

# Modelagem XGBOOST para a classificação de avaliações de livros

Ana Luzielma Dias Campos

Jaylhane Veloso Nunes

Raianny da Silva Soares

## Introdução

Quando se está procurando uma nova leitura, uma das coisas que pode ser observada é a avaliação do livro. De acordo com ela, pode-se ter uma ideia inicial se o livro é bom ou se ele segue uma dinâmica que se está habituado, já que muitas pessoas deram uma nota alta de avaliação. Pensando sobre isso, levantamos o questionamento: seria possível prever se um livro é bom sem ter acesso a nota da avaliação? Dessa forma pensamos em realizar um modelo para classificar a avaliação de um livro. Para esta tarefa utilizaremos o seguinte conjunto de dados: Goodreads-books| Kaggle e como inspiração para construção do modelo utilizaremos o seguinte guia: Tune xgboost models with early stopping to predict shelter animal status| Julia Silge.

Assim, uma das possibilidades é fazer uma categorização das avaliações dos livros em “Ruim”, “Bom” e “Ótimo”, considerando respectivamente os intervalos das notas como,  $[0, 3.5)$ ,  $[3.5, 4]$  e  $(4, 5]$ , e a partir daí prever a avaliação dos livros utilizando o XGboost. O critério de intervalo para as categorias das notas foi definido subjetivamente ao acaso entre as participantes do grupo, tentando balancear a quantidade de observações que ficariam em cada grupo.

Além disso, como o objetivo é classificar os livros sem olhar as avaliações, as notas não farão parte do modelo, elas serão utilizadas apenas para criar as categorias e estamos supondo que de alguma forma as variáveis como número de páginas, idade do livro, editora, quantidade de notas de avaliações e quantidade de avaliações escritas estão relacionadas com a avaliação do livro.

## Análise exploratória

- Limpeza dos dados

```
library(knitr)
opts_chunk$set(message=FALSE,
                warning=FALSE,
                echo = TRUE,
                #results="asis",
                cache = TRUE,
                dev = "png",
                dpi = 500)

#Pacotes necessários
library(tidyverse)
library(tidymodels)
library(lubridate)
library(vip)
```

```
library(GGally)

theme_set(theme_light(base_family = "IBMPlexSans"))

# Carregando os dados

livros <- read.csv("./Conjunto de Dados/books.csv",
                  encoding = "UTF-8",
                  header = TRUE) %>%

  select(-bookID,
         -title,
         -isbn,
         -isbn13) %>%
  na.omit()
```

- Verificando a língua

Analisando o conjunto de dados percebemos que algumas variáveis nos códigos da linguagem estavam em formato de numeração e precisaram ser removidas.

```
summary(as.factor(livros$language_code))
```

```
## 9780674842113 9780851742717 9781563841552 9781593600112      ale
##           1           1           1           1           1
##          ara          en-CA          en-GB          en-US          eng
##           1           7          214          1408          8908
##          enm          fre          ger          gla          glg
##           3          144           99           1           1
##          grc          ita          jpn          lat          msa
##          11           5           46           3           1
##          mul          nl          nor          por          rus
##          19           1           1          10           2
##          spa          srp          swe          tur          wel
##          218           1           2           1           1
##          zho
##          14
```

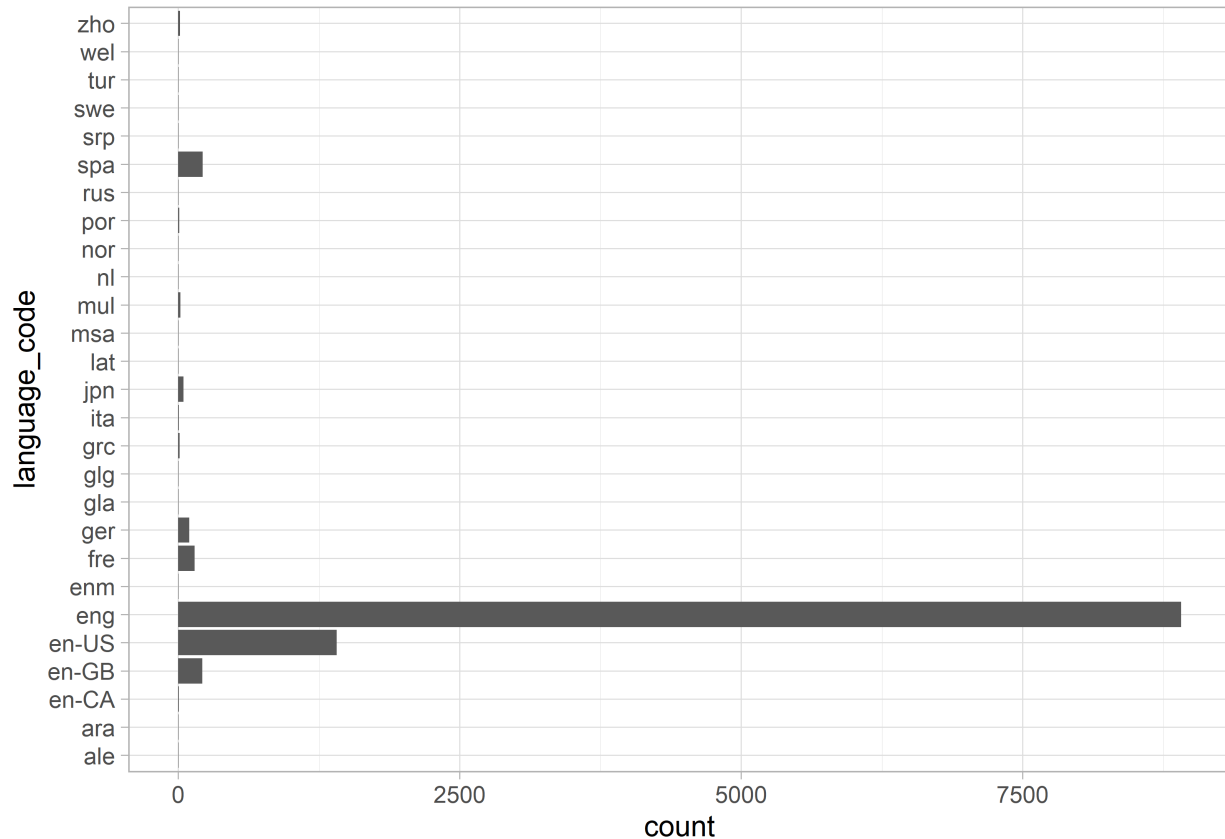
```
linhas_invalidas_de_language_code<- livros %>%
  filter(language_code=="9780674842113"|
         language_code=="9780851742717"|
         language_code=="9781563841552"|
         language_code=="9781593600112"|
         language_code=="")

livros <- livros %>%
  filter(!(language_code%in%linhas_invalidas_de_language_code$language_code))

rm(linhas_invalidas_de_language_code)
```

Fazendo o gráfico de barras para ver a variação dessas linguagens, obtemos o seguinte resultado:

```
livros %>%
  ggplot(aes(language_code))+
  geom_bar()+
  coord_flip()
```



