Desafio Oncase - Receitas

Wagner J.F. Silva

18 dezembro 2019

Abstract

Estatístico, mestre e doutorando em Ciência da Computação apaixonado por ciência de dados e entusiasta no uso e criação de modelos/algoritmos whitebox para extração e mineração de conhecimento a partir de dados diversos. Debruçado, até o final de sete dias, sobre o estudo da base de dados de receitas com objetivo de entender suas peculiaridades e pepitas escondidas como forma de expor parte do conhecimento obtido na academia para a Oncase.

O problema

Este notebook foi construído com objetivo de solucionar questões de negócios como método de seleção da Oncase. Em linhas gerais, o problema consiste em extrair conhecimento a partir de dados de receitas de comidas. Devido a quantidade imensurável de conhecimento disponível através deste conjunto de dados a Oncase propôs seis perguntas a serem respondidas a partir dos dados, sendo a última de caráter opcional, são elas:

- A categorias pertencem as comidas mais calóricas?
- Quais os top 10 ingredientes contidos nas receitas mais calóricas?
- Se você tivesse que recomendar 3 receitas baseando-se nos dados, quais seriam?
- Alguma característica presente nos dados determina a alta nota de uma receita?
- Considerando-se as categorias das top 100 receitas em avaliação, quantas receitas há atualmente no site Epicurious para cada categoria?
- [opcional] Construa um classificador para recomendar tags (categorias) para as receitas.

Adiantamos de antemão que todas questões exceto a sexta foram respondidas.

Preparação das ferramentas e dos dados

Inicialmente vamos instalar todos os pacotes necessários para as análises, mudar o diretório de trabalho e tratar os dados através de limpeza e criação de features. Uma vez que os pacotes foram instalados estes devem ser disponibilizados para o uso.

Instalação dos pacotes necessários

Leitura dos dados

A leitura dos dados é feita pelo pacote jsonlite e transformado num `data.frame`. Dessa forma, cada coluna descreve uma variável. Consta na base de dados 20.130 receitas e 12 variáveis em formatos diversos: texto, contínuas, datas.

```
setwd("G:/Dropbox_Novo/Dropbox/selecao_oncase/selecao_oncase")
recipes <- fromJSON('receitas.json')
```

Análise descritiva preliminar

Para as variáveis contínuas nós calculamos as principais medidas resumo. No entando, é possível observar que há valores extremamente fora do esperado para cada variável, estes valores indicam algum erro de medição visto que a distância do terceiro quartil é muito grande, o que não configura um outlier, mas um erro claro de medição. Para identificá-los, nós criamos uma variabel `id`. A partir dela conseguimos identificar, inicialmente, dois indivíduos com valores altamente discrepante.

```
recipes %>% select(rating, fat, calories, protein, sodium) %>% summary()
```

```
##      rating           fat               calories
##  Min.   :0.000   Min.   :      0.0   Min.   :       0
##  1st Qu.:3.750   1st Qu.:      7.0   1st Qu.:     198
##  Median :4.375   Median :     17.0   Median :     331
##  Mean   :3.713   Mean   :    346.1   Mean   :    6308
##  3rd Qu.:4.375   3rd Qu.:     33.0   3rd Qu.:     586
##  Max.   :5.000   Max.   :1722763.0   Max.   :30111218
##  NA's   :30      NA's   :4222        NA's   :4154
##     protein              sodium
##  Min.   :     0.00   Min.   :       0
##  1st Qu.:     3.00   1st Qu.:      80
##  Median :     8.00   Median :     294
##  Mean   :    99.95   Mean   :    6211
##  3rd Qu.:    27.00   3rd Qu.:     711
##  Max.   :236489.00   Max.   :27675110
##  NA's   :4201        NA's   :4156
```

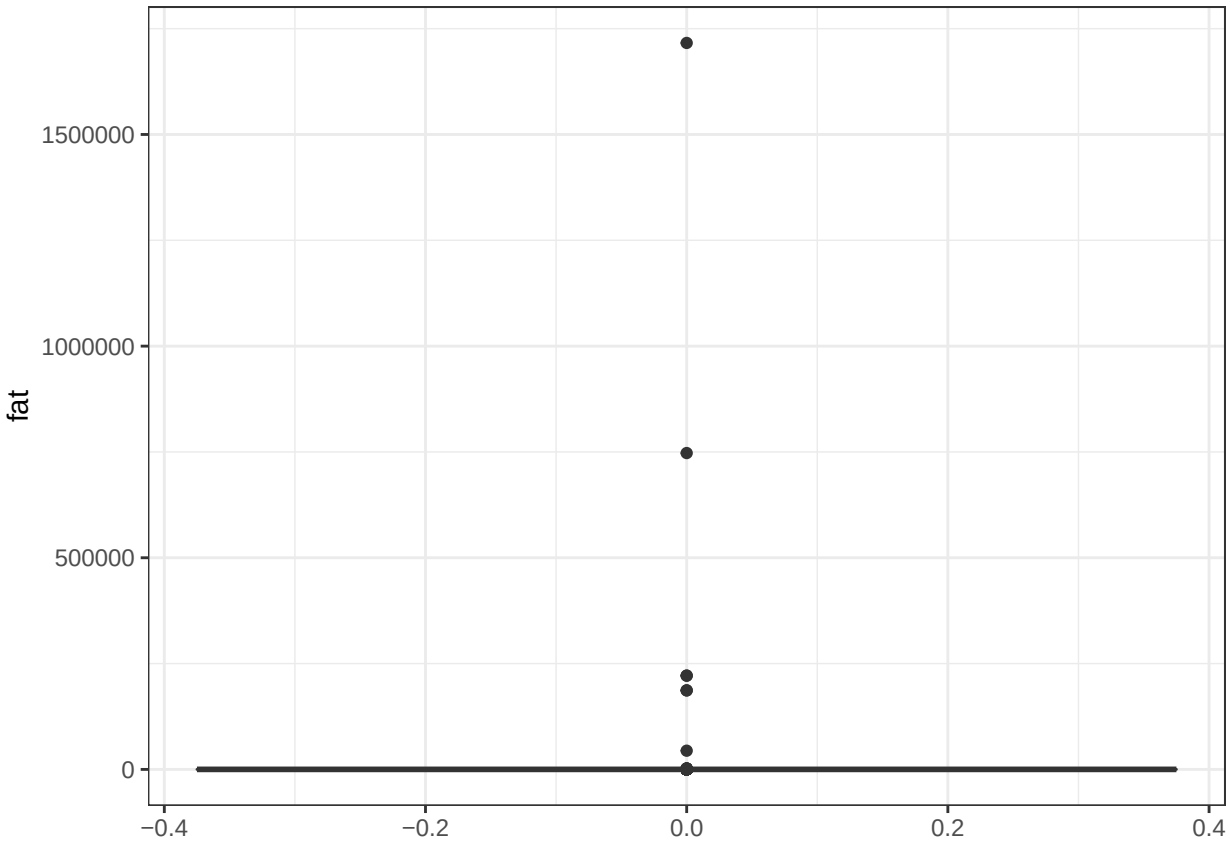
```
recipes <- recipes %>% mutate(id = seq_along(fat))
id_error <- recipes %>% select(fat, calories, protein, sodium) %>% apply(2, which.max)
id_error
```

```
##      fat calories  protein   sodium
##    11445    11445     1309    11445
```

Além destes, podemos observar outros valores estranhos nas variáveis contínuas, a saber: `fat, calories, protein` e `sodium`. Estes valores tem deformado a distribuição de probabilidade conforme podemos observar nos boxplots abaixo. Isto impede a identificação do uso adequado de modelos para explicação de comportamentos, caso necessário.

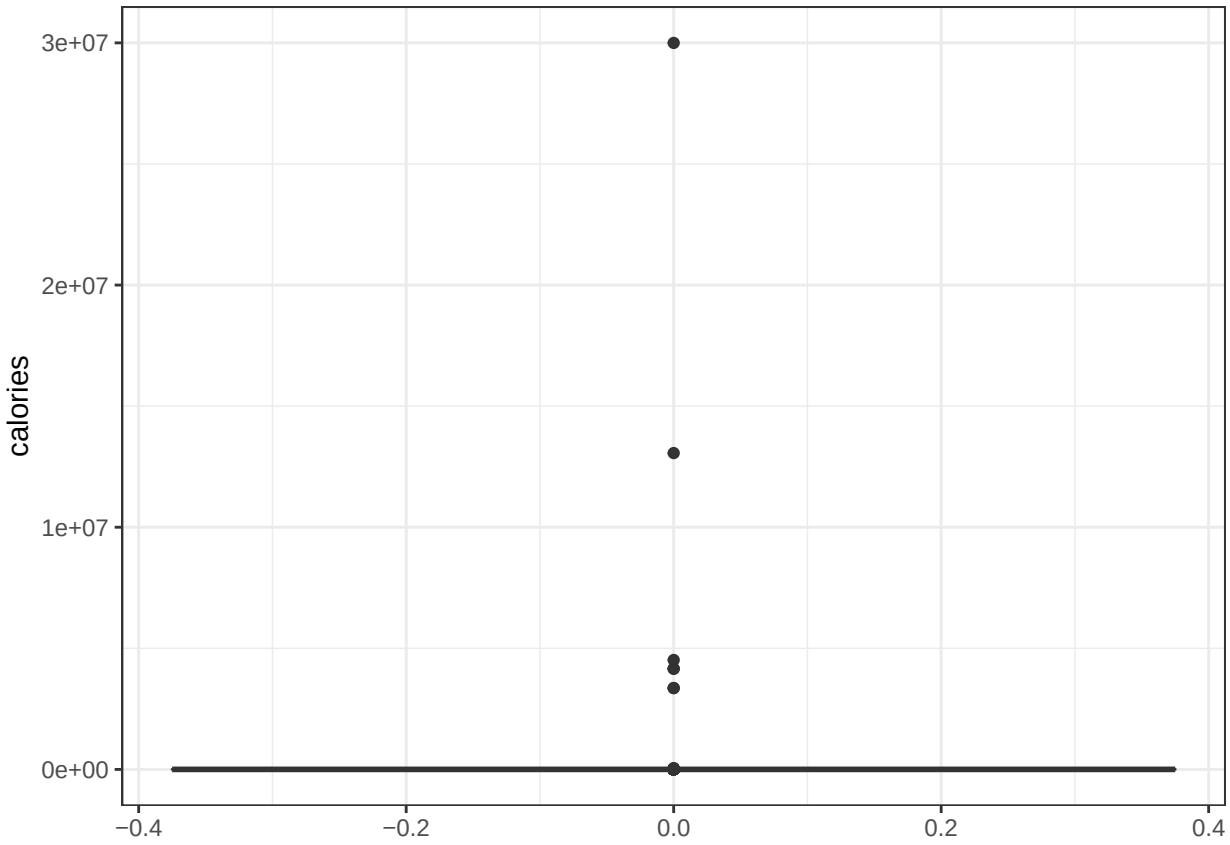
```
recipes[-id_error[1], ] %>% ggplot(aes(y = fat)) + geom_boxplot() + theme_bw()
```

```
## Warning: Removed 4222 rows containing non-finite values (stat_boxplot).
```



