Modelagem XGBOOST para a classificação de avaliações de livros

Ana Luzielma Campos
Jaylhane Nunes
Raianny Soares

07/02/2022

Introdução

Contextualização

Motivadas pelo interesse comum em leitura optamos por realizar a análise de um conjuntos de dados envolvendo livros.

O conjunto de dados selecionado possui **11.131 observações**, foi gerado por meio de **raspagem** de dados na **API** da plataforma **GoodReads** e disponibilizado por ***Soumik*** no site **Kagle**.

Nele é possível encontrar as 12 colunas seguintes:

bookID	average_rating	language_code	text_reviews_count
title	isbn	num_pages	publication_date
authors	isbn13	ratings_count	publisher

Chegamos a um consenso que diversos fatores influenciam na satisfação com a leitura e quisemos investigar se, com os dados disponíveis, seria possível obter um modelo que conseguisse predizer se o livro foi considerado: *ruim*, *bom* ou *ótimo* .

Uma Ideia Inicial

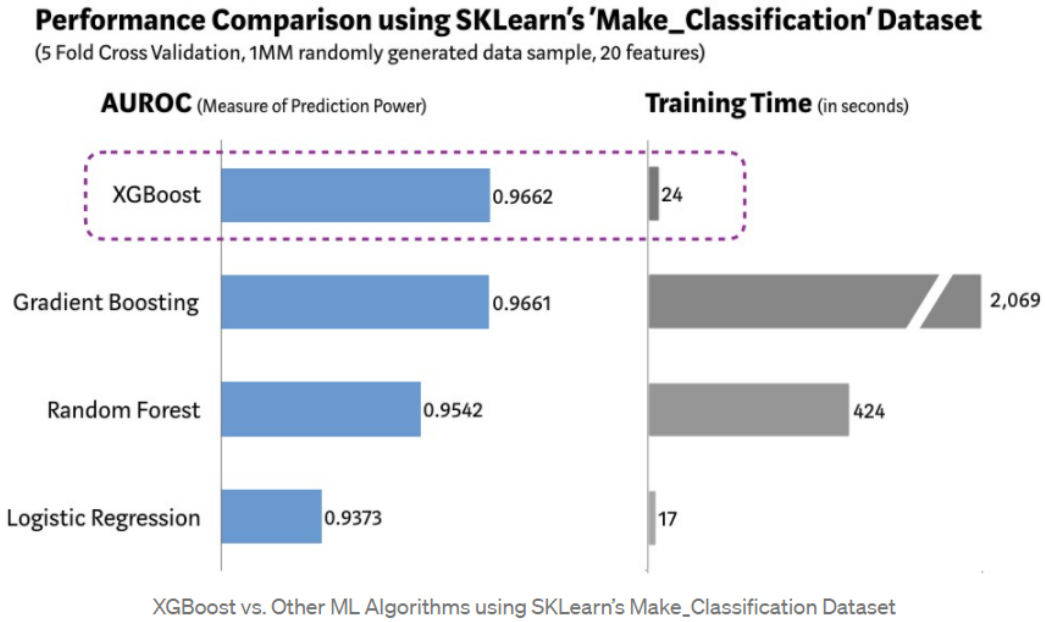


Figura 1: Vishal Morde, 2019 - XGBoost Algorithm: Long May She Reign!

Dentre as possibilidades percebemos que o **XGboost** tem um ótimo desempenho comparado a outros e que, apesar da nossa variável de avaliação ser uma variável contínua, um método de classificação poderia ser adequado, desde que encaixássemos intervalos em categorias.

A Inspiração Final

Julia Silge

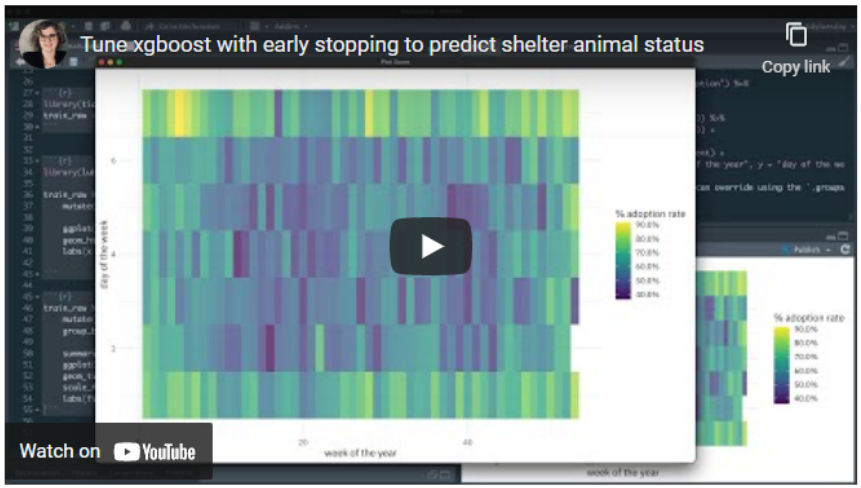
ABOUT BLOG

Tune xgboost models with early stopping to predict shelter animal status

By Julia Silge in rstats tidymodels

AUGUST 7, 2021

This is the latest in my series of screencasts demonstrating how to use the tidymodels packages, from just getting started to tuning more complex models. I participated in this week's episode of the SLICED playoffs, a competitive data science streaming show, where we competed to predict the status of shelter animals. 😺 I used xgboost's early stopping feature as I competed, so let's walk through how and when to try that out!



Indicamos

Julia Silge

DATA SCIENTIST & SOFTWARE ENGINEER

I'm a tool builder, author, international keynote speaker, and real-world practitioner focusing on data analysis and machine learning. I love making beautiful charts and communicating about technical topics with diverse audiences.