Como há muito pouca variação linguística comparado ao grupo inglês, dividiremos a categoria de `language_code` em duas: inglês e outros.

```
livros <- livros %>%
  mutate(publication_date = mdy(publication_date),
         average_rating = as.double(average_rating),
         num_pages = as.integer(num_pages),
         book_age = year(today())-year(publication_date),
         month_publication = as.factor(month(publication_date)),
         year_publication = as.factor(year(publication_date)),
         language_code = factor(
           ifelse(language_code %in% c("enm",
                                     "eng",
                                     "en-US",
                                     "en-GB",
                                     "en-CA"),
                 "English", "Other")
         )
  ) %>%
  select(-authors, -publisher) %>%
  na.omit()
```

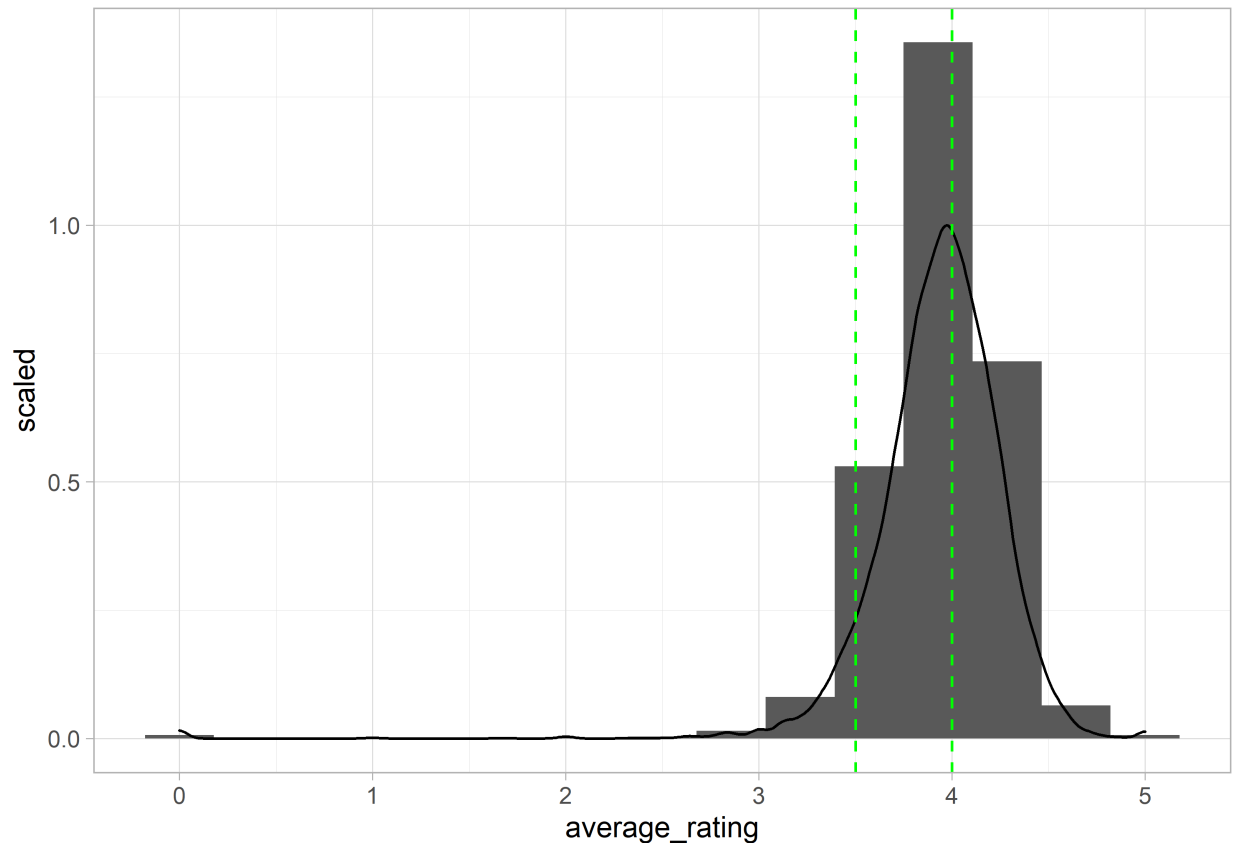
```
summary(livros)
```

```
## average_rating language_code num_pages ratings_count
## Min. :0.000 English:10539 Min. : 0.0 Min. : 0
## 1st Qu.:3.770 Other : 582 1st Qu.: 192.0 1st Qu.: 104
## Median :3.960 Median : 299.0 Median : 745
## Mean :3.934 Mean : 336.3 Mean : 17945
## 3rd Qu.:4.140 3rd Qu.: 416.0 3rd Qu.: 4996
## Max. :5.000 Max. :6576.0 Max. :4597666
##
## text_reviews_count publication_date book_age month_publication
## Min. : 0.0 Min. :1900-01-01 Min. : 2.00 9 :1278
## 1st Qu.: 9.0 1st Qu.:1998-07-17 1st Qu.: 17.00 10 :1212
## Median : 47.0 Median :2003-03-01 Median : 19.00 1 :1057
## Mean : 542.1 Mean :2000-08-29 Mean : 21.83 4 : 991
## 3rd Qu.: 238.0 3rd Qu.:2005-10-01 3rd Qu.: 24.00 5 : 922
## Max. :94265.0 Max. :2020-03-31 Max. :122.00 6 : 879
## (Other):4782
##
## year_publication
## 2006 :1700
## 2005 :1260
## 2004 :1069
## 2003 : 931
## 2002 : 798
## 2001 : 656
## (Other):4707
```

- Separando os grupos

Temos 50% das observações estão entre [0,3.96] e o 1º Q é 3.77, que é bem próximo, mostrando que há uma concentração de avaliações, verificando o histograma dessa variável temos:

```
livros %>%
  ggplot(aes(x=average_rating, after_stat(scaled)))+
  geom_histogram(aes(y=..density..),
                 bins = 15)+
  geom_density()+
  geom_vline(xintercept = c(3.5,4), color = "green", lty=2)
```



E conferindo a quantidade de observações menores de 3 temos:

```
livros %>%
  filter(average_rating<3.5) %>%
  count()
```

```
summary(livros$average_rating)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.000   3.770   3.960   3.934   4.140   5.000
```

```
quantile(livros$average_rating,.67)
```

```
## 67%
## 4.07
```

```
livros %>%
  filter(average_rating>4) %>%
  count()
```

Dessa forma, trabalharemos apenas com três categorias, “Ruim”, “Bom” e “Ótimo”, considerando respectivamente os intervalos de nota de  $[0, 3.5)$ , de  $[3.5, 4]$  e de  $(4, 5]$ , uma vez que pelo histograma é notado a distribuição nas avaliações de 3 a 5.