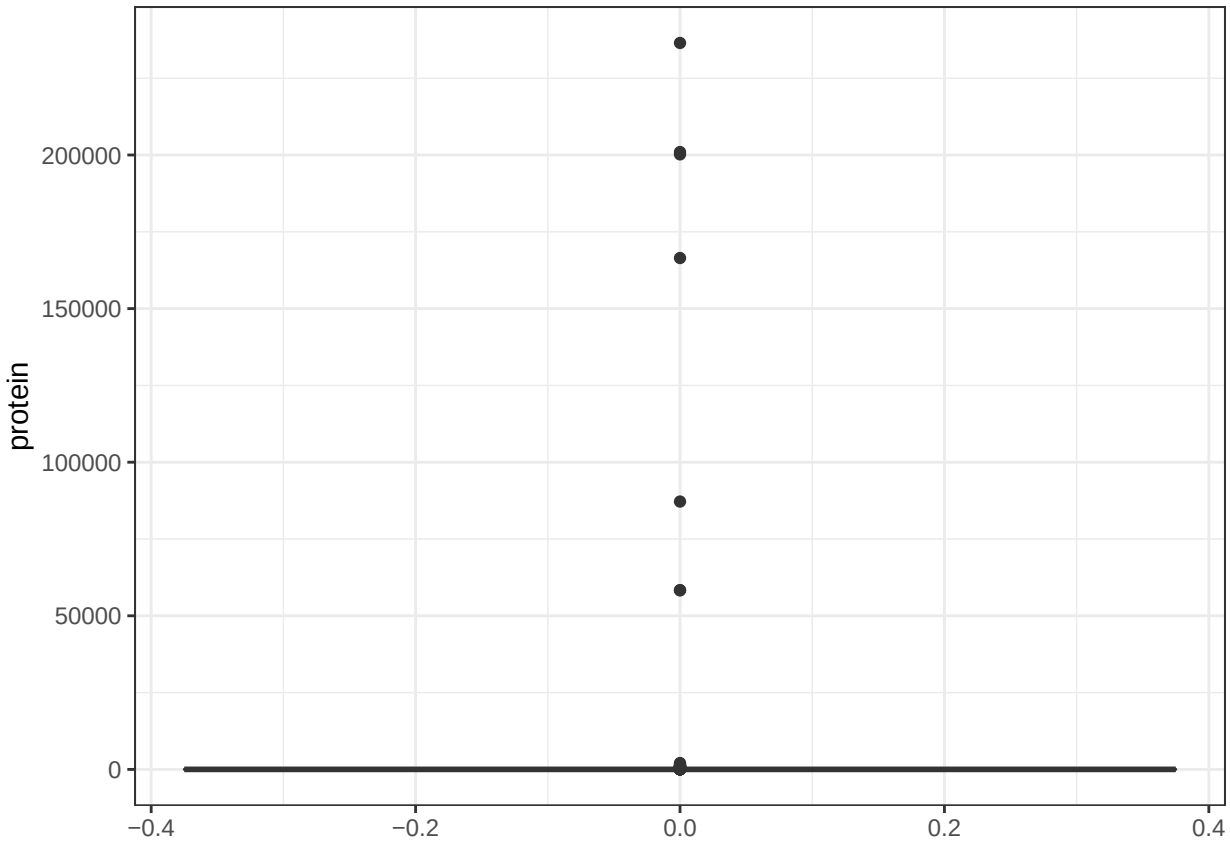

```
recipes[-id_error[2], ] %>% ggplot(aes(y = calories)) + geom_boxplot() + theme_bw()
```

```
## Warning: Removed 4154 rows containing non-finite values (stat_boxplot).
```



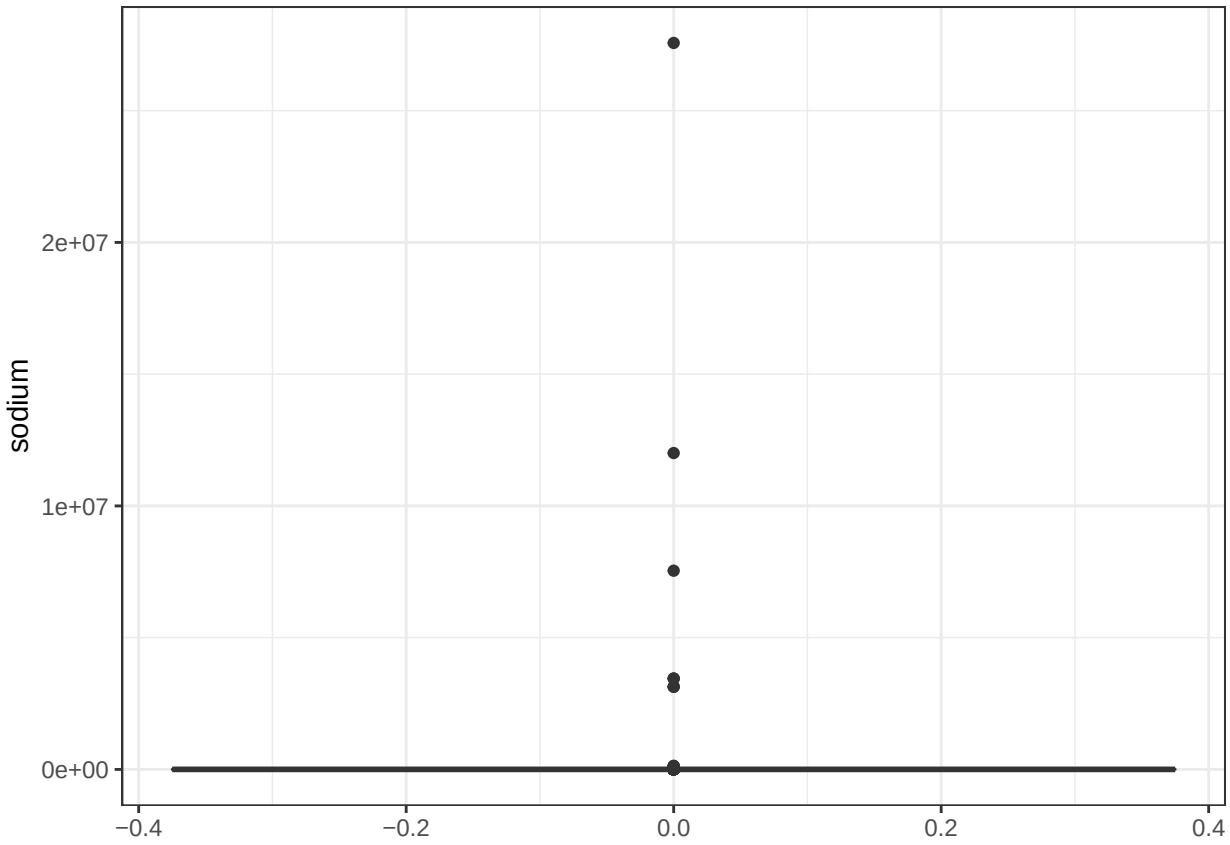

```
recipes[-id_error[3], ] %>% ggplot(aes(y = protein)) + geom_boxplot() + theme_bw()
```

```
## Warning: Removed 4201 rows containing non-finite values (stat_boxplot).
```




```
recipes[-id_error[4], ] %>% ggplot(aes(y = sodium)) + geom_boxplot() + theme_bw()
```

```
## Warning: Removed 4156 rows containing non-finite values (stat_boxplot).
```



Imputação de dados

Então, nós utilizamos fontes nutricionais externas para detectar estes valores discrepantes para cada variável descrita. Todas medidas consideram a média ingerida por uma pessoa por dia. Obviamente esta medida pode ser subestimada, pois dependende diretamente do tamanho da porção que serve cada receita e sobrestimada considerando que as métricas são diárias e não por refeição/receita. Dessa forma, supomos que há um equilíbrio nas métricas externas utilizadas. Em detalhes, as métricas são:

- Total de gordura (`fat`) - Segundo a Health a média de gordura para uma dieta diária de 2.800 calorias é de 93g. Baseado nessa informação trataremos 186g de gordura como limiar.
- Calorias (`calories`) - Ainda de acordo com a Health a média de calorias para um homem e uma mulher de um jovem é 2.800 e 2200 calorias, respectivamente. Devido a ausência do gênero como variável na base, utilizaremos como limiar a média entre homens e mulheres que é 2500 calorias.
- Proteína (`protein`) - Em entrevista para o NY Times a autora do livro "Devoured: How What We Eat Defines Who We Are", Sophie Egan afirma que em média um norte americano consome em média 100g de proteínas pode dia logo, o limiar adotado é 200g.
- Total de Sódio (`sodium`) - A American Heart Association conhecida como Heart informa que os americanos consomem em média 3.400mg de sódio por dia assim, este foi escolhido como valor limitante para definição de erro de medida ou não.

