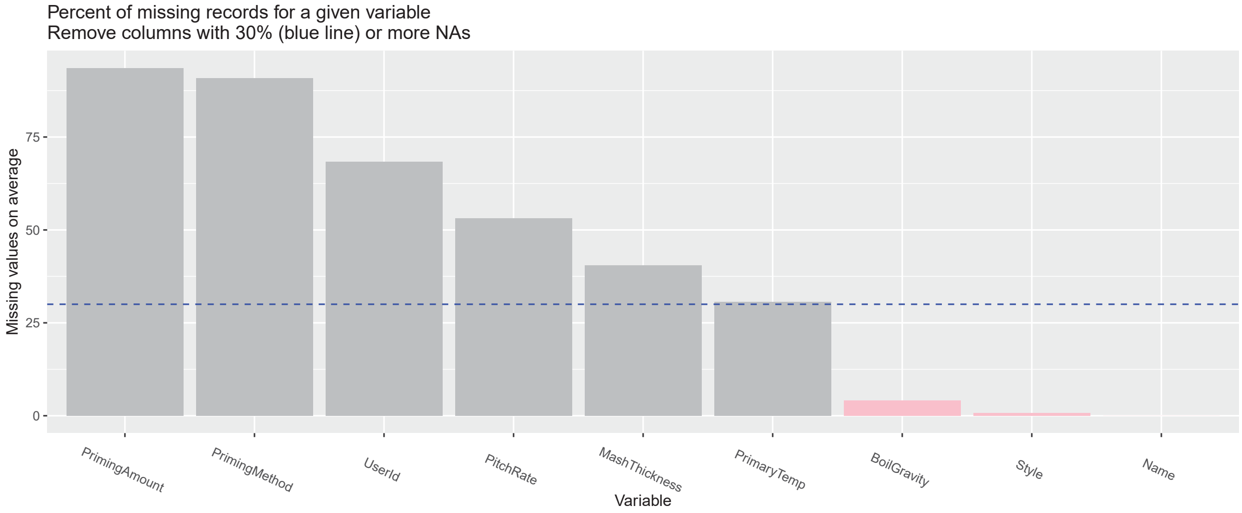
3.4.7.1 Visualize missing values of variables

There are six varibles that have more than thirty percent of missing values, such as “PrimingAmount”, “PrimingMethod”, “UserId”, “PitchRate”, “MashThickness” and “PrimaryTemp.” Those variables that got the missing rate above thirty percent will be dropped out, and the others will be left out for further consideration. In fact, there is special treatment for “Style”, “BoilGravity” and “Name” in the following section.

```
# Create a function to calculate proportion of missing values
mean_missingvalue_func <- function(x) {
  mean(is.na(x)) * 100
}

# Show variables with NA values
mean_na <- recipedata_sg_scale %>%
  summarise_all(mean_missingvalue_func) %>%
  gather("Variable", "NA_average") %>%
  filter(NA_average > 0) # Return columns with missing values

# Plot a barchart for the proportion of missing records in each variable
mean_na %>%
  mutate(na_vars = if_else(NA_average > 30, "gray", "pink")) %>%
  ggplot(aes(x = reorder(Variable, -NA_average), y = NA_average, fill = I(na_vars))) +
  geom_bar(stat = "identity") +
  geom_hline(yintercept = 30, color = "blue", linetype = 2) +
  theme(axis.text.x = element_text(angle = -25)) +
  labs(title =
         "Percent of missing records for a given variable\nRemove columns with 30% (blue line) or more
       x = "Variable",
       y = "Missing values on average")
```



3.4.7.2 Drop those variables that had more than 30 percent of missing values

```
# Drop incomplete variables
complete_var_recipedata_sg_scale <- mean_na$Variable[which(mean_na$NA_average > 30)]

recipedata_sg_scale_dropNAabove30 <- recipedata_sg_scale %>%
  select(-c(which(colnames(.) %in% complete_var_recipedata_sg_scale)))
```

3.4.7.3 Missing values in dependent variable “Style”

The variable “Style” had 596 missing values, while the “StyleID” did not have any missing. It might be caused by the typo of users when they added the beer recipe, so the project will fix this problem by matching the StyleID with missing value in Style.

Firstly, having a glimpse in the missing one will help us to choose an appropriate solution to handle this problem. In fact, the result shows that there is only one Style ID (111) represented for all 595 missings, and they can be replaced by the correct style using “styleData.csv”. StyleID (111) standed for “N/A”, so the project will drop those rows.

```
# Matching missing Style with Style ID
recipedata_sg_scale_dropNAabove30 %>% filter(is.na(Style)) %>% count(StyleID)
```

```
## # A tibble: 1 x 2
##   StyleID     n
##     <dbl> <int>
## 1     111   596
```

The missing values in Style will be dropped out.

```
# Drop rows with missing values in Style
recipedata_sg_scale_dropNAabove30_and_dropNAstyle <-
  recipedata_sg_scale_dropNAabove30 %>%
  filter(Style != is.na(Style))

# Double check the variable Style
skim(recipedata_sg_scale_dropNAabove30_and_dropNAstyle, Style)
```

```
## Skim summary statistics
##  n obs: 73265
##  n variables: 16
##
## -- Variable type:factor ------------------------------------------------------------------------
##  variable missing complete     n n_unique
##     Style       0    73265 73265      175
##                                  top_counts ordered
##  Ame: 11940, Ame: 7581, Sai: 2617, Ame: 2277   FALSE
```

3.4.7.4 Missing values in “BoilGravity”

Missing values in a numerial variable are commonly replaced by sample mean or sample median. It is believed that the distribution of “BoilGravity” will help to make a choice between those two numbers.

- Plot the histogram of Boil Gravity in Specific Gravity scale

```
# Histogram and boxplot of Boil Gravity
hist_boigravity <- recipedata_sg_scale_dropNAabove30_and_dropNAstyle %>%
  ggplot(aes(BoilGravity)) +
  geom_histogram(binwidth = 0.004, fill = "blue") +
  labs(title = "Histogram of BoilGravity in Specific Gravity scale",
       x = "BoilGravity",
       y = "Count")

boxplot_boilgravity <- recipedata_sg_scale_dropNAabove30_and_dropNAstyle %>%
  ggplot(aes(x = 1, y = BoilGravity)) +
  geom_boxplot(alpha = 0.1) +
  labs(title = "Boxplot of BoilGravity in Specific Gravity scale",
       x = "",
       y = "BoilGravity") +
```