

# 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 conjunto 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 prever se o livro foi considerado: *ruim*, *bom* ou *ótimo*.

## UMA IDEIA INICIAL

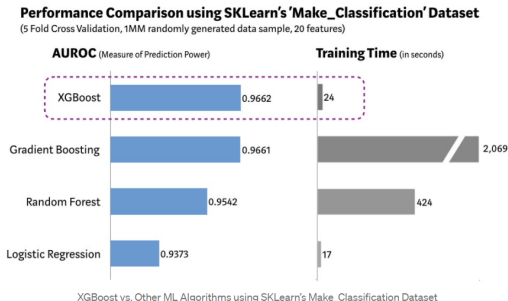


FIGURA 1: Vishal Morde, 2019 - XGBoost Algorithm: Long May She Reign!

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 [stats 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](http://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	Transformadas	Geradas
bookID	average_rating	book_rating <sup>1</sup>
title	language_code	
isbn	text_reviews_count	prop_text_reviews <sup>2</sup>
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

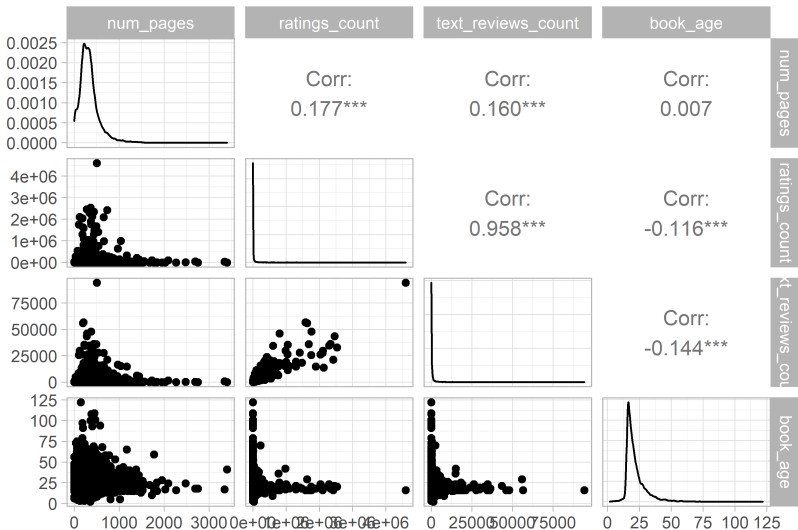
- ▶ Para iniciar separamos o conjunto em **treino** e **teste** baseando-nos em 75% das observações e balanceando-as com nossa variável resposta: `book_rating`.

```
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 multicolineariedade, 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)
```

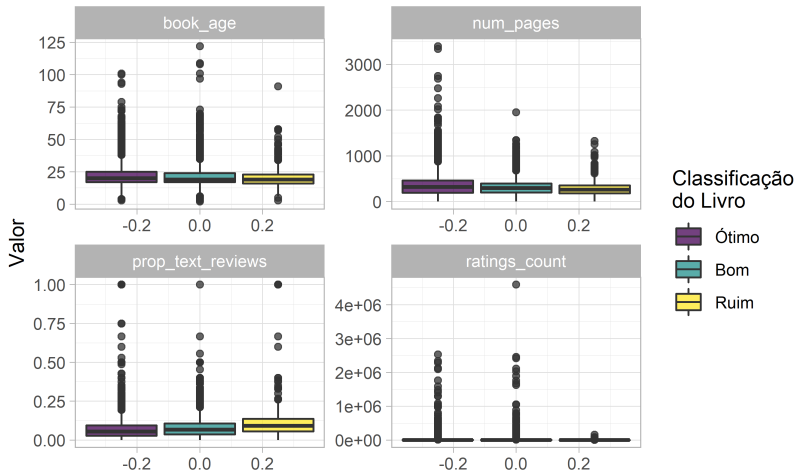
```
cor(livros_treino$prop_text_reviews, livros_treino$ratings_count,  
    use = "complete", method = "spearman")
```

```
[1] -0.3605
```