Em outras palavras, receitas que ultrapassem os limiares citados acima, nas suas respectivas variáveis, serão substituídas pela média truncada em 5%. O número de receitas que ultrapassam estes limiares são 678, 234, 441 e 328 para `fat`, `calories`, `protein` e `sodium`, respectivamente.

```r
c(sum(na.omit(recipes$fat) >= 93), sum(na.omit(recipes$calories) >= 2500),
  sum(na.omit(recipes$protein) >= 100), sum(na.omit(recipes$sodium) >= 3400))
```

```
## [1] 678 234 441 328
```

```r
id_error_fat <- which(recipes$fat >= 93)
id_error_calories <- which(recipes$calories >= 2500)
id_error_protein <- which(recipes$protein >= 100)
id_error_sodium <- which(recipes$sodium >= 3400)
```

```r
fat_trim_mean <- mean(recipes$fat, na.rm = TRUE, trim = .05)
calories_trim_mean <- mean(recipes$calories, na.rm = TRUE, trim = .05)
protein_trim_mean <- mean(recipes$protein, na.rm = TRUE, trim = .05)
sodium_trim_mean <- mean(recipes$sodium, na.rm = TRUE, trim = .05)

fat <- ifelse(is.na(recipes$fat), fat_trim_mean, recipes$fat)
calories <- ifelse(is.na(recipes$calories), fat_trim_mean, recipes$calories)
protein <- ifelse(is.na(recipes$protein), fat_trim_mean, recipes$protein)
sodium <- ifelse(is.na(recipes$sodium), fat_trim_mean, recipes$sodium)

fat[id_error_fat] <- fat_trim_mean
calories[id_error_calories] <- calories_trim_mean
protein[id_error_protein] <- protein_trim_mean
sodium[id_error_sodium] <- sodium_trim_mean
```

Criação de novas features

Acreditamos que a nota da receita está relacionada a dificuldade da mesma, pois receitas muito difíceis podem não ser concluídas com sucesso e com isso ter sua nota diminuída e vice versa. Então, elencamos algumas medidas para avaliar a dificuldade da receita, são elas: (1) número de processos (`num_process`); (2) cozida, assada, ou vai ao forno (`baked`); (3) número de caracteres (concisão) (`conciseness`); (4) preparation time (`time`).

```r
num_process <- sapply(recipes[[1]], function(x) length(x))
baked <- str_detect(recipes$directions, '°F|°C|°K')
conciseness <- str_length(recipes$directions)

hour <- sapply(str_extract_all(recipes$directions, "\\d+(?=\\shour)"),
               function(x) sum(as.numeric(x)))
minutes <- sapply(str_extract_all(recipes$directions, "\\d+(?=\\sminute)"),
               function(x) sum(as.numeric(x)))
time <- 24 * hour + minutes
```

Número de dias desde a publicação da receita pode ser um fator decisivo na nota da receita. Além disso, criamos uma variável que determina se uma "receita é clássica" ou não, elas são consideradas clássicas se tem mais de 10 anos (3650 dias) desde a publicação. Através de uma variável dummy `old_recipe` sabemos se as receitas são clássicas ou não.

```r
date <- recipes %>% mutate(date, date1 = as.Date(date)) %>% select(date1)
num_days <- as.numeric(Sys.Date() - date$date1)
num_days[which(is.na(num_days))] <- mean(num_days, na.rm = TRUE)
old_recipe <- (num_days >= 3650) %>% as.integer() %>% as.factor()
```

Outra importante variável para a nota da receita são as categorias que ela está inserida. Nós calculamos e armazenamos o número de categorias em que uma receita está inserida (`num_categories`) e verificamos

também se está é vegetariana ou não (`vegetarian`). Além disso, verificamos quais as categorias são mais frequentes nas receitas através da variável `freq_cat`

```
num_categories <- sapply(recipes$categories,
                         function(x) length(x))

vegetarian <- recipes %>%
  select(categories) %>%
  sapply(function(x) str_detect(x, pattern = 'Vegetarian')) %>%
  as.integer() %>% as.factor()

cat <- sapply(recipes$categories,
              function(x) stringi::stri_trans_general(x, "Latin-ASCII")  )

freq_cat <- factor(unlist(recipes$categories),
                         levels = unique(unlist(recipes$categories)))
summary(freq_cat) %>% head(10)
```

```
##       Bon Appétit       Peanut Free          Soy Free     Tree Nut Free
##              9355              8390              8088              7044
##        Vegetarian           Gourmet            Kosher       Pescatarian
##              6846              6648              6175              6042
##      Quick & Easy Wheat/Gluten-Free
##              5372              4906
```

A descrição da receita funciona como um subtítulo da receita e pode ser um fator de engajamento a executar a receita. Dessa forma, pode ter influência sobre a avaliação da receita. Portanto, três variáveis foram criadas, a primeira verifica se há descrição na receita, a segunda indica o número de orações de sentido completo na descrição e a terceira revela o número de caracteres na descrição. Elas são descritas pelas variáveis `desc`, `phrases` e `desc_len`, respectivamente.

```
desc <- recipes %>% select(desc) %>% is.na() %>% as.numeric()

phrases <- recipes %>%
  select(desc) %>%
  apply(1,
        function(x) str_count(x, pattern = '\\.|\\:|\\?|\\!|\\;'))
phrases <- ifelse(is.na(phrases), 0, phrases)

desc_len <- recipes %>% select(desc) %>% apply(1,
                                               function(x) str_length(x))
desc_len <- ifelse(is.na(desc_len), 0, desc_len)
```

Estudando a variável ingredientes podemos detectar que algumas receitas não apresentam nenhum ingrediente assim, não foram consideradas receitas válidas e foram deletadas do banco de dados. Para execução do processo de deleção utilizamos a variável `rwi` que significa "recipes wihtout ingredients". A base de dados `recipes2` armazena a nova base de dados sem as receitas inválidas. Em seguida, contamos o número de ingredientes por receita.

```
rwi <- which(sapply(recipes$ingredients, length) == 0)
rwi
```

```
##  [1]   774  1077  1080  1136  1454  1641  1908  2067  2692  2726  3331
## [12]  4507  4508  5147  5163  5425  5559  5817  6330  6700  6879  7026
## [23]  7585  7608  7769  7803  7882  8178  8335  9591  9663  9924 10086
## [34] 10207 10239 10601 10946 11176 11225 11632 11699 12056 12698 12790
## [45] 13152 13207 13243 13245 13303 13348 13382 13468 13555 13711 13945
```

```
## [56] 14260 14685 15325 15977 16046 16211 16331 16836 16904 16947 17331
## [67] 17619 18109 18262 18678 18744 18768 19128 19525 19548 19643
```

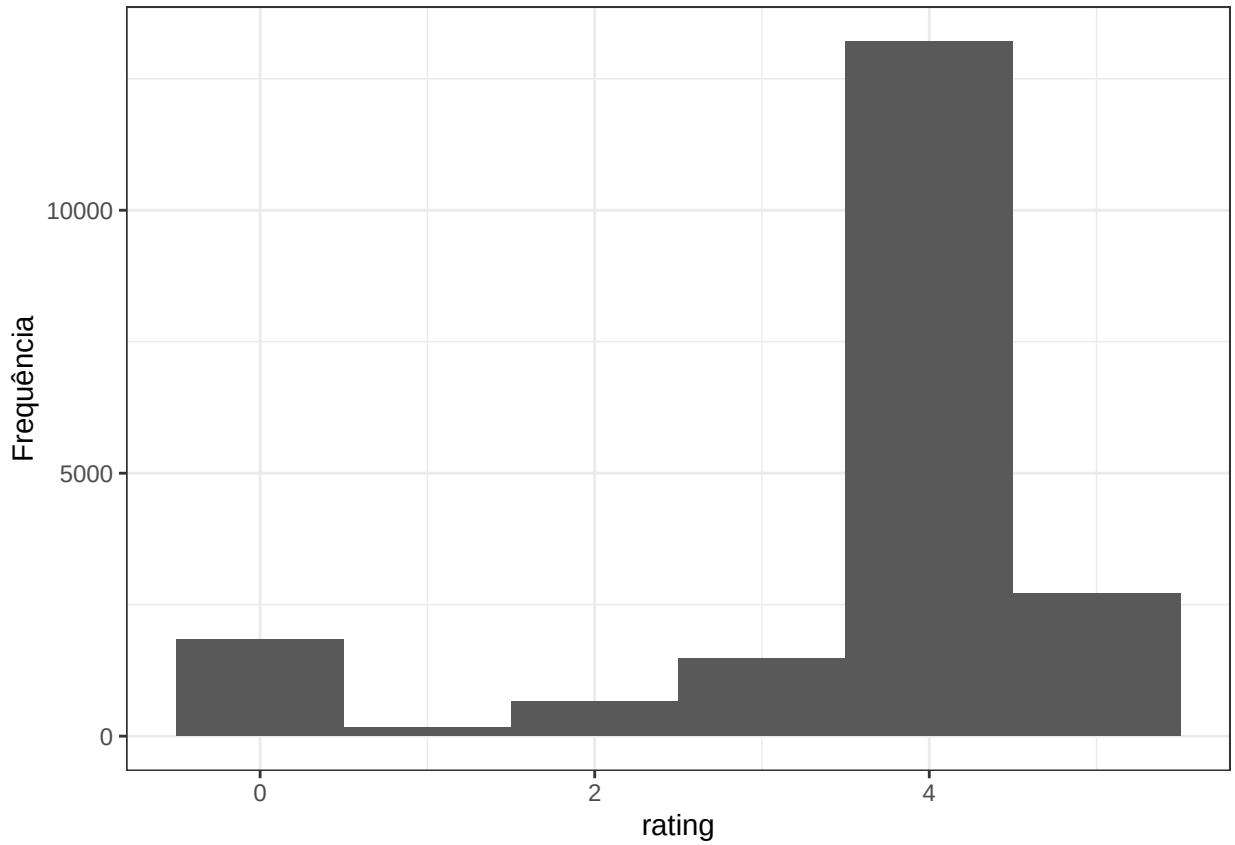
```
recipes2 <- recipes[-rwi, ]

num_ingredients <- recipes2$ingredients %>% sapply(function(x) length(x))
```