Sendo assim, nosso conjunto de dados final é composto por três categorias: “Ruim”, “Bom” e “Ótimo”. Aplicando no conjunto de dados:

```

livros <- livros %>%
  mutate(
    book_rating =
      case_when(average_rating<3.5 ~ "Ruim",
                average_rating<=4 ~ "Bom",
                TRUE ~ "Ótimo")
  ) %>%
  select(-average_rating)

livros %>%
  group_by(book_rating) %>%
  count()

```

*# Salvando os dados atuais*

```

write.csv(livros,
          "./Conjunto de Dados/books_t.csv",
          fileEncoding = "UTF-8",
          row.names = FALSE)

```

## Análise Descritiva e Exploratória

*# Carregando conjunto de dados após a limpeza*

```

livros <- read.csv("./Conjunto de Dados/books_t.csv",
                  encoding = "UTF-8") %>%
  mutate_if(is.character,factor) %>%
  mutate(month_publication=factor(month_publication),
         year_publication=factor(year_publication),
         book_rating=factor(book_rating,
                             levels = c("Ótimo","Bom","Ruim")))

```

- Separando em Treino e Teste

```

set.seed(1904, kind = "Mersenne-Twister", normal.kind = "Inversion")

livros_split <- initial_split(livros, prop = .75, strata = book_rating)

livros_treino <- training(livros_split)

livros_teste <- testing(livros_split)

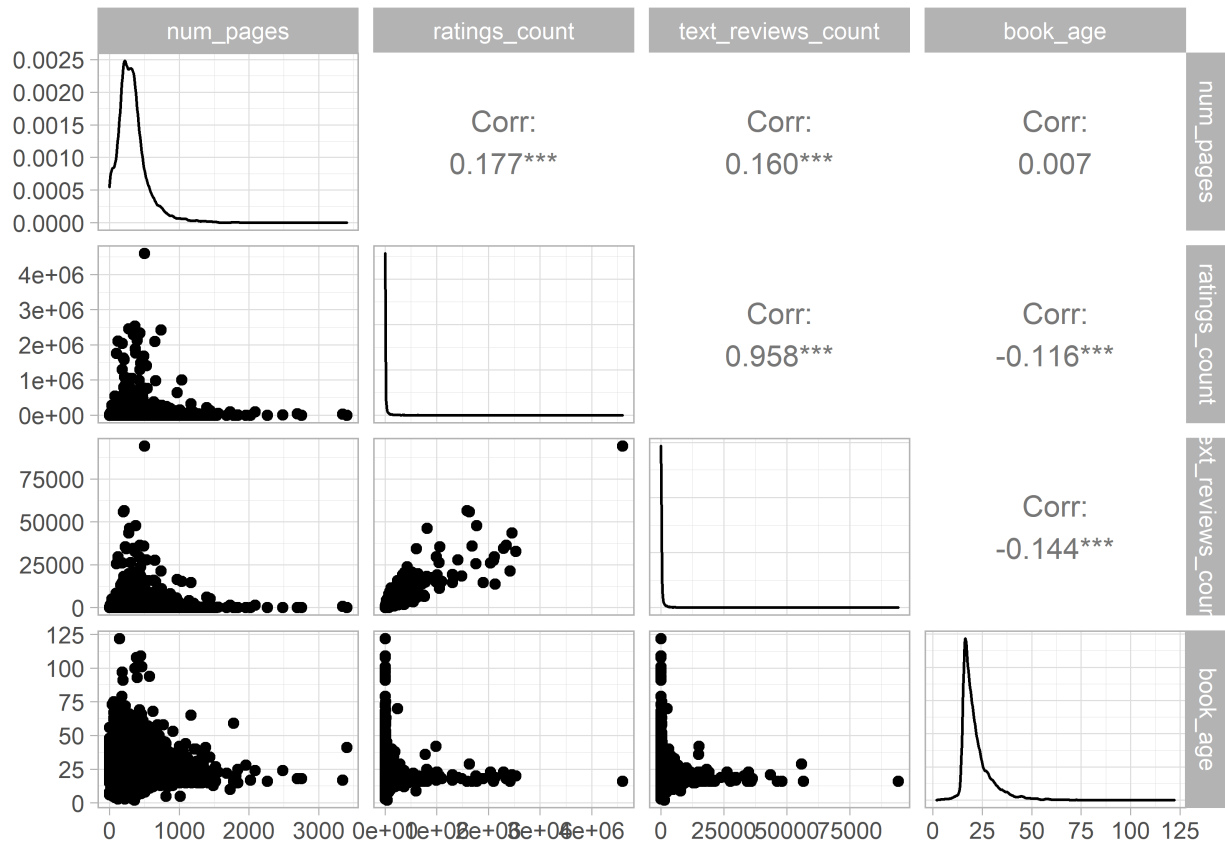
```

- Verificando a correlação das variáveis quantitativas

```

livros_treino %>%
  select(where(is.numeric)) %>%
  ggpairs(upper = list(continuous = wrap("cor", method = "spearman")))

```



Dado que identificamos alta correlação entre as variáveis `text_reviews_count` e `rating_count` a variável `text` será removida pois não necessariamente todo mundo que dá uma nota de avaliação também deixa uma avaliação escrita, o que inclusive explica a forte correlação entre essas variáveis, pois certamente todos que deixaram avaliação escrita também deixaram nota, no entanto, consideramos essa medida importante para avaliar se o livro é ótimo ou ruim, supondo que quando um livro for uma dessas duas opções as pessoas façam mais questão de comentar.

Sendo assim, criaremos uma variável proporção:

```
livros_treino <- livros_treino %>%
  mutate(prop_text_reviews = text_reviews_count / ratings_count) %>%
  select(-text_reviews_count)

cor(livros_treino$prop_text_reviews, livros_treino$ratings_count,
     use = "complete", method = "spearman")
```

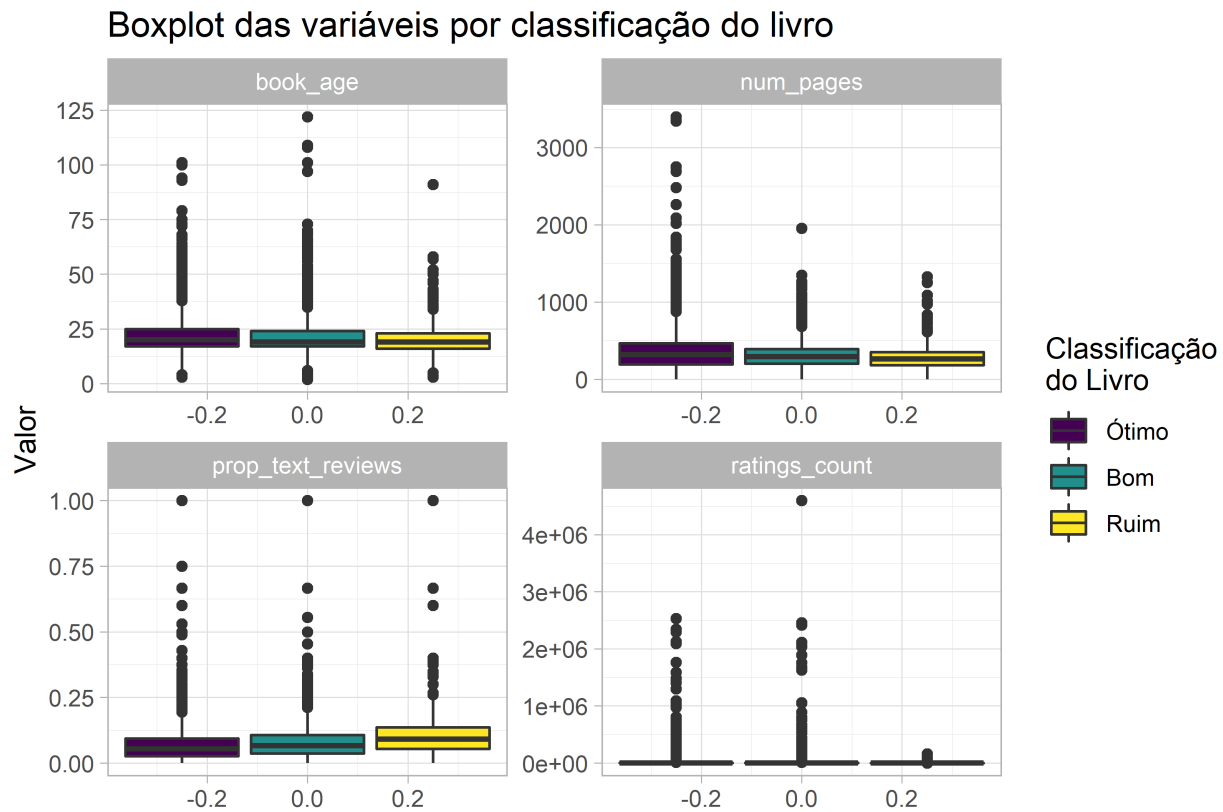
```
## [1] -0.3605444
```

Com essa nova variável tivemos uma baixa correlação, assim evitamos a multicolineariedade.