READ LATEST POSTS →

juliasilge.com

- *Agora, à nossa análise!*

Desenvolvimento

Engenharia de Dados

A análise exploratória consistiu em:

- Limpeza dos Dados
- Análise Descritiva

Dado os nossos objetivos percebemos que algumas colunas eram dispensáveis e outras poderiam ser transformadas, de forma que:

Excluídas	Transfomadas	Geradas
bookID	average_rating	book_rating[1]
title	language_code	
isbn	text_reviews_count	prop_text_reviews[2]
authors	publication_date	book_age
isbn13		
publisher		

Nosso **MAIOR** *desafio durante a análise e a modelagem esteve relacionado a essa fase de engenharia de dados*

Análise Exploratória

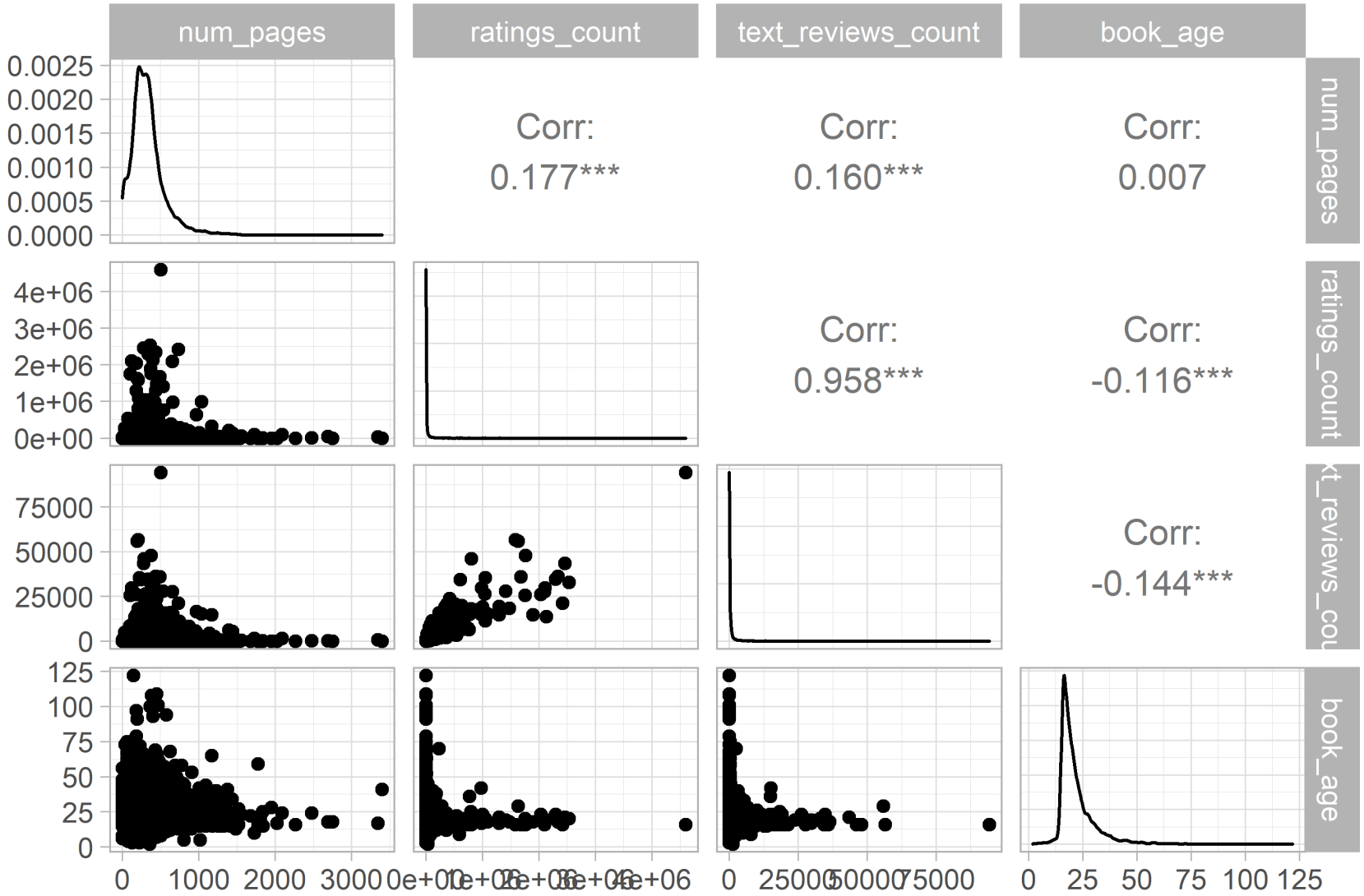
- Para iniciar separamos o conjunto em **treino** e **teste** baseando-nos em 75% das observações e balanceando-as com nossa variável resposta: `book_rating`.

```
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)
```

- Em seguida verificamos a dispersão e correlação entre as variáveis numéricas:

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

Análise Exploratória



Análise Exploratória

Com o resultado anterior percebemos que:

- Correlação forte entre `text_reviews_count` e `ratings_count`.
- Seria necessário descartar `text_reviews_count` devido riscos de multicolinearidade, mas como consideramos as avaliações escritas relevantes para as categorias *ruim* e *ótimo* usamos a proporção entre `text_reviews_count/ratings_count`.