A variável resposta tem algumas notas (`rating`) faltando, supondo que esta ausência é aleatória, imputamos estes valores pela média. Além disso, observamos que há uma assimetria positiva na variável resposta, notamos também que a nota mínima recebida por uma receita é zero e máxima igual a 5.0.

```
recipes2 %>%
  select(rating) %>%
  summary()
```

```
##      rating
##  Min.   :0.000
##  1st Qu.:3.750
##  Median :4.375
##  Mean   :3.715
##  3rd Qu.:4.375
##  Max.   :5.000
##  NA's   :11
```

```
recipes2 <- recipes2 %>% mutate(rating, rating = ifelse(is.na(rating),
                                                        mean(rating, na.rm = TRUE), rating))

recipes2 %>%
  ggplot(aes(rating)) + geom_histogram(bins = 6) + ylab('Frequency') +
  ylab('Frequência') + theme_bw()
```



Através das variáveis originais e das criadas neste estudo (features) montamos um novo dataset denominado de `recipes3`. Com este novo dataset estudamos quais variáveis podem influenciar na nota de uma receita.

```
recipes3 <- data.table(recipes2$title, recipes2$rating, fat[-rwi], calories[-rwi],
                       protein[-rwi], sodium[-rwi], num_categories[-rwi],num_days[-rwi],
                       num_ingredients, num_process[-rwi], baked[-rwi],
                       conciseness[-rwi], desc[-rwi], desc_len[-rwi], old_recipe[-rwi],
                       phrases[-rwi], time[-rwi], vegetarian[-rwi],
                       recipes2$categories, recipes2$ingredients)

names(recipes3) <- c('title', 'rating', 'fat', 'calories', 'protein', 'sodium',
                     'num_categories', 'num_days', 'num_ingredients', 'num_process',
                     'baked', 'conciseness', 'desc', 'desc_len', 'old_recipe', 'phrases',
                     'time', 'vegetarian', 'categories', 'ingredients')

recipes3$desc <- recipes3$desc %>%
  as.integer() %>%
  as.factor()

recipes3$baked <- recipes3$baked %>%
  as.integer() %>%
  as.factor()
```

As 10 receitas com maior teor calórico são descritas abaixo.

```
caloric_recipes <- recipes3 %>%
  arrange(desc(calories)) %>% slice(1 : 10)
```

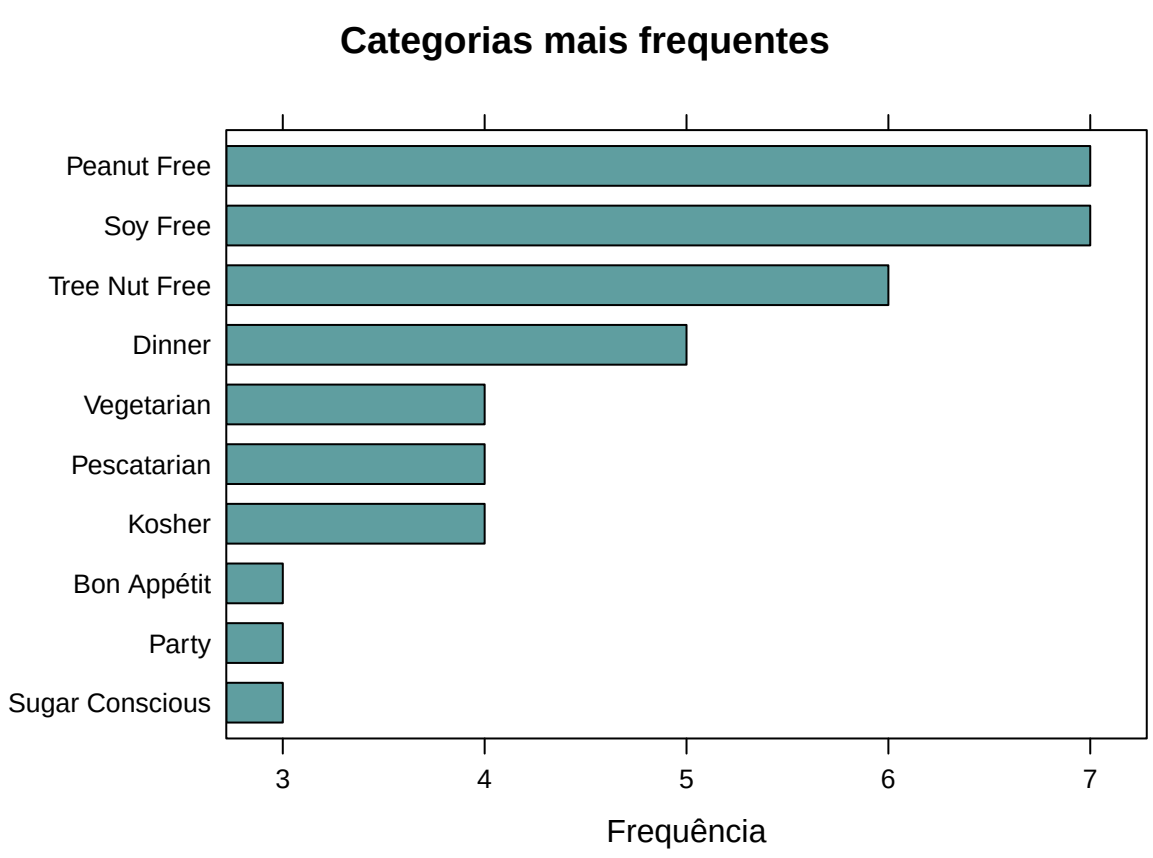
```
caloric_recipes
```

```
##                                                 title rating      fat
## 1                       Strawberry-Blueberry Napoleons    5.00 76.00000
## 2                         Southern-Style Fried Chicken    0.00 76.00000
## 3                         Southern-Style Fried Chicken    0.00 76.00000
## 4                                         Gaga Filling    0.00 22.17656
## 5                             Trenton Tomato Pie Pizza    5.00 22.17656
## 6                   Squash Blossoms Stuffed With Ricotta    5.00 22.17656
## 7  Crown Roast of Lamb with Shallots, Mustard and Mint    5.00 22.17656
## 8                          Winter Fruit and Nut Stuffing    5.00 44.00000
## 9   Sauteed Pork Chops with Sweet-and-Sour Red Cabbage    3.75 22.17656
## 10                                         Nutty Monk    0.00 18.00000
##    calories  protein    sodium num_categories num_days num_ingredients
## 1      2493 52.00000  444.0048             19     5598              10
## 2      2484 15.80889 1791.0000             13     2540               8
## 3      2484 15.80889 1791.0000             13     2540               8
## 4      2478 15.80889  444.0048             16     4902               8
## 5      2477 15.80889  444.0048              6     1224               8
## 6      2477 32.00000  425.0000             15     3772              15
## 7      2476 27.00000  177.0000              9     5598              14
## 8      2473 88.00000  444.0048             17     5598              14
## 9      2466 15.80889 1759.0000             15     4867              12
## 10     2457  4.00000   22.0000             14     3975               7
##    num_process baked conciseness desc desc_len old_recipe phrases time
## 1            6     1        2461    1        0          1       0  128
## 2            2     1         907    1        0          0       0   30
## 3            2     1         907    1        0          0       0   30
## 4            4     1         885    0       50          1       1   30
## 5            6     1         944    0      160          0       1   43
## 6            5     1         995    0      434          1       4   32
## 7            5     1        1355    1        0          1       0  106
## 8            4     1        1241    0      184          1       1  110
## 9            5     1        1574    0      111          1       1   55
## 10           3     1         855    0      152          1       1 1162
##    vegetarian
## 1           1
## 2           0
## 3           0
## 4           1
## 5           0
## 6           1
## 7           0
## 8           1
## 9           0
## 10          0
##
## 1  Milk/Cream, Mixer, Berry, Dairy, Dessert, Bake, Fourth of July, Vinegar, Summer, Chill, Phyllo/Pu
## 2                                                                                  Chicken, Poultry, Fry, Picn
## 3                                                                                  Chicken, Poultry, Fry, Picn
## 4                                 Dairy, Egg, Herb, Mushroom, Vegetable, Side, Vegetarian, Gourmet, Su
## 5
## 6                                                        Tomato, Quick & Easy, Dinner, Parmesan, Rico
## 7
```

```
## 8                                                    Fruit, Nut, Side, Bake, Thanksgiving, Stuffing/Dressi
## 9                                                              Onion, Pork, Roast, Dinner, Meat, Pork Chop
## 10                                                        Bitters, Liqueur, Alcoholic, Cocktail Party, Wedd
##
## 1
## 2
## 3
## 4
## 5
## 6                                                                                          1 garlic clove, minced,
## 7  3 tablespoons butter, room temperature, 1 tablespoon plus 1/2 teaspoon Dijon mustard, 2 teaspoons
## 8                                                    12 tablespoons (1 1/2 sticks) butter, 2 large Anjou p
## 9
## 10
```