- Análise das variáveis

```
livros_treino %>%
  select(where(is.numeric), book_rating) %>%
  pivot_longer(-book_rating) %>%
```

```
ggplot(.,aes(fill = book_rating)) +
  geom_boxplot(aes(y=value)) +
  facet_wrap(~ name, scales = "free") +
  labs(x="",
       y="Valor",
       fill = "Classificação\ndo Livro",
       title = "Boxplot das variáveis por classificação do livro")+
  scale_fill_viridis_d()
```

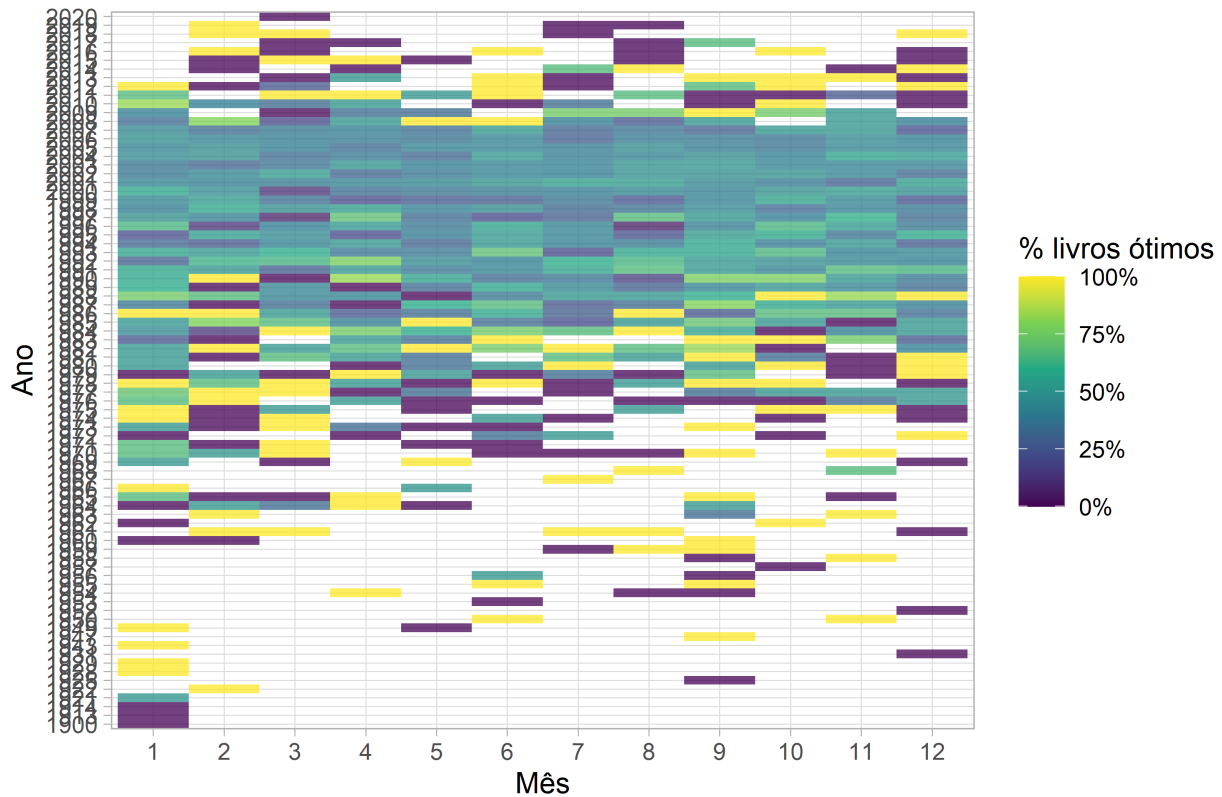


Pelos box-plots é notado que as distribuições das classificações de acordo com as variáveis há poucas diferenças entre si.

```
grafico_otimos_mes_ano <- livros_treino %>%
  mutate(book_rating = book_rating == "Ótimo") %>%
  group_by(
    mes = month_publication,
    ano = year_publication
  ) %>%
  summarise(book_rating = mean(book_rating)) %>%
  ggplot(aes(mes,ano, fill = book_rating)) +
  geom_tile(alpha = .75) +
  scale_fill_viridis_c(labels = scales::percent) +
  labs(fill = "% livros ótimos", x="Mês", y="Ano",
       title = "Composição dos livros avaliados como: ÓTIMO")+
  theme(legend.position = "right");grafico_otimos_mes_ano
```



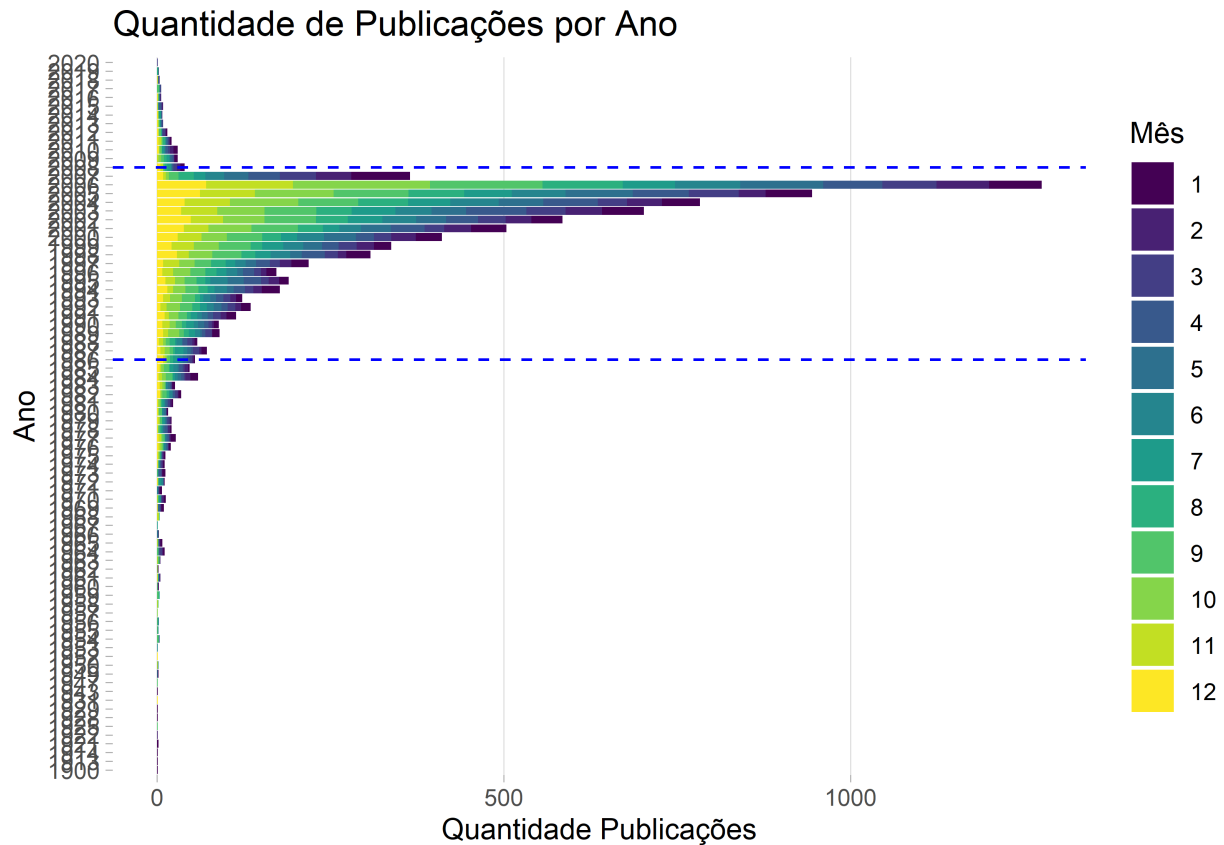
## Composição dos livros avaliados como: ÓTIMO