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

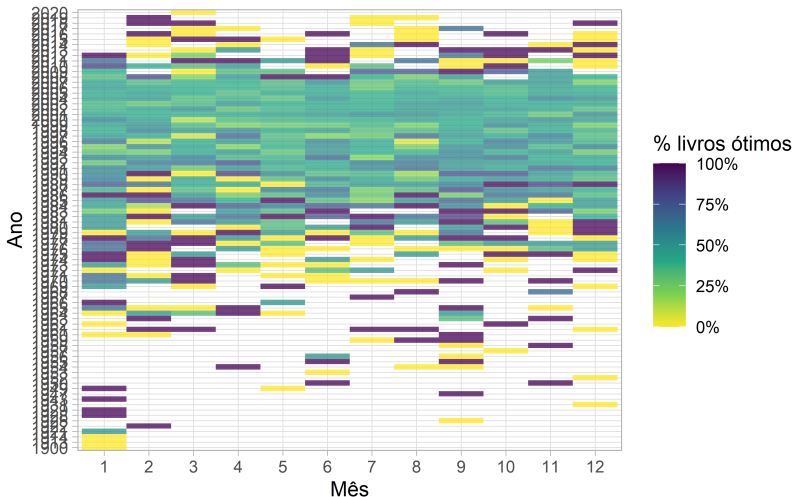
# CONTINUANDO A ANÁLISE EXPLORATÓRIA E DESCRITIVA

## Boxplot das variáveis por classificação do livro

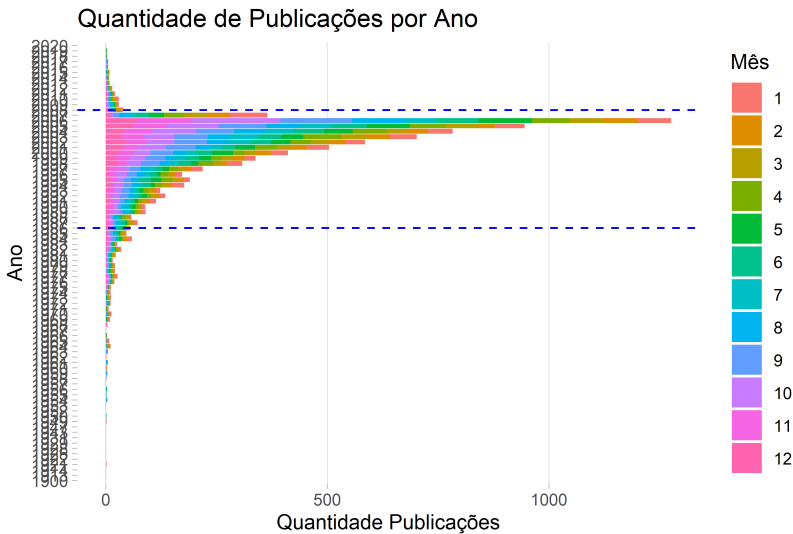


# CONFERINDO ASPECTOS QUE CONSIDERAMOS IMPORTANTES

## Escala dos livros avaliados como: ÓTIMO

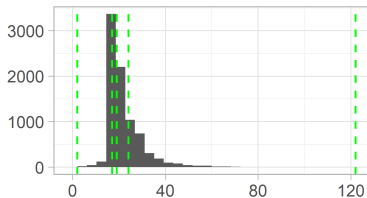


MAIS GRÁFICOS =)