Considerando as 10 receitas com maior teor calórico verificamos as principais categorias que estas receitas pertencem, respectivamente, são: (1) Peanut Free; (2) Soy Free; (3) Tree Nut Free; (4) Dinner; (5) Vegetarian; (6) Pescatarian; (7) Kosher; (8) Bon Appétit; (9) Party; (10) Sugar Conscious.

```
stats <- txt_freq(unlist(caloric_recipes$categories))
library(lattice)
stats$key <- factor(stats$key, levels = rev(stats$key))
barchart(key ~ freq, data = head(stats, 10),
         col = "cadetblue", main = "Categorias mais frequentes", xlab = "Frequência")
```



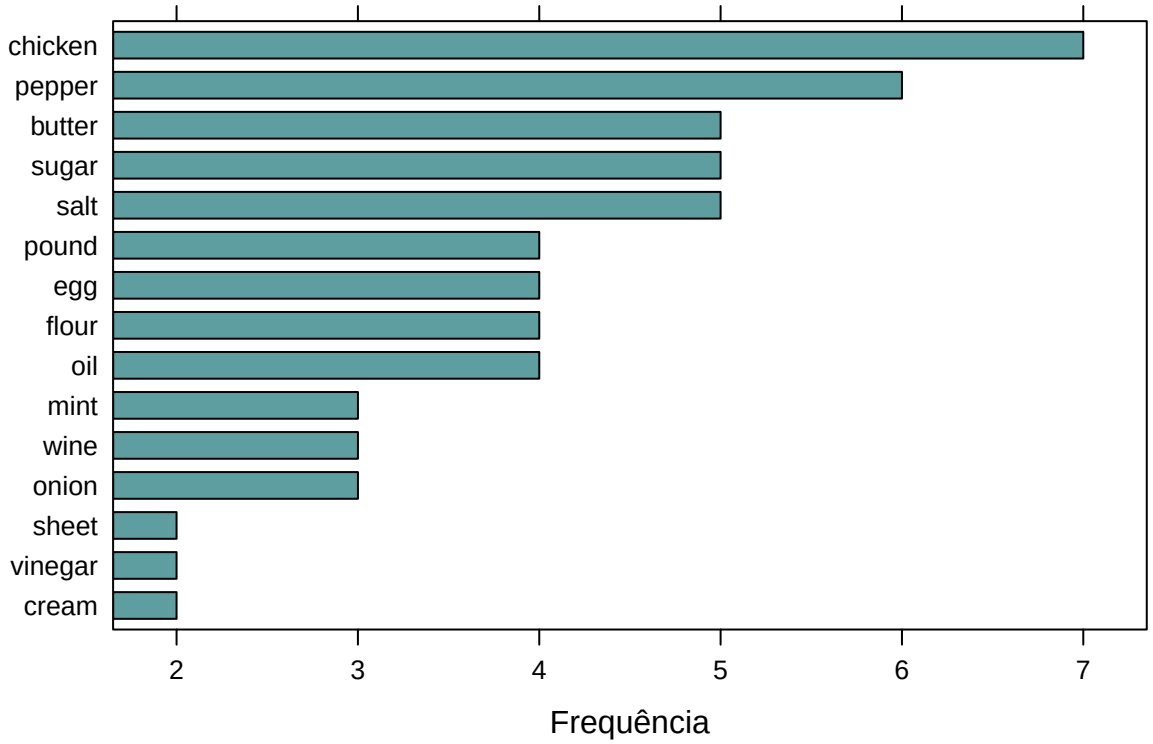

```
stats %>% slice(1 : 10)
```

```
##                key freq freq_pct
```

```
## 1      Peanut Free    7 5.109489
## 2         Soy Free    7 5.109489
## 3    Tree Nut Free    6 4.379562
## 4           Dinner    5 3.649635
## 5       Vegetarian    4 2.919708
## 6      Pescatarian    4 2.919708
## 7           Kosher    4 2.919708
## 8      Bon Appétit    3 2.189781
## 9            Party    3 2.189781
## 10 Sugar Conscious    3 2.189781
```

Considerando as 10 receitas com maior teor calórico podemos observar através do gráfico abaixo que o ingrediente mais presente nelas é `chicken` seguido de `pepper`, `butter`, `sugar`, `salt` e outros.

```
drop_words <- c('cup', 'cups', 'teaspoon', 'teaspoons', 'tablespoons', 'tablespoon',
                'peel', 'pound', 'ground', 'preserve', 'inch', 'piece', 'stick', 'slice',
                'large', 'optional', 'ounce', 'leave', 'purpose', 'powder', 'broth',
                'baking', 'ounces', 'pieces', 'cube')


ingredients <- caloric_recipes$ingredients %>%
  lapply(function(x) removeWords(x, drop_words)) %>%
  lapply(function(x) removeWords(x, stopwords('english'))) %>%
  lapply(function(x) str_replace_all(x, '[^[:alnum:]]', ' ')) %>%
  lapply(function(x) removeNumbers(x)) %>%
  lapply(function(x) stripWhitespace(x))


if(file.exists('english-ewt-ud-2.4-190531.udpipe')){
  ud_model <- udpipe_load_model('english-ewt-ud-2.4-190531.udpipe')
}else{
  ud_model_temp <- udpipe_download_model(language = "english")
  ud_model <- udpipe_load_model(ud_model_temp$file_model)
}

x <- udpipe_annotate(ud_model, x = unlist(ingredients))
x <- as.data.frame(x)

stats <- subset(x, upos %in% "NOUN")
stats <- txt_freq(x = stats$lemma)

stats$key <- factor(stats$key, levels = rev(stats$key))
barchart(key ~ freq, data = head(stats, 15),
         col = "cadetblue", main = "Ingredientes mais presentes nas receitas altamente calóricas", xlab
```



Seguindo o conceito da fronteira de Pareto, recomendamos três receitas segundo 5 critérios por ordem de prioridade, a saber: (1) maiores notas; (2) baixo teor de gordura; (3) baixo teor de calorias; (4) pouco sódio; (5) poucos processos; (6) receita simples (consisa).

```
recipes3[with(recipes3, order(-rating, fat, calories, sodium,
                              num_process, conciseness)), ] %>%
  slice(1 : 3)
```

```
##                          title rating fat calories protein sodium
## 1:          To Clarify Butter       5   0        0       0      0
## 2:          To Clarify Butter       5   0        0       0      0
## 3: Dashi (Japanese Sea Stock)       5   0        0       0     26
##    num_categories num_days num_ingredients num_process baked conciseness
## 1:              3     5598               1           1     0         232
## 2:              3     5598               1           1     0         232
## 3:              3     5598               3           1     0         406
##    desc desc_len old_recipe phrases time vegetarian
## 1:    1        0          1       0    3          0
## 2:    1        0          1       0    3          0
## 3:    0       77          1       1    3          0
##                     categories
## 1: Dairy,Quick & Easy,Gourmet
## 2: Dairy,Quick & Easy,Gourmet
## 3:   Soup/Stew,Spring,Gourmet
##
## 1:
## 2:
```

```
## 3: 6 cups cold water,1 oz (30 grams) kombu (dried kelp), about 20 square inches,2 (5-gram) packages
```

Modelagem

Com objetivo de identificar se há algum fator determinante para a alta nota de uma receita transformamos a nota numa variável dummy, em que receitas com notas iguais ou maiores que 4.0 são consideradas receitas bem avaliadas, caso contrário não. Em seguida, extraímos as regras mais importantes de uma árvore de decisão que fazem uma receita ser ou não bem avaliada.

Selecionando as regras mais importantes notamos, por exemplo, que as receitas que tem mais de 8.5 ingredientes tendem, com probabilidade igual a 0.59, de serem bem avaliadas, com cobertura de 57%. É interessante mencionar também que as receitas tendem a serem mal avaliadas, com probabilidade igual a 0.46, quando tem menos de 8.5 ingredientes, levam mais de 8.5 horas para serem feitas, estão no site a mais e 1726 dias e tem menos de 125.5 calorias.