É observado que há uma maior avaliação de livros a partir dos anos 80 até em torno do ano de 2012. Nesses anos tiveram muitas avaliações de livros e a porcentagem de avaliação para ótimo está em torno de 25% a 75% em sua maioria.

Verificando a distribuição de livros publicados ao longo dos anos temos:

```
livros_treino %>%
  group_by(
    mes = month_publication,
    ano = year_publication
  ) %>%
  count() %>%
  ggplot(aes(n,ano, fill=mes))+
  geom_col()+
  geom_hline(yintercept = "1986", color = "blue", lty=2)+
  geom_hline(yintercept = "2008", color = "blue", lty=2)+
  theme(panel.border = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor = element_blank())+
  labs(x = "Quantidade Publicações",
       y = "Ano",
       fill = "Mês",
       title = "Quantidade de Publicações por Ano")+
  scale_fill_viridis_d()
```



O aumento das publicações dos livros foram bem proporcionais entre 1986 e 2007, não é perceptível algum mês que se destaque entre os demais. Inclusive até mesmo no ano 2008 em que teve uma redução, ainda assim também foi proporcional entre os meses.

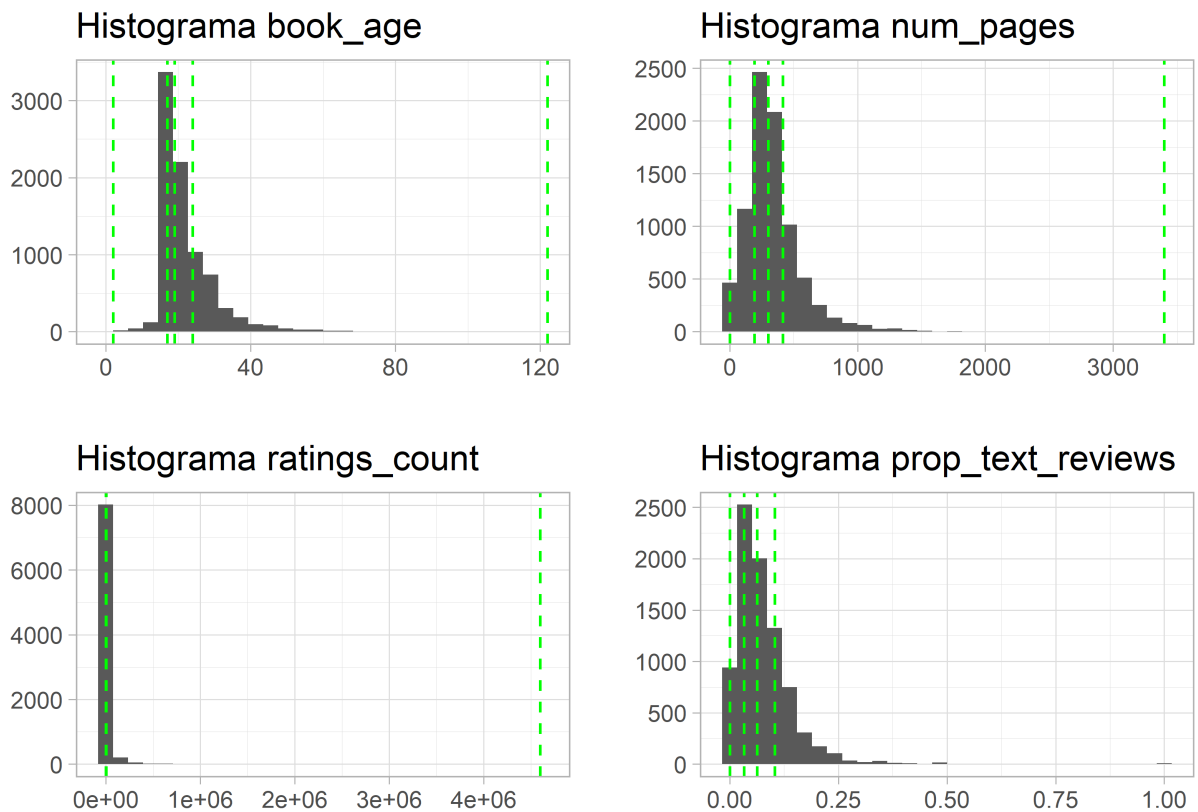
Como o boxplot apresentou muitos outliers e percebemos uma concentração nos dados iremos realizar uns filtros para melhorar a modelagem:

```
gridExtra::grid.arrange(ncol=2,
                          livros_treino %>%
    ggplot(aes(x=book_age)) +
    geom_histogram(bins=30)+
    geom_vline(xintercept = quantile(livros_treino$book_age),
               color="green", lty=2)+
    labs(title = "Histograma book_age",
         x="",
         y=""),
,
    livros_treino %>%
    ggplot(aes(x=num_pages)) +
    geom_histogram(bins=30)+
    geom_vline(xintercept = quantile(livros_treino$num_pages),
               color="green", lty=2)+
    labs(title = "Histograma num_pages",
         x="",
         y=""),
,
    livros_treino %>%
```

```

ggplot(aes(x=ratings_count)) +
  geom_histogram(bins=30)+
  geom_vline(xintercept = quantile(livros_treino$ratings_count),
            color="green", lty=2)+
  labs(title = "Histograma ratings_count",
        x="",
        y="")
',
livros_treino %>%
  ggplot(aes(x=prop_text_reviews)) +
  geom_histogram(bins=30)+
  geom_vline(xintercept = quantile(livros_treino$prop_text_reviews, na.rm = TRUE),
            color="green", lty=2)+
  labs(title = "Histograma prop_text_reviews",
        x="",
        y="")
)

```



Com os histogramas das variáveis numéricas percebemos que a maioria tem uma assimetria a direita, então com isso decidimos aplicar filtros nessas variáveis para diminuir os outliers dos nossos dados. A única variável em que o filtro não será aplicado por não ter a assimetria será a **ratings\_count**.