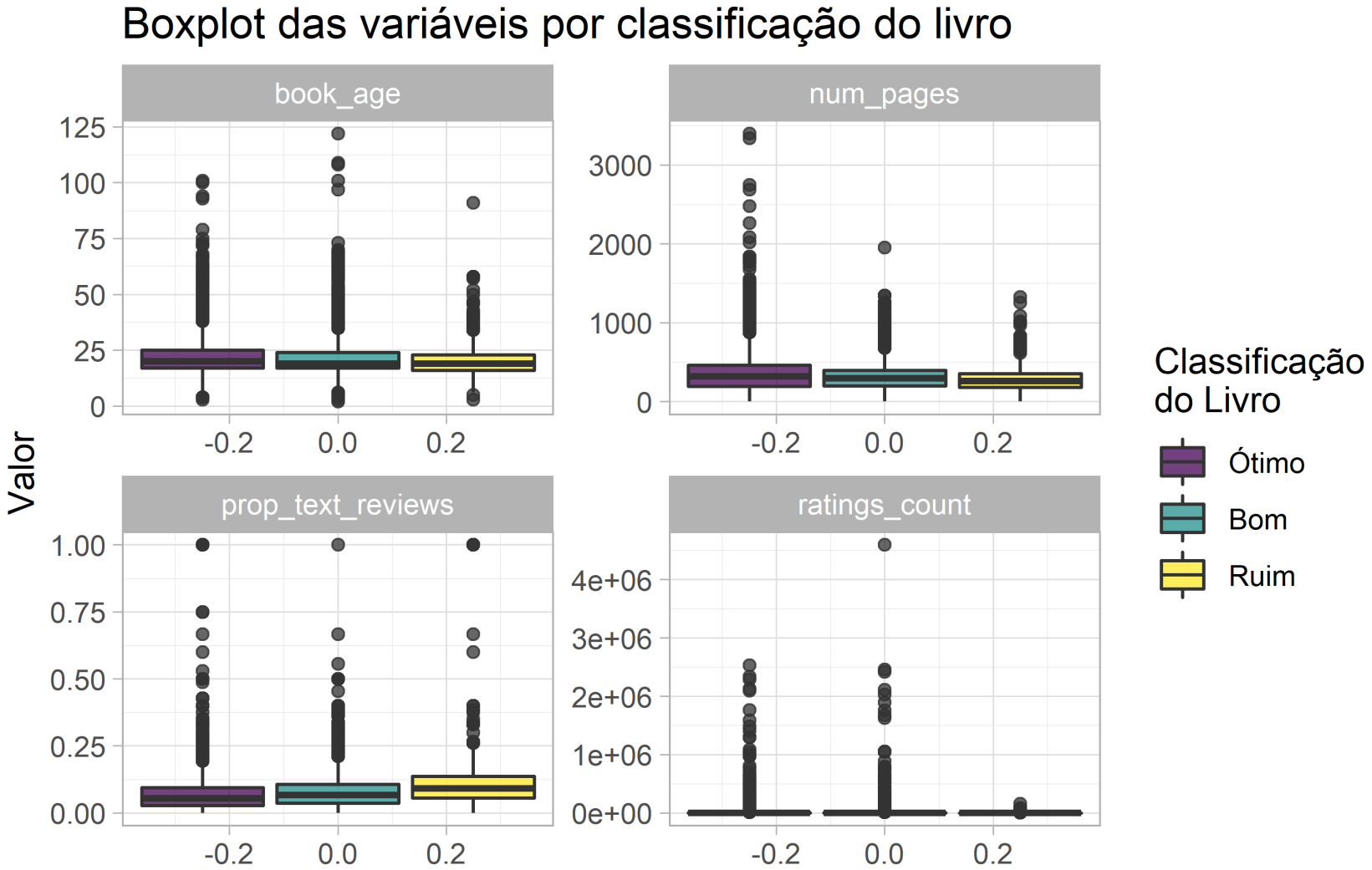
```
livros_treino <- livros_treino %>%
  mutate(prop_text_reviews = text_reviews_count / ratings_count) %>%
  select(-text_reviews_count)

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

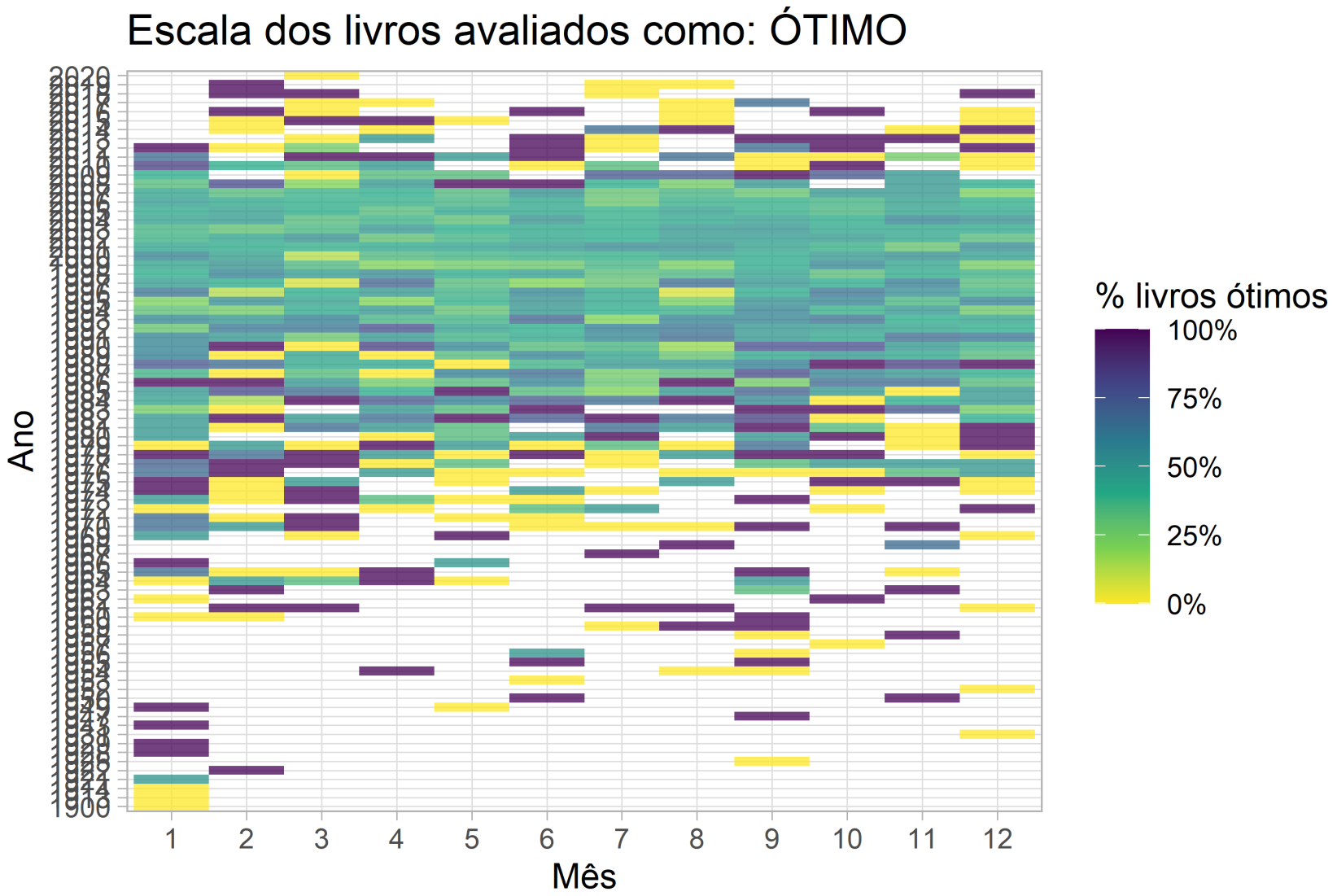
```
[1] -0.3605
```

- A correlação entre `prop_text_reviews` e `ratings_count` não indicou multicolinearidade, então prosseguimos com essa variável como preditora.

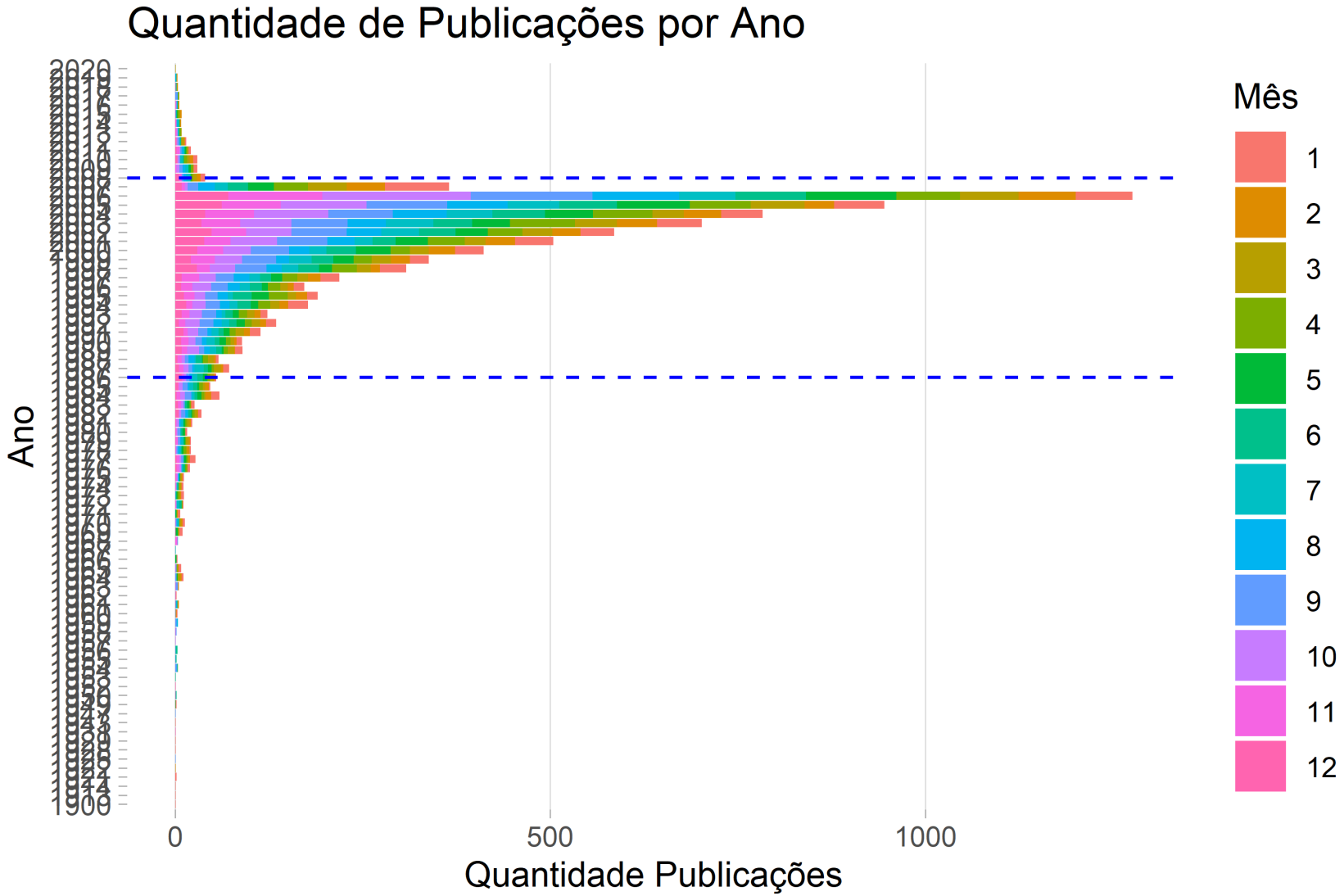