Rule number: 22 [rating=0 cover=1776 (9%) prob=0.46] num_ingredients< 8.5 time>=8.5 num_days>=1726 calories< 125.5

```
recipes4 <- recipes3 %>%
  mutate(rating, rating = ifelse(rating >= 4.0, 1, 0) %>% as.factor())

fit_tree <- rpart(formula = rating ~.,
                  data = recipes4 %>%
                    dplyr::select(-categories, -ingredients))
#plot(fit_tree)
#text(fit_tree, cex = .8)

asRules(fit_tree)
```

```
##
##  Rule number: 3 [rating=1 cover=10615 (53%) prob=0.99]
##    title='Wichcraft's Roasted Turkey, Avocado, Bacon, Onion Relish, Arugula Sandwich
```

Neste ponto, selecionamoas as cem melhores receitas segundo a nota de avaliação para o estudo das categorias que possuem mais receitas.

```
recipes_top100 <- recipes3 %>%
  arrange(desc(rating)) %>%
  slice(1 : 100)
recipes_top100
```

```
##                                                                    title
## 1                                          Mahi-Mahi in Tomato Olive Sauce
## 2                            Mozzarella-Topped Peppers with Tomatoes and Garlic
## 3    Tuna, Asparagus, and New Potato Salad with Chive Vinaigrette and Fried Capers
## 4                                          Moroccan-Style Preserved Lemons
## 5                                         Fontina Mac with Squash and Sage
```

```
## 6                                  Pancetta Roast Chicken with Walnut Stuffing
## 7                Collard-and-Prosciutto Chicken Roulades Over Watercress Salad
## 8                    Southwest Corn Bread Stuffing with Corn and Green Chilies
## 9                                                                Jeweled Rice
## 10                                                          Short Rib Pot Pie
## 11                              Apricot-Pistachio Muffins Baked on the Grill
## 12                                 Asian Noodles with Barbecued Duck Confit
## 13                                          Chocolate-Mint Shamrock Shake
## 14                                                      Tropical Rum Punch
## 15                                                            Pastry Dough
## 16                                                     Radishes with Burrata
## 17                              Cranberry, Shallot, and Dried-Cherry Compote
## 18                                                    Blueberry Cheesecake
## 19                                   Grilled Corn with Lime-Cilantro Butter
## 20                                           Maple Butter-Pecan Ice Cream
## 21                Grilled Vegetable Antipasto with Herbed Chevre and Crostini
## 22                      Roasted Bell Peppers with Basil and Balsamic Vinegar
## 23                                                 Seasoned Nori Wrappers
## 24                                                       Orange Mint Julep
## 25                                                 Marbled Raspberry Cream
## 26                                                              Sangria IV
## 27                                                 Southwest Veggie Nachos
## 28                                                Chicken and Sausage Jambalaya
## 29          Salmon Confit in Olive Oil with Arugula Salad and Balsamic Vinegar
## 30                                              Cherry Cola Barbecue Sauce
## 31                                                     Golden Turkey Stock
## 32                                                  Fennel-Rubbed Pork Roast
## 33                                                  Honey-Cinnamon Ice Cream
## 34                                       Crab Cakes with Red Chili Mayonnaise
## 35                                  Ice Cream Soda with Lime, Mint and Ginger
## 36                                              My Favorite Roast Turkey
## 37                          Mashed Sweet Potatoes with Brown Sugar and Pecans
## 38                                   Breakfast Bowl With Quinoa and Berries
## 39                                                 Cheese Puffs Gougères
## 40                                              Ghostly Chocolate-Nut Bark
## 41                Rib-Eye Steaks with Radicchio, Pear, and Blue Cheese Salad
## 42                                                   Spicy Tahini Sauce
## 43                          Corn Salad With Hazelnuts, Pecorino, and Mint
## 44                                                 Habanero-Orange Salsa
## 45                                                 Seasoned Nori Wrappers
## 46                                                  Champagne Vinaigrette
## 47                                                Toasted Baguette Crumbs
## 48                                                              Veal Stock
## 49                                              Café Azul's Pastry Dough
## 50                           Dry-Rubbed Flank Steak with Grilled Corn Salsa
## 51                                                    Classic Coconut Cake
## 52                                                 Icy Lemon-Mint Parfaits
## 53                                                          Toffee Sauce
## 54                                                    Minty, Boozy Chicken
## 55                                                    Chicken Tikka Masala
## 56      Lemon Cornmeal Cake with Lemon Glaze and Crushed-Blueberry Sauce
## 57                                                          Pickled Onions
## 58               Haricots Verts With Poached Eggs And Tarragon Vinaigrette
## 59                  Vanilla Cheesecake Tartlets with Vanilla-Vodka Berries
```

```
## 60                                                    Mango and Red Pepper Chutney
## 61                     Grilled Butterflied Leg of Lamb with Lemon, Herbs, and Garlic
## 62                                                  Junior's Russian Dressing
## 63                                Giant Chocolate Candy Bar With Peanuts and Nougat
## 64                                                      Café Iguana Margarita
## 65                                                Country Pâté (Pâté de Campagne)
## 66                                                     Chocolate Marshmallow Eggs
## 67                                                           Hamburger Soup
## 68                                  Squid Ink Pasta with Shrimp, Nduja, and Tomato
## 69                                           Pork Chile Verde with Red Chile Salsa
## 70                                                       Red Vinegar Pickles
## 71                                                             Bee's Knees
## 72                                Flourless Pistachio Cake with Strawberry Meringue
## 73                                                      Wild Mushroom Pierogies
## 74                                                   Whole Fish Baked in Sea Salt
## 75                                        Rosemary and Thyme Braised Lamb Shoulder
## 76                               Shawarma-Spiced Chicken Pita with Tahini-Yogurt Sauce
## 77                                                       Plum-Almond Tartlets
## 78                                    Panna Cotta Parfaits with Raspberry Compote
## 79                                                                  Za'atar
## 80                                                      Creamed Corn with Bacon
## 81                                                 Our Favorite Texas Beef Chili
## 82                                                  Leftover-Roast-Chicken-Stock
## 83   'Wichcraft's Roasted Turkey, Avocado, Bacon, Onion Relish, & Aïoli on Ciabatta
## 84                                                 Concord Grape and Champagne Cocktails
## 85                                                             Buttered Polenta
## 86                                                                 Almond Crust
## 87                                                             Blackberry Syrup
## 88                                  Chris Lilly's Flank Steak and Shiitake Yakitori
## 89                                  Classic Tomato Toast with Mayonnaise and Chives
## 90                                                  Green Pozole with Chicken
## 91                                                                    Blatjang
## 92                   Daniel Boulud's Short Ribs Braised in Red Wine with Celery Duo
## 93                                                   Homemade Cultured Butter
## 94                                    Honey-Turmeric Pork with Beet and Carrot Salad
## 95                                 Grilled Tea-Brined Turkey with Tea-and-Lemon Gravy
## 96  Beef Tenderloin with Red Wine Sauce, Creamed Spinach, and Truffled French Fries
## 97                                                               Smoky Tomato Sauce
## 98                                                 Lime, Garlic, and Oregano Mojo
## 99                                                          Olive Focaccia Dough
## 100                                                 Tuscan Tuna Salad with Fennel
##     rating      fat   calories  protein     sodium num_categories num_days
## 1        5 22.17656   22.17656 22.17656   22.17656             17     3918
## 2        5  7.00000  107.00000  5.00000  344.00000              7     5598
## 3        5 33.00000  421.00000 10.00000  383.00000             17     3941
## 4        5 22.17656   22.17656 22.17656   22.17656              4     4263
## 5        5 22.17656   22.17656 22.17656   22.17656             23     2226
## 6        5 87.00000 1203.00000 89.00000  583.00000             13     4277
## 7        5 22.17656   22.17656 22.17656   22.17656             15     1260
## 8        5 15.00000  293.00000  7.00000  565.00000             15     5598
## 9        5 18.00000  517.00000  7.00000   20.00000             15     2414
## 10       5 22.17656   22.17656 22.17656   22.17656             12     2134
## 11       5 10.00000  247.00000  5.00000  185.00000             13     1995
## 12       5 25.00000  519.00000 14.00000 1237.00000             10     4072
```

```
## 13       5 22.17656   22.17656 22.17656   22.17656             7     2115
## 14       5  0.00000  230.00000  1.00000   26.00000            12     5598
## 15       5 16.00000  234.00000  3.00000   99.00000             5     5598
## 16       5 19.00000  235.00000 13.00000  388.00000            16     2067
## 17       5  2.00000  199.00000  1.00000  203.00000             6     5598
## 18       5 66.00000  954.00000 14.00000  767.00000            14     5598
## 19       5  9.00000  132.00000  2.00000  109.00000            18     3568
## 20       5 22.17656   22.17656 22.17656   22.17656            12     5598
## 21       5 22.17656   22.17656 22.17656   22.17656            26     3716
## 22       5  8.00000  103.00000  1.00000    4.00000             6     5598
## 23       5 22.17656   22.17656 22.17656   22.17656             3     4620
## 24       5 22.17656   22.17656 22.17656   22.17656             8     2134
## 25       5 18.00000  228.00000  2.00000   17.00000             7     5598
## 26       5  0.00000  215.00000  1.00000   11.00000            13     5598
## 27       5 22.17656   22.17656 22.17656   22.17656            15     2526
## 28       5 70.00000 1117.00000 56.00000 1632.00000            14     3233
## 29       5 22.17656   22.17656 22.17656   22.17656            19     5598
## 30       5  2.00000  279.00000  2.00000  792.00000            11     4208
## 31       5  5.00000   82.00000  8.00000   24.00000             8     4096
## 32       5 22.17656   22.17656 22.17656   22.17656            14     2205
## 33       5 16.00000  240.00000  2.00000   74.00000            24     3226
## 34       5 33.00000  640.00000 21.00000 1183.00000             9     5598
## 35       5  7.00000  347.00000  3.00000   89.00000             7     5598
## 36       5 10.00000  225.00000 28.00000 2397.00000             5     1864
## 37       5  7.00000  386.00000  5.00000  435.00000            17     5598
## 38       5  7.00000  173.00000  5.00000    3.00000            18     1447
## 39       5  6.00000   84.00000  2.00000  146.00000            14     2912
## 40       5 20.00000  331.00000  4.00000   23.00000             5     1892
## 41       5 75.00000  944.00000 50.00000  380.00000            15     4460
## 42       5 22.17656   22.17656 22.17656   22.17656             7     2346
## 43       5 22.17656   22.17656 22.17656   22.17656            17     1244
## 44       5 22.17656   22.17656 22.17656   22.17656            20     3506
## 45       5 22.17656   22.17656 22.17656   22.17656             3     4620
## 46       5 73.00000  749.00000  1.00000  937.00000             4     3871
## 47       5  0.00000   14.00000  1.00000   30.00000             7     5598
## 48       5  1.00000   29.00000  4.00000   57.00000             6     5598
## 49       5 19.00000  280.00000  4.00000  339.00000             2     5598
## 50       5 23.00000  494.00000 41.00000 1136.00000             9     1581
## 51       5 44.00000  860.00000  8.00000  343.00000            18     3233
## 52       5  6.00000  229.00000  2.00000   18.00000            17     5009
## 53       5 28.00000  395.00000  1.00000   88.00000            12     2235
## 54       5 41.00000  607.00000 36.00000  143.00000            11     1416
## 55       5 53.00000  687.00000 38.00000 1351.00000            14     2472
## 56       5 13.00000  405.00000  6.00000  383.00000            12     3941
## 57       5  0.00000   32.00000  1.00000   23.00000            24     3172
## 58       5 27.00000  314.00000 11.00000  484.00000            22     2021
## 59       5 19.00000  315.00000  3.00000  261.00000             9     5598
## 60       5  6.00000  257.00000  3.00000  687.00000            10     5598
## 61       5 41.00000  572.00000 46.00000  859.00000            11     5598
## 62       5 26.00000  337.00000  2.00000 1303.00000             3     2246
## 63       5 23.00000  909.00000 30.00000 1302.00000             7     1157
## 64       5  0.00000  148.00000  0.00000   82.00000            10     5598
## 65       5 33.00000  397.00000 18.00000  695.00000            15     4032
## 66       5  1.00000   68.00000  1.00000    5.00000            13     3590
```