E agora o novo boxplot com os filtros aplicados:

```

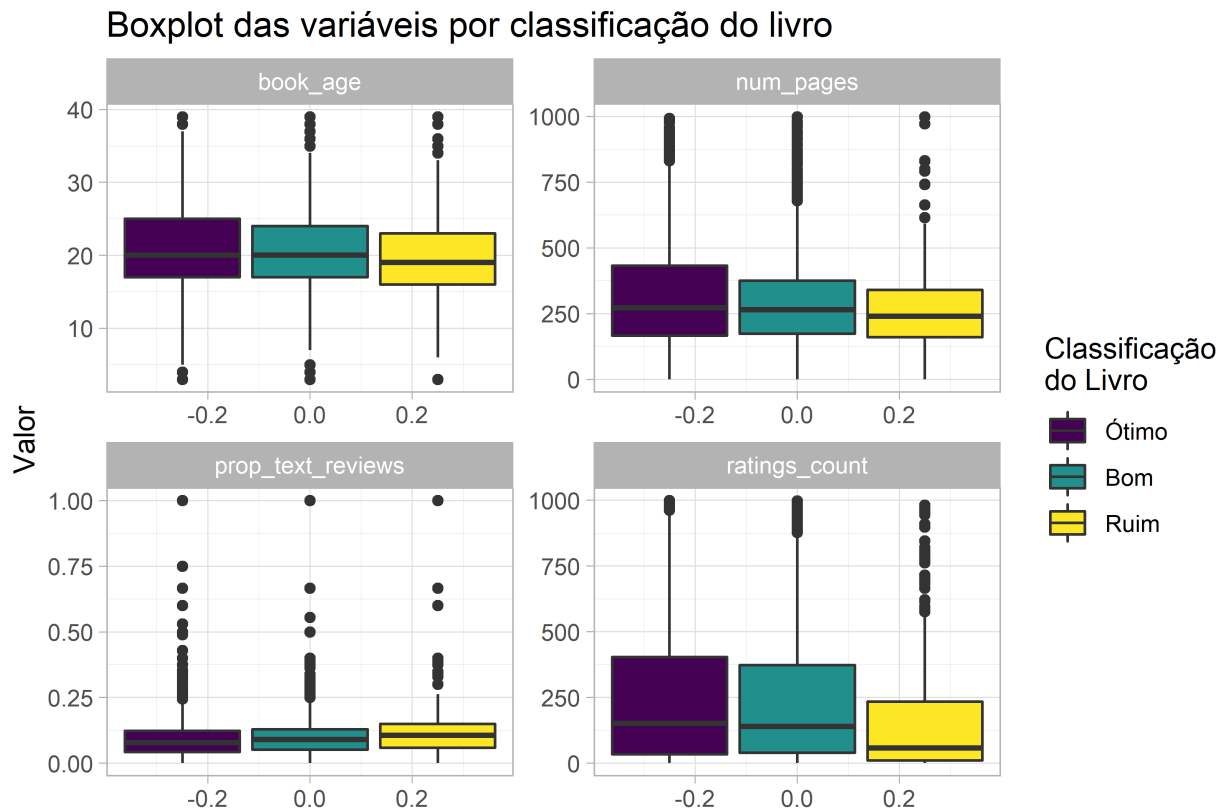
livros_treino %>%
  filter(book_age<40) %>%

```

```

filter(num_pages<1000) %>%
filter(ratings_count<1000) %>%
select(where(is.numeric),book_rating) %>%
pivot_longer(-book_rating) %>%
ggplot(.,aes(fill = book_rating)) +
geom_boxplot(aes(y=value)) +
facet_wrap(~ name, scales = "free") +
labs(x="",
      y="Valor",
      fill = "Classificação\ndo Livro",
      title = "Boxplot das variáveis por classificação do livro")+
scale_fill_viridis_d()

```



Com as mudanças feitas, houve uma diferença notável nas distribuições da idade do livro, número de páginas e avaliações, as amplitudes e variações em comparação aos box-plots anteriores. As contagens de avaliações, na classificação ótimas a amplitude e a variação é maior que as demais, se assemelha a classificação “Bom”, “Ruim” é a que possui mais outliers mas sua amplitude e variação é a menor.

## Modelagem

Considerando as alterações sofridas no conjunto de dados após a análise exploratória, será necessário carregar novamente o conjunto de dados, bem como gerar novo conjunto de treino e teste:

```

livros <- read.csv("./Conjunto de Dados/books_t.csv",
                  encoding = "UTF-8") %>%
  mutate(publication_date=as.Date(publication_date),
         prop_text_reviews = text_reviews_count / ratings_count,
         prop_text_reviews = ifelse(prop_text_reviews %in% c(NaN,Inf), 0, prop_text_reviews),
         book_rating=factor(book_rating,
                           levels = c("Ótimo","Bom","Ruim"))) %>%
  select(-month_publication, -year_publication, -text_reviews_count) %>%
  filter(book_age<40) %>%
  filter(num_pages<1000) %>%
  filter(ratings_count<1000)

set.seed(1904, kind = "Mersenne-Twister", normal.kind = "Inversion")

livros_split <- initial_split(livros, prop = .75, strata = book_rating)

livros_treino <- training(livros_split)

livros_teste <- testing(livros_split)

####Criando Métricas####

(livros_metricas <- metric_set(accuracy, roc_auc, mn_log_loss))

## # A tibble: 3 x 3
##   metric      class      direction
##   <chr>      <chr>      <chr>
## 1 accuracy   class_metric maximize
## 2 roc_auc    prob_metric  maximize
## 3 mn_log_loss prob_metric  minimize

####Criando Folds####

set.seed(1989)
(livros_folds <- vfold_cv(livros_treino, strata = book_rating, v=10))