Continuando a Análise Exploratória e Descritiva



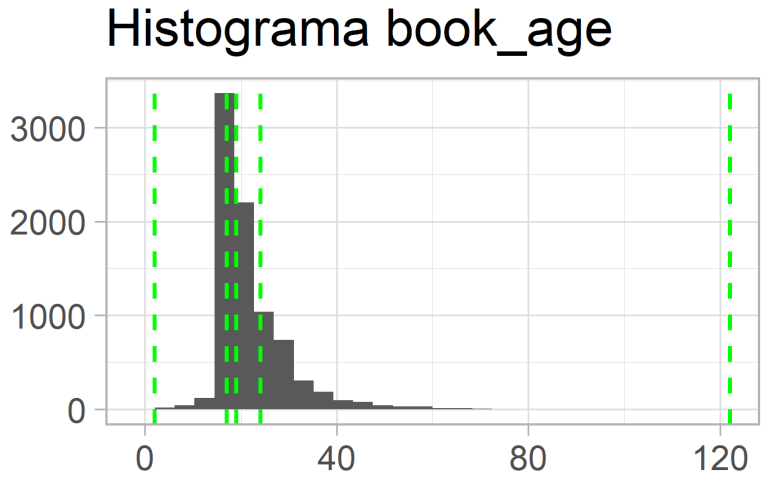
Conferindo aspectos que consideramos importantes

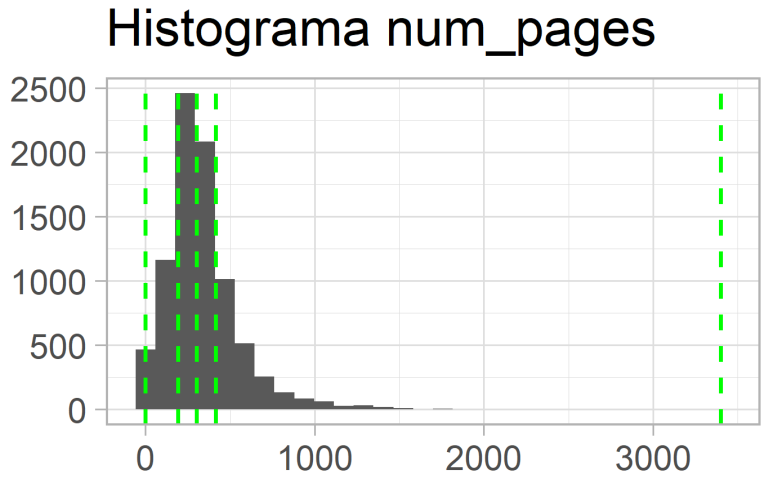


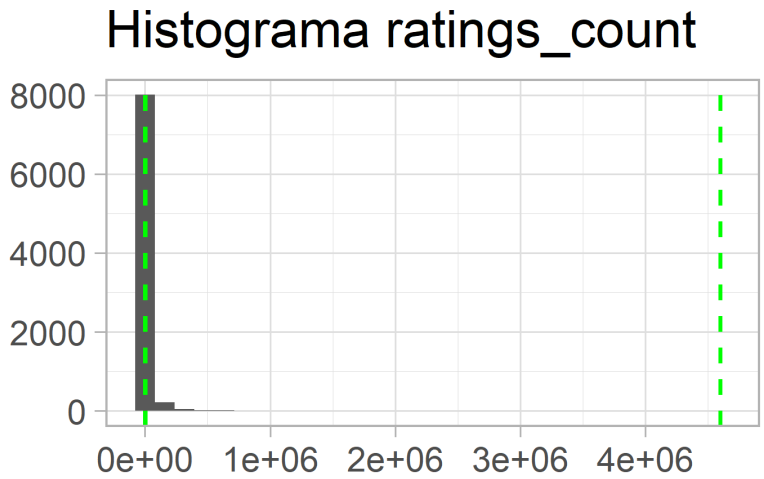
Mais gráficos =)

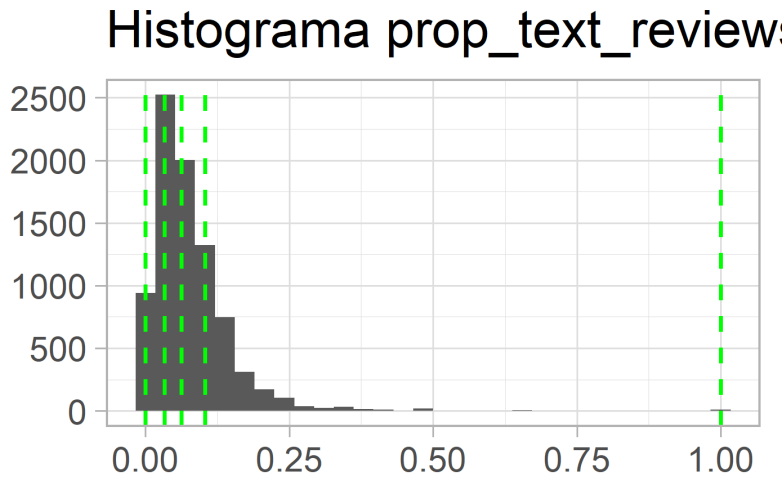


Definindo Filtros

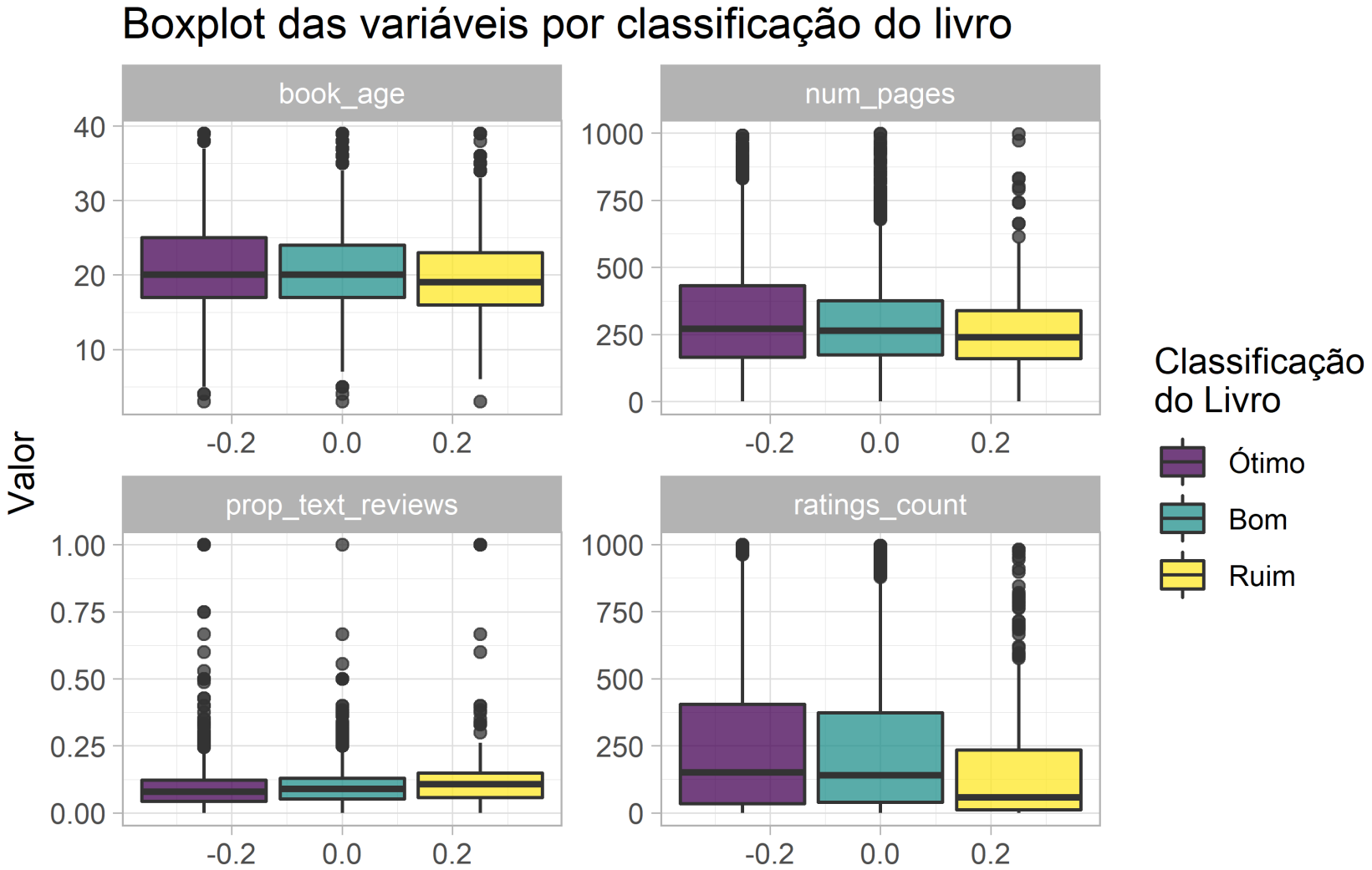








Novo boxplot com filtros aplicados



A modelagem

A modelagem

- Com as análises identificamos mudanças necessárias no conjunto de dados, demandando nova divisão em **treino** e **teste**. Os filtros aplicados foram:

Variável	Filtro
book_age	<40
num_pages	<1000
rating_count	<1000
prop_text_reviews	Nenhum

- Posteriormente criamos métricas e *folds* que serão utilizadas para *tunar* o módelo:

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

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

- E em seguida as demais etapas da modelagem:

Pré-Processamento de Dados