```
## 67       5 21.00000  428.00000 24.00000  445.00000             19     1478
## 68       5 22.00000  603.00000 31.00000  867.00000             14     1764
## 69       5 27.00000  431.00000 25.00000  214.00000             17     3263
## 70       5  1.00000   88.00000  3.00000 3194.00000             10     3415
## 71       5  0.00000  136.00000  0.00000    1.00000              8     5598
## 72       5 26.00000  460.00000 15.00000   66.00000             10     1737
## 73       5 11.00000  158.00000  3.00000   38.00000             18     5598
## 74       5 38.00000  553.00000 48.00000  444.00480             17     4398
## 75       5 80.00000 1012.00000 57.00000 1287.00000              3     2087
## 76       5 25.00000  497.00000 42.00000 1060.00000             11     1423
## 77       5 22.17656   22.17656 22.17656   22.17656              9     3459
## 78       5 15.00000  290.00000  3.00000   21.00000             15     4620
## 79       5  0.00000    4.00000  0.00000  210.00000              5     5598
## 80       5 28.00000  337.00000  8.00000  302.00000              6     5598
## 81       5 39.00000  837.00000 15.80889 2198.00000              8     1940
## 82       5  3.00000   34.00000  1.00000   40.00000              7     3954
## 83       5 87.00000 1405.00000 55.00000 1336.00000             16     4780
## 84       5  1.00000  688.00000  3.00000   18.00000             11     4816
## 85       5  8.00000  213.00000  3.00000  612.00000             17     5040
## 86       5 22.17656 1919.00000 29.00000  610.00000              7     5598
## 87       5  0.00000  131.00000  1.00000    5.00000              3     2233
## 88       5  6.00000  194.00000 17.00000  932.00000             13     2430
## 89       5 12.00000  194.00000  4.00000  713.00000             16     1619
## 90       5 22.17656   22.17656 22.17656   22.17656              7     5598
## 91       5  6.00000  342.00000  5.00000  437.00000             21     4964
## 92       5 22.17656   22.17656 22.17656   22.17656             12     5598
## 93       5 22.17656   22.17656 22.17656   22.17656              3     1164
## 94       5 22.17656   22.17656 22.17656   22.17656              6     1640
## 95       5 53.00000 1600.00000 15.80889  444.00480              8     5522
## 96       5 43.00000  703.00000 37.00000  422.00000             17     5566
## 97       5 22.17656   22.17656 22.17656   22.17656              8     2382
## 98       5 20.00000  204.00000  1.00000  369.00000             10     5598
## 99       5  6.00000  229.00000  6.00000  421.00000              4     5598
## 100      5 34.00000  454.00000 23.00000  393.00000             18     4780
##     num_ingredients num_process baked conciseness desc desc_len old_recipe
## 1                10           2     0         851    0      113          1
## 2                 7           2     1         448    1        0          1
## 3                16           5     0        1559    0      101          1
## 4                 4           2     0         401    0      387          1
## 5                12           1     0         873    0      199          0
## 6                16           5     1        1027    0       76          1
## 7                10           3     1         900    0      457          0
## 8                12           5     1        1258    0       72          1
## 9                17          11     0        2596    0      144          0
## 10               16          11     1        2446    0      157          0
## 11               12           5     0        1038    0      421          0
## 12               14          10     1        1368    0      371          1
## 13                5           1     0         249    0      206          0
## 14                7           1     0         264    0       69          1
## 15                5           4     0        1527    0       84          1
## 16                6           4     0         356    0      152          0
## 17               10           4     0         705    1        0          1
## 18               12           5     1        1376    0       67          1
## 19                7           3     0         271    0      195          0
```

```
## 20   8  3  0  1019  1    0  1
## 21  32 14  0  2104  0  397  1
## 22   6  1  0   438  0   66  1
## 23   3  1  0   255  1    0  1
## 24   6  2  0   467  0   63  0
## 25   3  1  0   304  1    0  1
## 26  13  2  0   337  0  415  1
## 27  14  1  1   349  0   55  0
## 28  18  2  1   907  1    0  0
## 29   6  3  1   719  0   85  1
## 30   9  1  0   575  0  110  1
## 31   8  3  1   952  0   98  1
## 32   6  4  1   493  0   68  0
## 33   8  4  0  1612  0 1062  0
## 34  11  3  0   881  0  122  1
## 35   8  2  0   610  0  113  1
## 36  23  9  1  2025  1    0  0
## 37   9  4  1   673  1    0  1
## 38   4  1  0   174  0  370  0
## 39  11  8  1  1582  0  284  0
## 40   6  5  1  1465  1    0  0
## 41  10  2  0   461  0   69  1
## 42   9  1  0   536  1    0  0
## 43  12  4  1   813  0   59  0
## 44   6  2  0   307  0  128  0
## 45   3  1  0   255  1    0  1
## 46   9  1  0   280  1    0  1
## 47   1  2  1   280  0  191  1
## 48  11  6  1  1461  0   64  1
## 49   6  3  0   771  0  367  1
## 50  19  5  1  1118  0   96  0
## 51  14  4  1  1494  0  442  0
## 52   8  3  0   833  1    0  1
## 53   4  1  0   243  0  132  0
## 54  16  4  1  1429  0  206  0
## 55  18  7  0  1381  0  117  0
## 56  13  4  1  1302  0  109  1
## 57   4  1  0   257  0  119  0
## 58  11  5  0  1101  0   81  0
## 59  11  5  1   932  1    0  1
## 60  15  3  0   695  1    0  1
## 61   9  5  1   998  0   87  1
## 62   7  2  0   343  0  261  0
## 63  13 13  1  3180  0  266  0
## 64   7  2  0   353  0   53  1
## 65  17  9  1  1683  0   87  1
## 66   7  8  1  1845  0  860  0
## 67  17  5  0   979  0  293  0
## 68  11  5  0  1780  0   80  0
## 69  15  4  0  1345  0  572  0
## 70  16  1  0   291  1    0  0
## 71   4  1  0    96  1    0  1
## 72  14  9  1  2241  0  173  0
## 73  12  6  0  1674  0  305  1
```

```
## 74              11           5     1        1005     0       80           1
## 75              14           8     1        1538     0      524           0
## 76              21           5     1        1035     0      215           0
## 77              13           2     1         631     1        0           0
## 78               8           1     0         604     1        0           1
## 79               4           2     0         141     0      289           1
## 80               8           3     0         715     0       50           1
## 81              20           7     0        2273     0      351           0
## 82               9           2     0         409     0      327           1
## 83              17           5     1        1746     0      317           1
## 84               7           3     1         650     1        0           1
## 85               5           1     0         334     0      286           1
## 86               7           1     0         375     0      118           1
## 87               4           2     0         335     0       78           0
## 88              14           5     0        1039     0      851           0
## 89               7           1     0         238     0       78           0
## 90              20           5     0        1496     1        0           1
## 91              12           2     0         652     0       43           1
## 92              16           6     1        1841     0      355           1
## 93               4           5     0        1357     0      273           0
## 94              13           4     0         946     0      139           0
## 95              14           5     1        2107     0      248           1
## 96              18           5     1        1758     0       60           1
## 97               9           3     0         549     0       68           0
## 98               6           1     0         138     0      180           1
## 99               8           2     0         592     0      126           1
## 100             13           2     0         484     0      234           1
##     phrases time vegetarian
## 1         1   22          0
## 2         0   53          0
## 3         1  265          0
## 4         3    5          0
## 5         3   41          1
## 6         1   95          0
## 7         4   24          0
## 8         1  108          1
## 9         2  111          1
## 10        2  260          0
## 11        3   25          1
## 12        3   54          0
## 13        2    0          0
## 14        1    0          0
## 15        1   24          0
## 16        1    0          1
## 17        0   57          0
## 18        1  320          0
## 19        3   10          1
## 20        0   77          0
## 21        5   72          1
## 22        1   34          0
## 23        0    0          0
## 24        1   48          0
## 25        0   96          0
## 26        3   30          0
```

```
## 27        2    6          1
## 28        0   87          0
## 29        0    5          0
## 30        2   58          0
## 31        2  212          0
## 32        1   80          0
## 33       10  128          1
## 34        1  223          0
## 35        1    2          0
## 36        0 1857          0
## 37        0   66          1
## 38        4    0          1
## 39        2   92          1
## 40        0  102          0
## 41        1    3          0
## 42        0    6          0
## 43        1   26          1
## 44        1    0          1
## 45        0    0          0
## 46        0    0          0
## 47        1   18          0
## 48        2  363          0
## 49        3   24          0
## 50        1   25          0
## 51        4   71          1
## 52        0  245          1
## 53        2    4          1
## 54        2   80          0
## 55        2  223          0
## 56        2   30          1
## 57        1   30          1
## 58        1   20          1
## 59        0  109          0
## 60        0   86          0
## 61        1   74          0
## 62        2    0          1
## 63        2  161          0
## 64        1    0          0
## 65        1   76          0
## 66        8  773          0
## 67        7   20          0
## 68        2   27          0
## 69        8   97          0
## 70        0  192          1
## 71        0    0          0
## 72        1   95          1
## 73        3   88          1
## 74        1   44          0
## 75        5  710          0
## 76        2   20          0
## 77        0   20          1
## 78        0   82          0
## 79        3    0          0
## 80        1   10          0
```

```
## 81        5  149          0
## 82        3   48          0
## 83        3    9          0
## 84        0  216          0
## 85        2   47          1
## 86        1   48          0
## 87        1   15          0
## 88        8   32          0
## 89        1    0          1
## 90        0   82          0
## 91        1   96          1
## 92        3   91          0
## 93        2 1205          0
## 94        1  300          0
## 95        3 1304          0
## 96        1  134          0
## 97        1   34          0
## 98        2    0          0
## 99        3   53          0
## 100       4    0          0
##
## 1
## 2
## 3
## 4
## 5                                                                  Cheese, Herb, Pasta, Low Cal, Di
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19                                                                                                 S
## 20
## 21                                                  Cheese, Dairy, Herb, Pepper, Tomato, Vegetable
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29                                                                                  Salad, Fish
## 30
## 31
## 32
## 33  Milk/Cream, Ice Cream Machine, Dairy, Dessert, Freeze/Chill, Christmas, Fourth of July, Thanksgiv
```

```
## 34
## 35
## 36
## 37
## 38                                                                                                          Harpe
## 39
## 40
## 41
## 42
## 43
## 44                                                                              Sauce, Side, Vegetarian, Quick &
## 45
## 46
## 47
## 48
## 49
## 50
## 51
## 52
## 53
## 54
## 55
## 56
## 57                    Onion, Side, Low Fat, Vegetarian, Quick & Easy, High Fiber, Cinco de Mayo, Backyard
## 58                                                                    Egg, Breakfast, Brunch, Side, Poach, Veget
## 59
## 60
## 61
## 62
## 63
## 64
## 65
## 66
## 67                                                                                                   HarperCollin
## 68
## 69
## 70
## 71
## 72
## 73
## 74
## 75
## 76
## 77
## 78
## 79
## 80
## 81
## 82
## 83
## 84
## 85
## 86
## 87
```

```
## 88
## 89
## 90
## 91                                                                                                  Condiment/Spi
## 92
## 93
## 94
## 95
## 96
## 97
## 98
## 99
## 100
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21  Safflower oil or canola oil, for oiling the grill, 1/2 cup olive oil, 1/4 cup balsamic vinegar, 5
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
## 37
## 38
## 39
## 40
```

```
## 41
## 42
## 43
## 44
## 45
## 46
## 47
## 48
## 49
## 50
## 51
## 52
## 53
## 54
## 55
## 56
## 57
## 58
## 59
## 60
## 61
## 62
## 63
## 64
## 65
## 66
## 67
## 68
## 69
## 70
## 71
## 72
## 73
## 74
## 75
## 76
## 77
## 78
## 79
## 80
## 81
## 82
## 83
## 84
## 85
## 86
## 87
## 88
## 89
## 90
## 91
## 92
## 93
## 94
```

```
## 95
## 96
## 97
## 98
## 99
## 100
```