```

## Pre-processamento dos Dados

```

livros_rec <- recipe(book_rating ~ ., data = livros_treino) %>%
  theme::step_downsample(book_rating) %>%
  step_date(publication_date, features = c("month"),
            keep_original_cols = FALSE) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  step_zv(all_numeric_predictors()) %>%
  #step_pca(all_predictors(), threshold = .80) %>%
  prep()

head(prep(livros_rec) %>%
     bake(new_data =NULL))

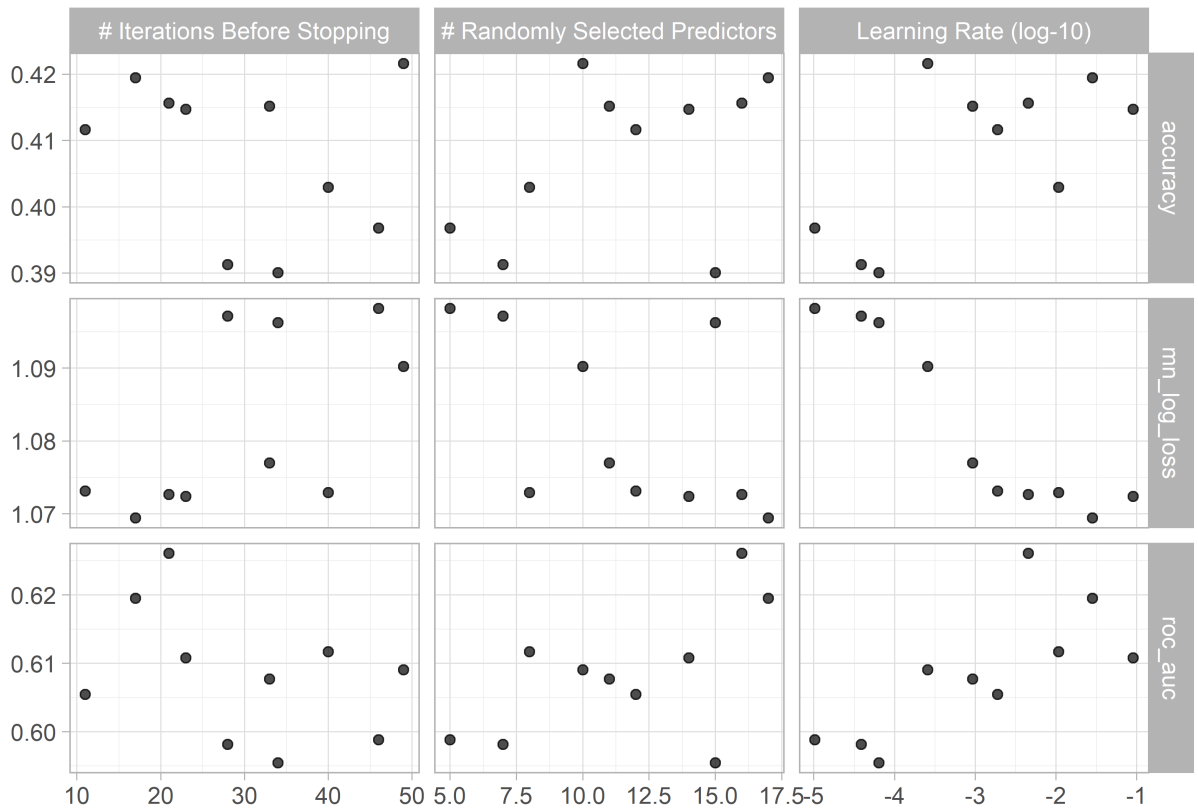
```

## Grid de Procura, Tune e Parada antecipada

```
stopping_spec <-  
  boost_tree(  
    trees = 500,  
    mtry = tune(),  
    learn_rate = tune(),  
    stop_iter = tune()  
  ) %>%  
  set_engine("xgboost", validation = 0.2) %>%  
  set_mode("classification")  
  
stopping_grid <-  
  grid_latin_hypercube(  
    mtry(range = c(5L, 18L)),  
    learn_rate(range = c(-5, -1)),  
    stop_iter(range = c(10L, 50L)),  
    size = 10  
  )  
  
early_stop_wf <- workflow(livros_rec, stopping_spec)  
  
doParallel::registerDoParallel()  
set.seed(2022)  
stopping_rs <- tune_grid(  
  early_stop_wf,  
  livros_folds,  
  grid = stopping_grid,  
  metrics = livros_metricas  
)
```

## Avaliação do Modelo

```
autoplot(stopping_rs)
```



```
show_best(stopping_rs, metric = "mn_log_loss")
```

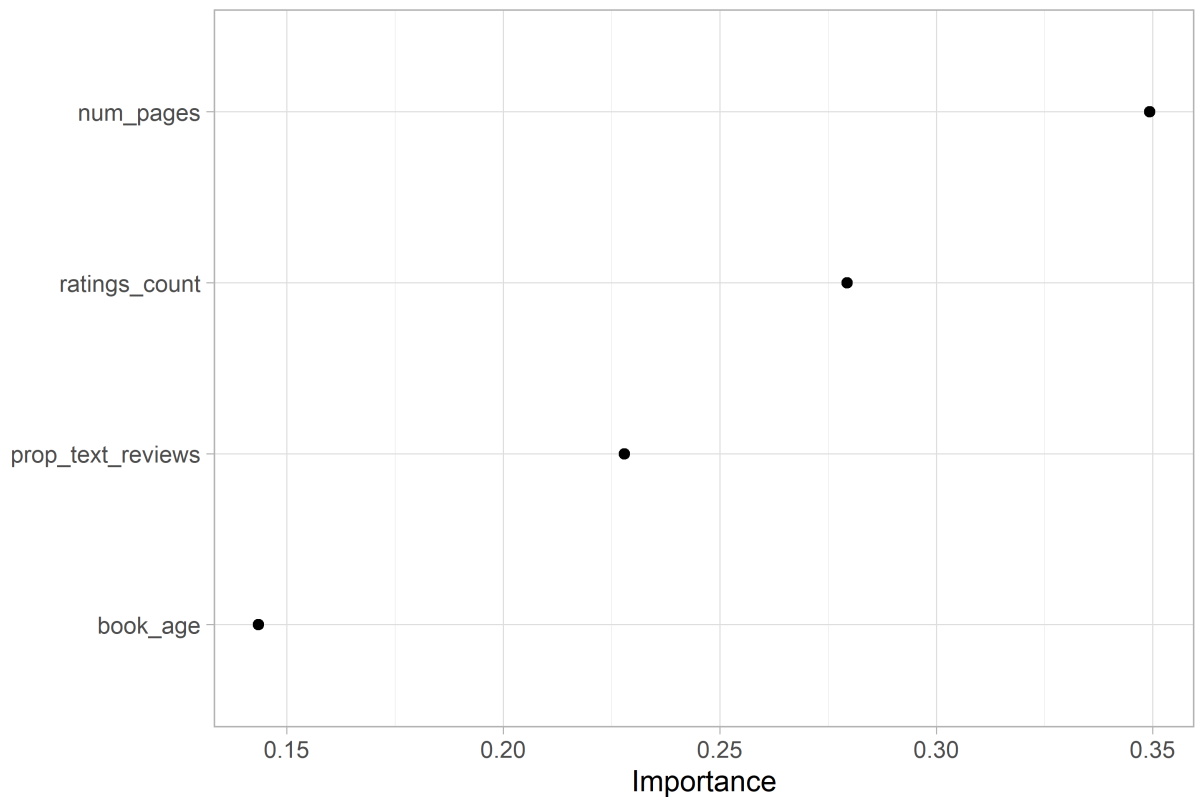
```
stopping_fit <- early_stop_wf %>%
  finalize_workflow(select_best(stopping_rs, "mn_log_loss")) %>%
  last_fit(livros_split)

stopping_fit
```

```
collect_metrics(stopping_fit)
```

```
extract_workflow(stopping_fit) %>%
  extract_fit_parsnip() %>%
  vip(num_features = 15, geom = "point")+
  ggtitle("Variáveis mais importantes no modelo")
```

### Variáveis mais importantes no modelo



As variáveis mais importantes para o modelo são os número de páginas, contagem de avaliações, comentários de textos e a idade do livro.

O que faz sentido pois a junção dessas variáveis para fazer uma classificação, É difícil ter muitos livros com muitas páginas, conseguir seguir uma linha de raciocínio e uma trama na qual prenda o leitor. Além disso, quanto mais páginas provavelmente mais caro será o livro.