```
livros_rec <- recipe(book_rating ~ ., data = livros_treino) %>%
  themis::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_predictors()) %>%
  prep()
```

Grid de Procura e de Parada antecipada

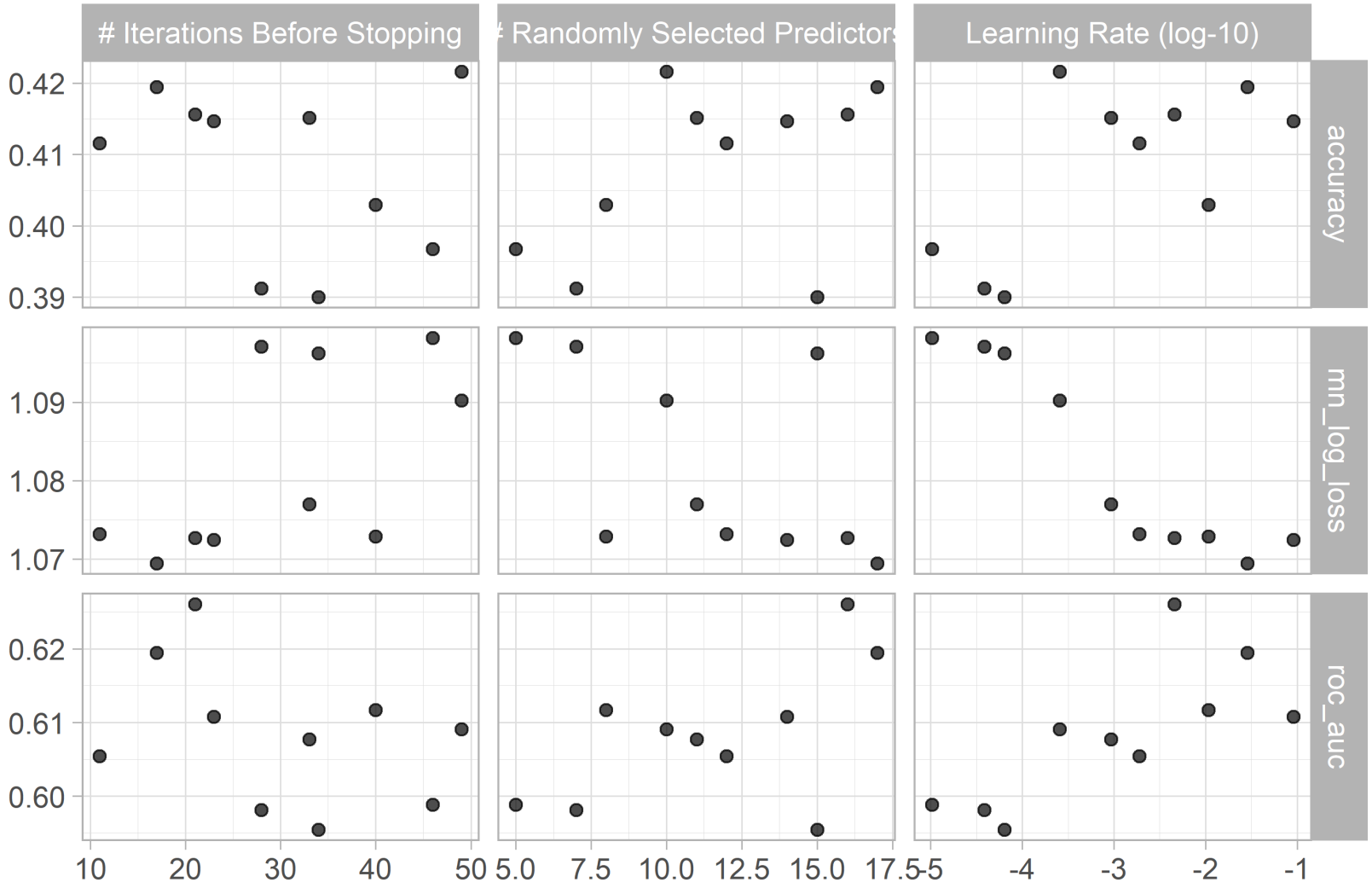
```
stopping_spec <-
  boost_tree(
    trees = 500,
    mtry = tune(),
    learn_rate = tune(),
    stop_iter = tune()
  ) %>%
  set_engine("xgboost", validation = 0.2) %>%
  set_mode("classification")
stopping_grid <-
  grid_latin_hypercube(
    mtry(range = c(5, 18)),
    learn_rate(range = c(-5, -1)),
    stop_iter(range = c(10, 50)),
    size = 10
  )
early_stop_wf <- workflow(livros_rec, stopping_spec)
```

Definido os grids de procura e parada, é hora de **TUNAR** o modelo!