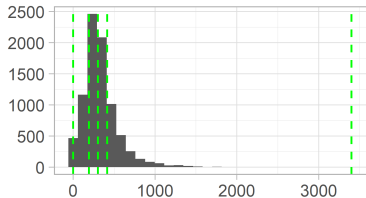


## DEFININDO FILTROS

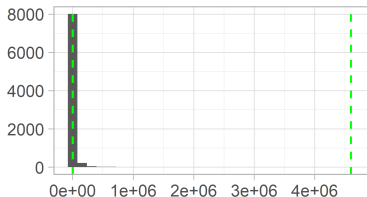
### Histograma book\_age



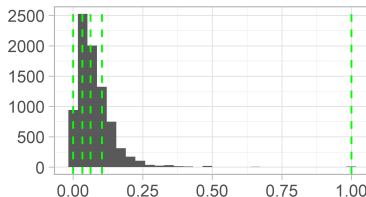
### Histograma num\_pages



### Histograma ratings\_count

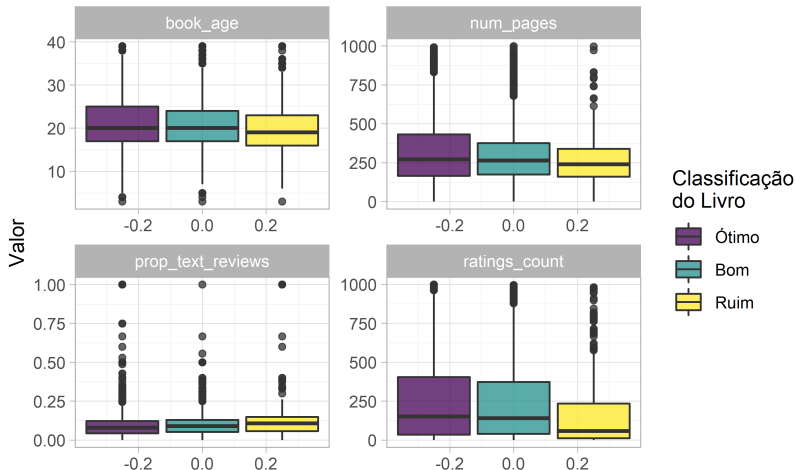


### Histograma prop\_text\_review



## NOVO BOXPLOT COM FILTROS APLICADOS

Boxplot das variáveis por classificação do livro





# 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 modelo:

```
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

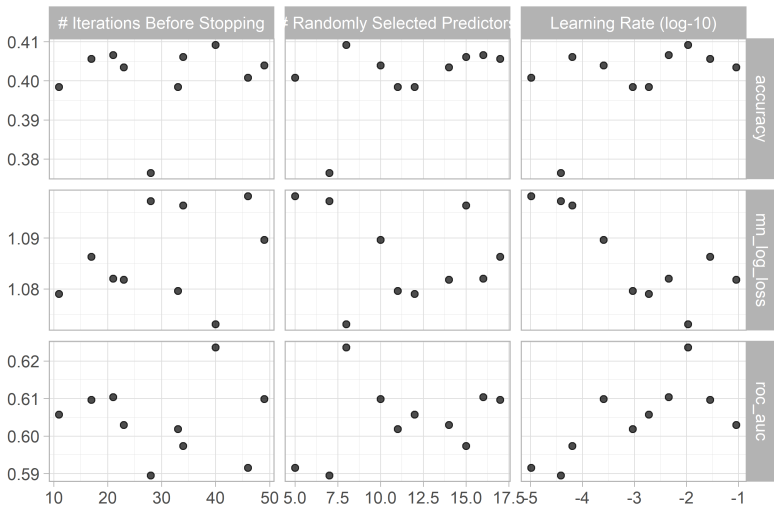
```
stopping_spec <-  
  boost_tree(  
    trees = 500,  
    mtry = tune(),  
    learn_rate = tune(),  
    stop_iter = tune()  
  ) %>%  
  set_engine("xgboost", validation = 0.2) %>%  
  set_mode("classification")  
stopping_grid <-  
  grid_latin_hypercube(  
    mtry(range = c(5, 18)),  
    learn_rate(range = 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
8	0.0108	40	1.073	10	0.0045	Preprocessor1_Model05
12	0.0019	11	1.079	10	0.0041	Preprocessor1_Model04
11	0.0009	33	1.080	10	0.0028	Preprocessor1_Model03
14	0.0910	23	1.082	10	0.0034	Preprocessor1_Model09
16	0.0046	21	1.082	10	0.0048	Preprocessor1_Model01

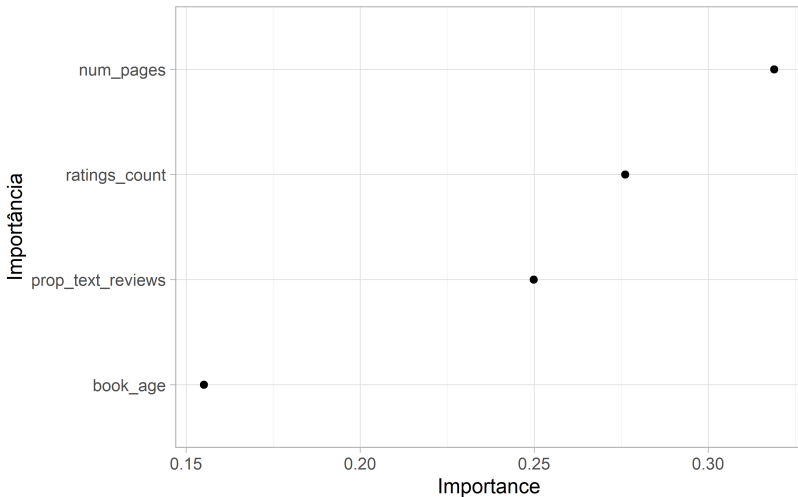
- ▶ Em seguida realizamos o modelo final e avaliamos as demais métricas:

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



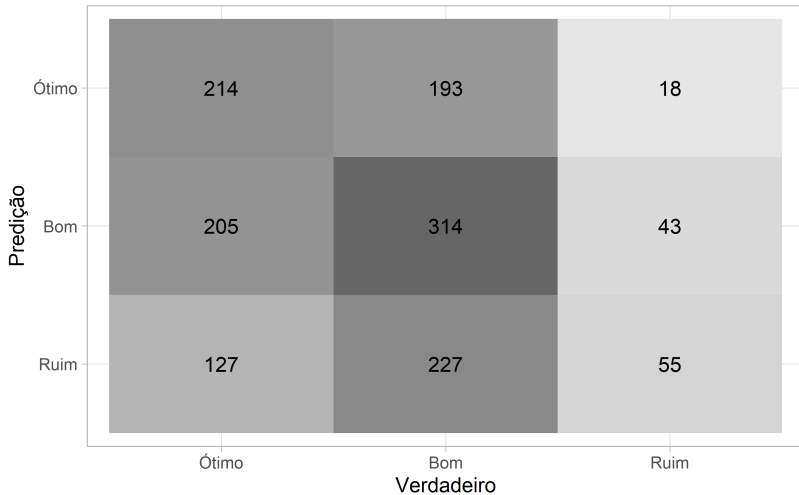
# VARIÁVEIS VIP

## Variáveis mais importantes no modelo



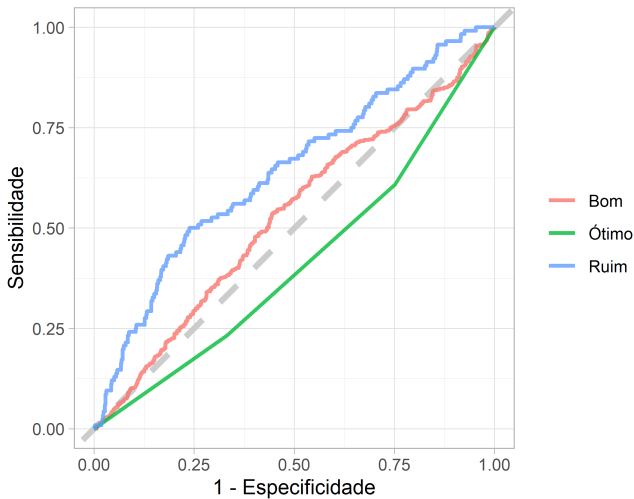
*HEATMAP*

Mapa de Calor das Predições



# AVALIANDO A CURVA ROC

## Curva ROC Modelo Final



# CONCLUSÃO

SE O MODELO DISSER QUE O LIVRO É ÓTIMO, DESCONFIE...

► Algumas métricas:

```
## Verdadeiro positivo
```

```
senbilidade_geral <- collect_predictions(stopping_fit) %>%  
  sens(book_rating, .pred_class)
```

```
## Verdadeiro negativo
```

```
especificidade_geral <- collect_predictions(stopping_fit) %>%  
  spec(book_rating, .pred_class)
```

```
sens_e_spec_categorias <- collect_predictions(stopping_fit, summarize =  
  roc_curve(book_rating, .pred_class:.pred_Ruim) %>%  
  group_by(.level) %>%  
  summarise(mean_sens=mean(sensitivity),  
            mean_spec=mean(specificity))
```

```
kable(senbilidade_geral, booktabs= TRUE) %>%  
  kable_styling(latex_options = "HOLD_position", position = "center")
```

## 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 Kaggle. Disponível em: <https://www.kaggle.com/jealousleopard/goodreadsbooks>