Com a função abaixo fizemos um crawler com objetivo de verificar quais das 100 primeiras receitas estão no site epicurious. A função `epicurious` recebe o nome das receitas (`title`) e retorna o link da receitas no site, caso ela se encontre nele, é retornado também uma variável booleana que indica se a variável foi encontrada na busca do site ou não.

```
title_top100 <- recipes_top100 %>% dplyr::select(title)

urls <- rbindlist(lapply(title_top100, function(x){
  url <- paste('https://www.epicurious.com/search/', x, sep = '')
  tibble(url)
}), fill = TRUE)


epicurious <- function(recipe_title){
  recipe_title <- recipe_title %>% stringi::stri_trans_general('Latin-ASCII')
  recipe_title <- recipe_title %>% str_replace('\'s', '')

  names_recipes <- paste(paste('recipe', seq_along(recipe_title),
                               sep = ''), '.xml', sep = '')
  out <- rep(0, length(recipe_title))

  for(i in seq_along(recipe_title)){
    recipe_url <- recipe_title[i] %>%
      as.character()

    recipe_url <- paste('https://www.epicurious.com/search/', recipe_url, sep = '')

    if(!(file.exists(names_recipes[i]))){
      download.file(recipe_url, destfile = names_recipes[i])
    }

    recipes_xml <- read_html(names_recipes[i]) %>%
      htmlParse()

      links <- getHTMLLinks(recipes_xml, externalOnly = TRUE,
                            xpQuery = "//a/@href", baseURL = docName(recipes_xml),
                            relative = FALSE)
      url_recipe <- links[links %>% str_detect('/recipes/food/views/')][1]
      out[i] <- paste('https://www.epicurious.com', url_recipe, sep = '')

      first_word_recipe_title <- recipe_title[i] %>%
        str_replace('-', ' ') %>%
        str_replace('[:punct:]', '') %>%
        str_to_lower() %>%
        removeWords(stopwords('english')) %>%
        str_replace_all('[^[:alnum:]]', ' ') %>%
        removeNumbers() %>%
        stripWhitespace() %>%
```

```
        word()

      out[i] <- ifelse(out[i] %>%
                         str_detect(first_word_recipe_title), out[i], 0)
  }
  list('link' = out,
       'inside_epicurious' = ifelse(out != 0, TRUE, FALSE) )
}
```

O nome das receitas que estão no site podem ser vistos abaixo:

A seguir, vemos as trinta categorias que possuem mais receitas.

```
cat_existents <- unique(unlist(recipes_top100[recipes_in_epicurious$inside_epicurious,
num_recipes_cat <- lapply(recipes_top100[recipes_in_epicurious$inside_epicurious,
                                         ]$categories,
                                  function(x) factor(x, cat_existents))

num_recipes_cat <- lapply(num_recipes_cat, function(x) dummy_cols(x))
num_recipes_cat <- lapply(num_recipes_cat, function(x) data.matrix(x,
                                                                   rownames.force = NA))

num_recipes_cat <- t(sapply(num_recipes_cat, function(y) apply(y, 2, sum)))
num_recipes_cat <- apply(num_recipes_cat, 2,  function(x) sum(x))[-1]

recipes_by_cat <- data.frame(categorie = as.factor(cat_existents),
                         num_recipes_cat = num_recipes_cat) %>%
  arrange(desc(num_recipes_cat))

g <- recipes_by_cat[1:30, ] %>% ggplot(aes(x = reorder(categorie, num_recipes_cat), y = num_recipes_cat
  geom_bar(stat = 'identity') +  coord_flip() + theme_bw() +
  xlab('Frequência') + ylab('Categoria') + ggtitle('Trinta categorias com mais receitas')
g
```

Considerações, limitações da solução e feedback

O presente desafio deu oportunidade de mostrar conhecimento em áreas diversas de ciência de dados, desde a limpeza de dados, passando pela necessidade de conhecimento de especialistas até a identificação de fatores que influenciam uma variável resposta.

Esta solução não é definitiva podendo assim, ser aprimorada em tempo oportuno. Portanto, há algumas limitações que podem ser melhoradas em etapas futuras, tais como:

- Definição do limiar para detecção de erro de medição - esta etapa pode ser facilmente refinada quando se há conhecimento melhorado do domínio, disponibilidade de consulta a especialista (cliente); ou ainda através de investigação de modelos estatísticos avançados não usuais na literatura de ciência de dados;
- O crawler que verifica a existência de uma receita no site epicurious é um protótipo, podendo ser melhorado;
- A sexta questão pode ser resolvia através de sistemas de recomendação, por exemplo, cgontent-based filtering. Porém há limitações técnicas e teóricas, temporárias, que impossibilitam a solução do problema no tempo estabelecido;
- Feedback - A variável `rating` de modo purista deve ser analisado por um modelo estatístico cuja distribuição de probabilidade tenha suporte no intervalo [0, 5], ou seja, modelos dessa natureza garantem matematicamente que as predições e previsões não estaram fora do domínio da variável. Portanto, a detecção de fatores importantes que influenciam na nota de uma receita pode ser feita através de modelos desta natureza. Contudo, a carga teórica para apredizagem de modelos deste tipo pode durar de uma semana a um mês. Uma possível solução para isto é perder um pouco de informação binarizando

(tornando dummy) a variável em estudo com isso, podemos aplicar um modelo de regressão logística ou árvore de decisão para extração de regras. Sendo o último a abordagem utilizada.

Por fim, abaixo temos o tempo para solução de todas as questões obrigatórias.

```
Sys.time() - start
```

```
## Time difference of 1.327477 mins
```