```
doParallel::registerDoParallel()
set.seed(2022)
stopping_rs <- tune_grid(
  early_stop_wf,
  livros_folds,
  grid = stopping_grid,
  metrics = livros_metricas
)
```

Resultados

Interações e Taxa de Aprendizagem



Avaliação do modelo

- Avaliando o melhor resultado de acordo com **mn_log_loss**, pois é com ele que acompanhamos a capacidade de aprendizagem do modelo:

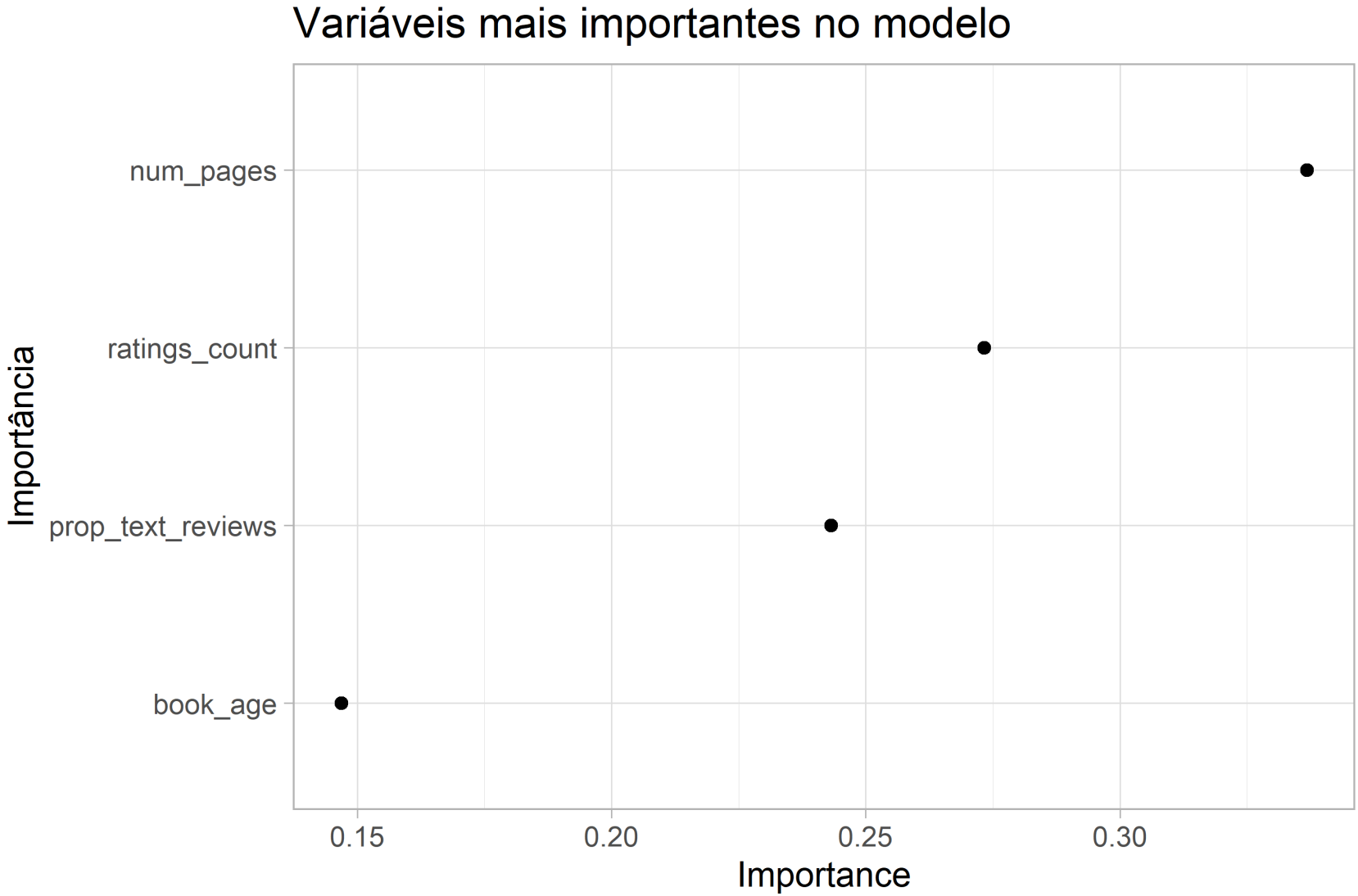
```
show_best(stopping_rs, metric = "mn_log_loss")
```

mtry	learn_rate	stop_iter	mean	n	std_err	.config
17	0.0286	17	1.069	10	0.0059	Preprocessor1_Model10
14	0.0910	23	1.072	10	0.0043	Preprocessor1_Model09
16	0.0046	21	1.073	10	0.0050	Preprocessor1_Model01
8	0.0108	40	1.073	10	0.0028	Preprocessor1_Model05
12	0.0019	11	1.073	10	0.0039	Preprocessor1_Model04

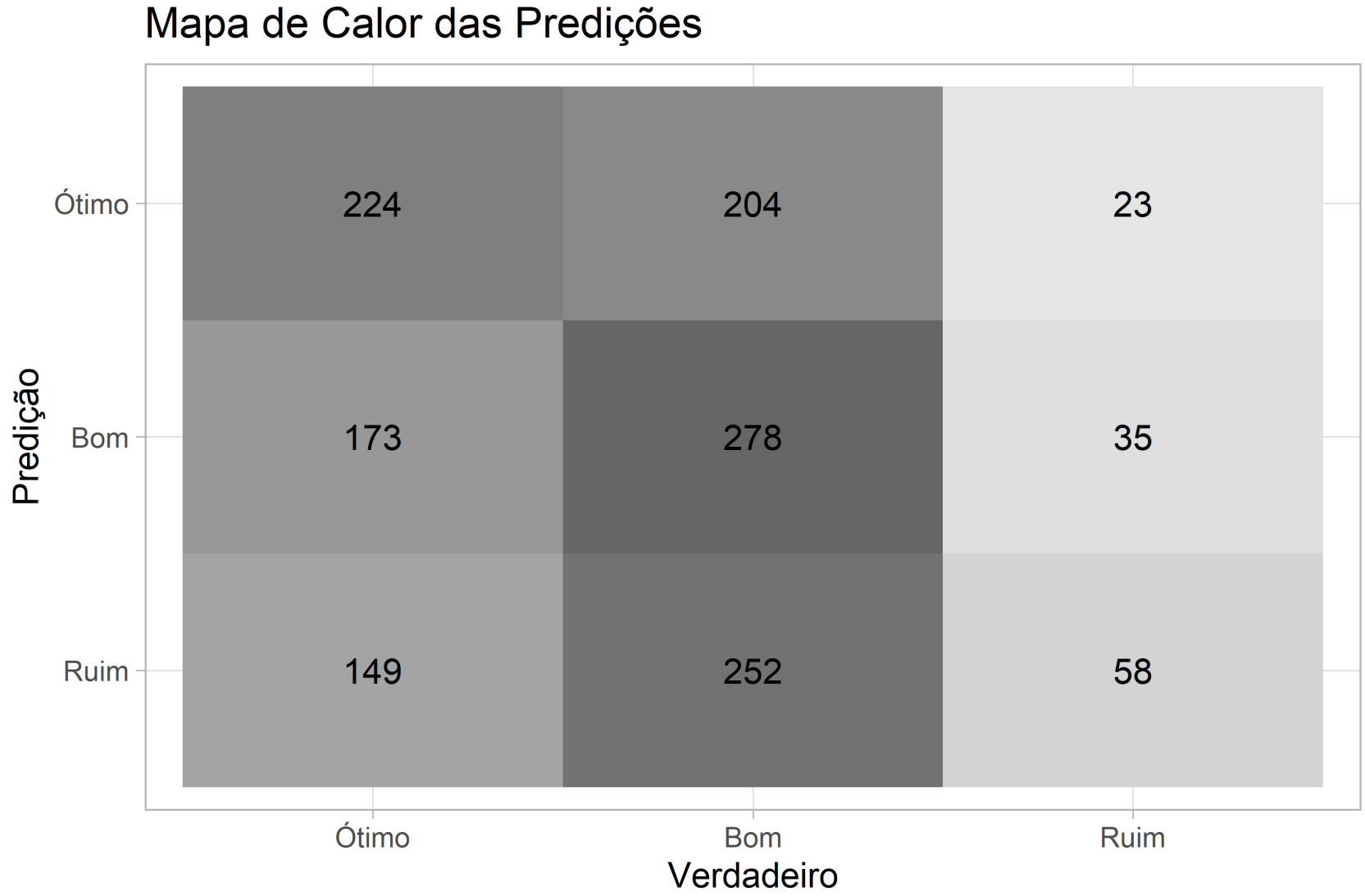
- Em seguida realizamos o modelo final e avaliamos as demais métricas:

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

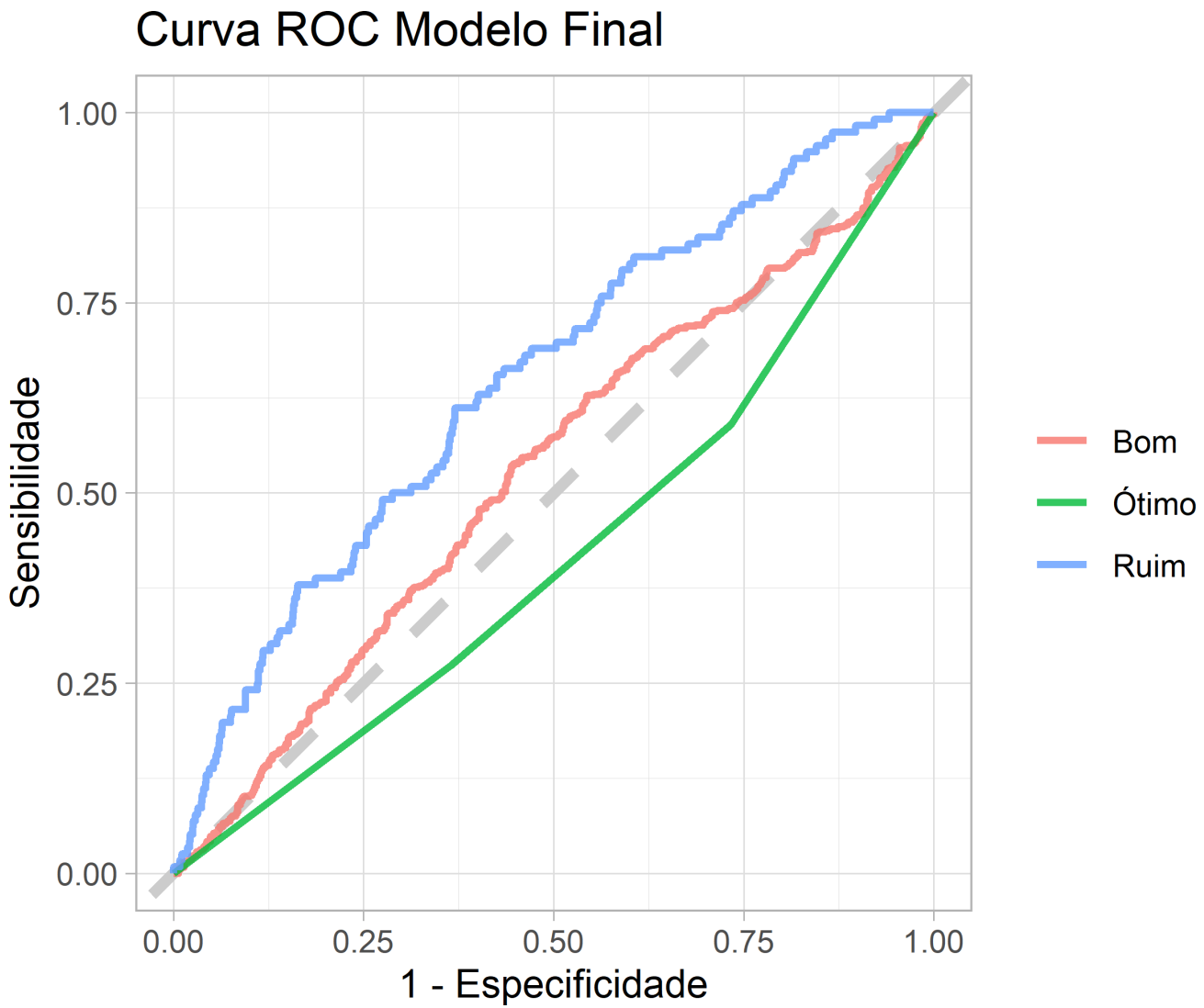
Variáveis VIP



Heatmap


Mapa de Calor das Predições
Predição
Ótimo
Bom
Ruim
224
204
23
173
278
35
149
252
58
Ótimo
Bom
Ruim
Verdadeiro

Avaliando a Curva ROC



Conclusão

Se o modelo disser que o livro é Ótimo, desconfie...

- Algumas métricas:

.metric	.estimator	.estimate
sens	macro	0.4297

.metric	.estimator	.estimate
spec	macro	0.7018

.level	mean_sens	mean_spec