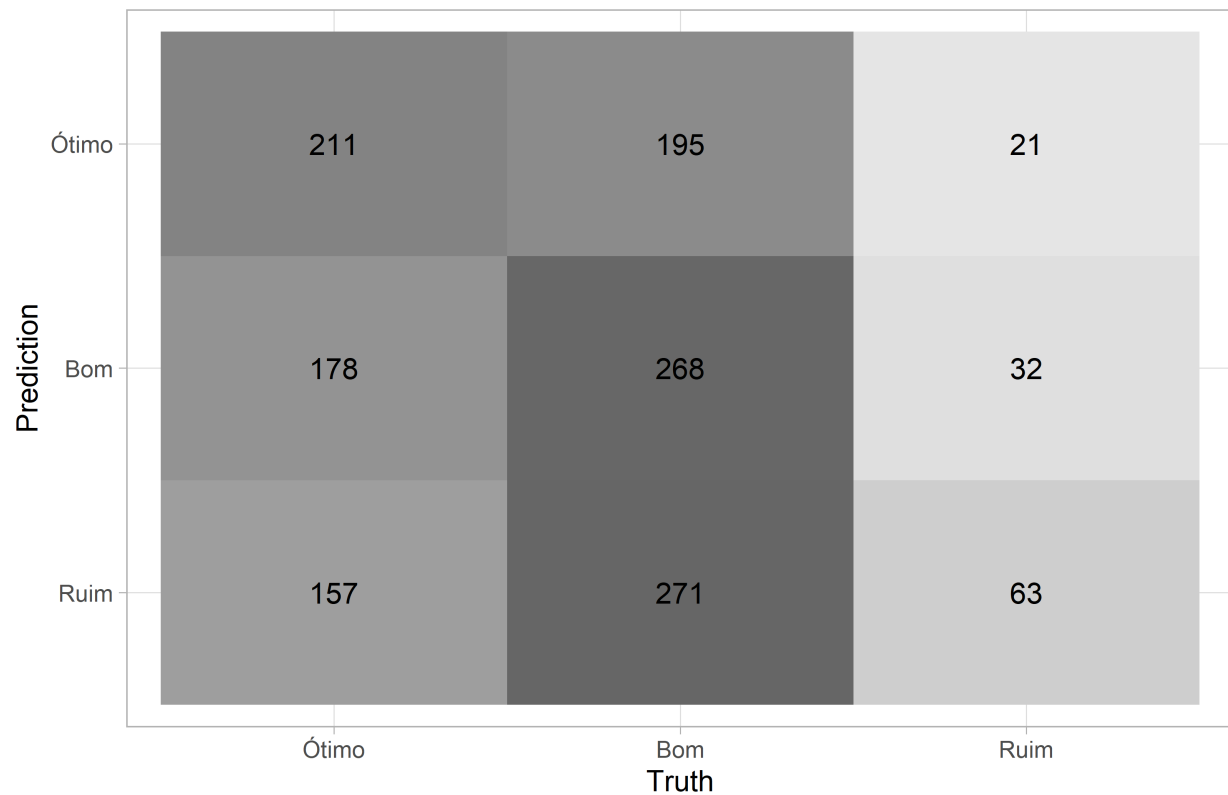
Principalmente, atualmente, a questão de um livro está sendo muito avaliado, muito divulgado nas redes, faz com que mais pessoas queiram consumir eles, tanto pela curiosidade de saber por que ele é tão bem avaliado e discutido. Em relação a idade

A idade do livro é um fator interessante, há os livros que se tornam clássicos, os que são deixados de lado e os que é possível fazer sucesso mesmo com um certo tempo de publicação.

```
collect_predictions(stopping_fit) %>%  
  conf_mat(book_rating, .pred_class) %>%  
  autoplot(type = "heatmap")+  
  ggtitle("Mapa de Calor das Predições")
```



Mapa de Calor das Predições



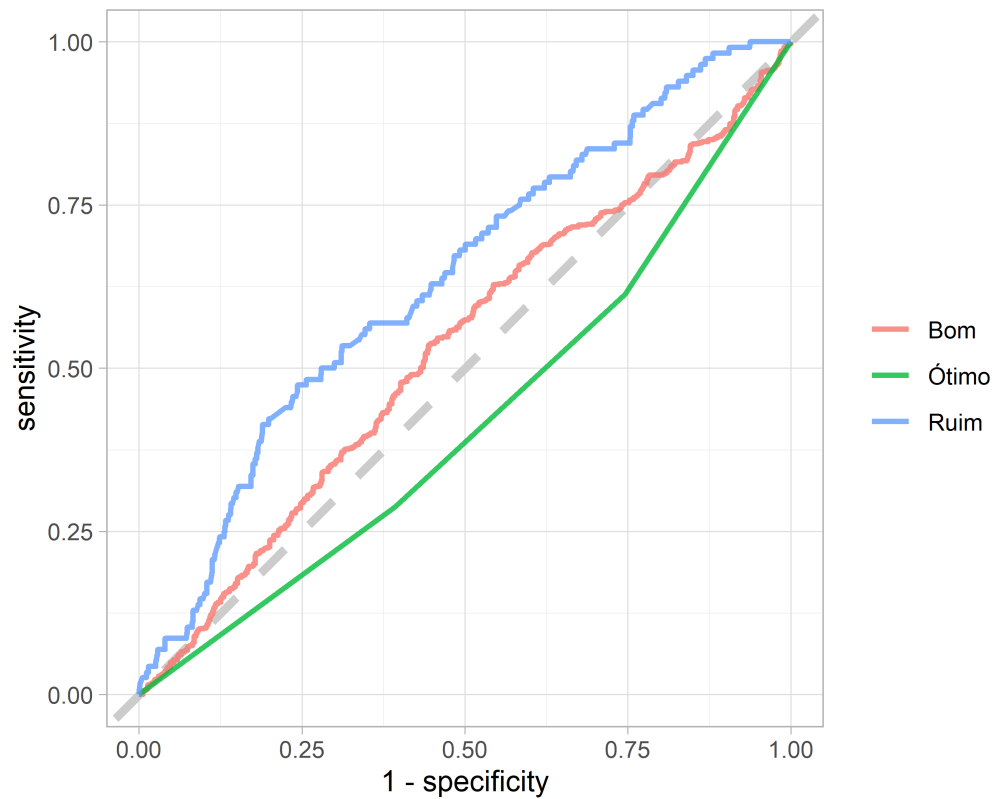
O modelo não foi o melhor, principalmente para avaliar “Ótimo” e “Bom”.

Dos livros classificados como “Ótimo”, o modelo classificou 189 como “Bom” e 146 como “Ruim”.

Dos livros classificados como “Bom”, o modelo classificou 200 como “Ótimo” e 209 como “Ruim”.

```
collect_predictions(stopping_fit, summarize = FALSE) %>%
  roc_curve(book_rating, .pred_class:.pred_Ruim) %>%
  ggplot(aes(1 - specificity, sensitivity, color = .level)) +
  geom_abline(lty = 2, color = "gray80", size = 1.5) +
  geom_path(alpha = 0.8, size = 1) +
  coord_equal() +
  labs(color = NULL,
       title = "Curva ROC Modelo Final")
```

Curva ROC Modelo Final



```
## Verdadeiro positivo
collect_predictions(stopping_fit, summarize = TRUE) %>%
  sens(book_rating, .pred_class)
```

```
## Verdadeiro negativo
collect_predictions(stopping_fit, summarize = TRUE) %>%
  spec(book_rating, .pred_class)
```