Bom	0.5144	0.5152
Ótimo	0.5725	0.3805
Ruim	0.6383	0.5100

Mais algumas considerações...

- A parte da engenharia de dados, como já comentada, além de desafiadora realiza importante papel na qualidade do modelo
- Algumas mudanças poderiam melhorar o modelo, tais como:
  - uma melhora na limpeza dos dados, removendo ainda mais outliers;
  - melhor definição nos níveis e categorias das variáveis preditoras;
  - variáveis preditoras mais informativas e com mais distinção entre os níveis.
- Por fim, não consideramos satisfatório a capacidade preditora do modelo e concluímos que a melhor forma de saber se um livro é *ótimo*, *bom* ou *ruim*, é lendo-o. ;)

Referências:

- MORDE, Vishal. **XGBoost Algorithm: Long May She Reign!** - Abril, 2019. Publicado em *Towards Data Science*. Disponível em: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
- SILGE, Julia. **Tune xgboost models with early stopping to predict shelter animal status** - Agosto, 2021. Publicado em Julia Silge. Disponível em: https://juliasilge.com/blog/shelter-animals/
- R Core Team (2021). **R:*A language and environment for statistical computing*** . R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- SOUMIK . **Goodreads-books *comprehensive list of books listed in goodreads*** - Maio, 2019. Publicado em Kagle. Disponível em: https://www.kaggle.com/jealousleopard/goodreadsbooks