Classificação de gêneros musicais

Daniel Valentins de Lima

08/07/2020

```
library(tidyverse)
library(janitor)
library(tidymodels)
library(textrecipes)
library(tidytext)

theme_set(theme_minimal())
```

Importando os bancos

```
letras_bossa_nova <- read_csv("./letras_mus_br_bossa-nova.csv") %>%
  mutate(genero = "bossa_nova")

letras_mpb <- read_csv("./letras_mus_br_mpb.csv") %>%
  mutate(genero = "mpb")

letras <- letras_bossa_nova %>%
  bind_rows(letras_mpb) %>%
  select(-titulo, -artista) %>%
  mutate(genero = as_factor(genero))
```

Modelagem

Bancos de treino e teste

```
set.seed(0099)
letras_split <- initial_split(letras)

letras_training <- training(letras_split)
letras_testing <- testing(letras_split)
```

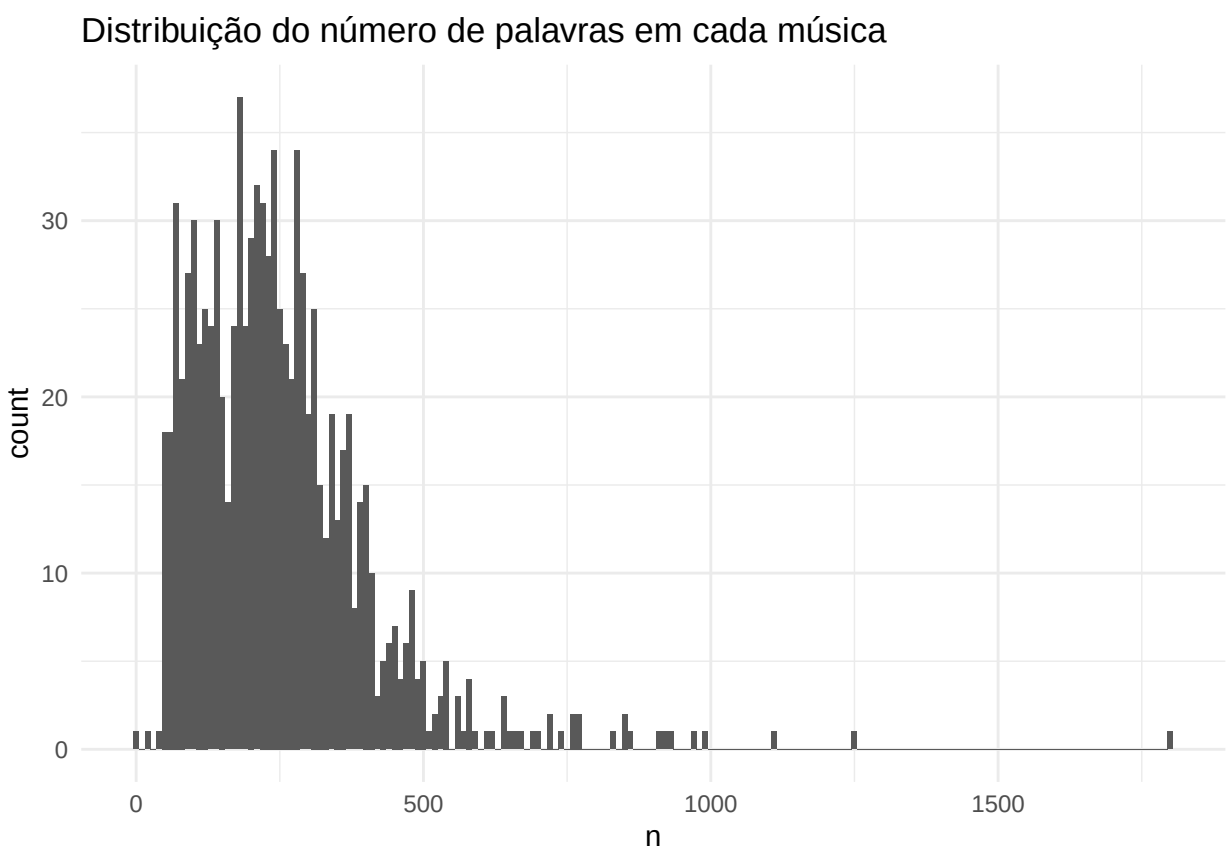
Processamento dos dados

```
letras_tokens <- letras_training %>%
  unnest_tokens(tokens, letras)

letras_tokens
```

```
## # A tibble: 237,008 x 3
##       X1 genero     tokens
##    <dbl> <fct>      <chr>
##  1     0 bossa_nova era
##  2     0 bossa_nova uma
##  3     0 bossa_nova casa
##  4     0 bossa_nova muito
##  5     0 bossa_nova engraçada
##  6     0 bossa_nova não
##  7     0 bossa_nova tinha
##  8     0 bossa_nova teto
##  9     0 bossa_nova não
## 10     0 bossa_nova tinha
## # ... with 236,998 more rows
```

```
letras_tokens %>%
  count(X1) %>%
  ggplot(aes(n)) +
  geom_histogram(binwidth = 10) +
  labs(title = "Distribuição do número de palavras em cada música")
```



```
rec_spec <- recipe(genero ~ letras, data = letras_training) %>%
  step_tokenize(letras) %>%
  step_stopwords(letras, language = "pt") %>%
  step_tokenfilter(letras, max_tokens = tune()) %>%
  step_tf(letras, weight_scheme = "binary")
```

Modelo SVM

```
mod_spec <- svm_rbf(cost = tune(), rbf_sigma = tune()) %>%
  set_engine("kernlab") %>%
  set_mode("classification")
```

```
set.seed(0099)
letras_folds <- vfold_cv(letras_training, v = 5)
```

```
letras_wf <- workflow() %>%
  add_recipe(rec_spec) %>%
  add_model(mod_spec)
```

```
tune_res <- tune_grid(
  letras_wf,
  resamples = letras_folds,
  grid = 25,
  control = control_grid(verbose = T)
)
```

Métricas

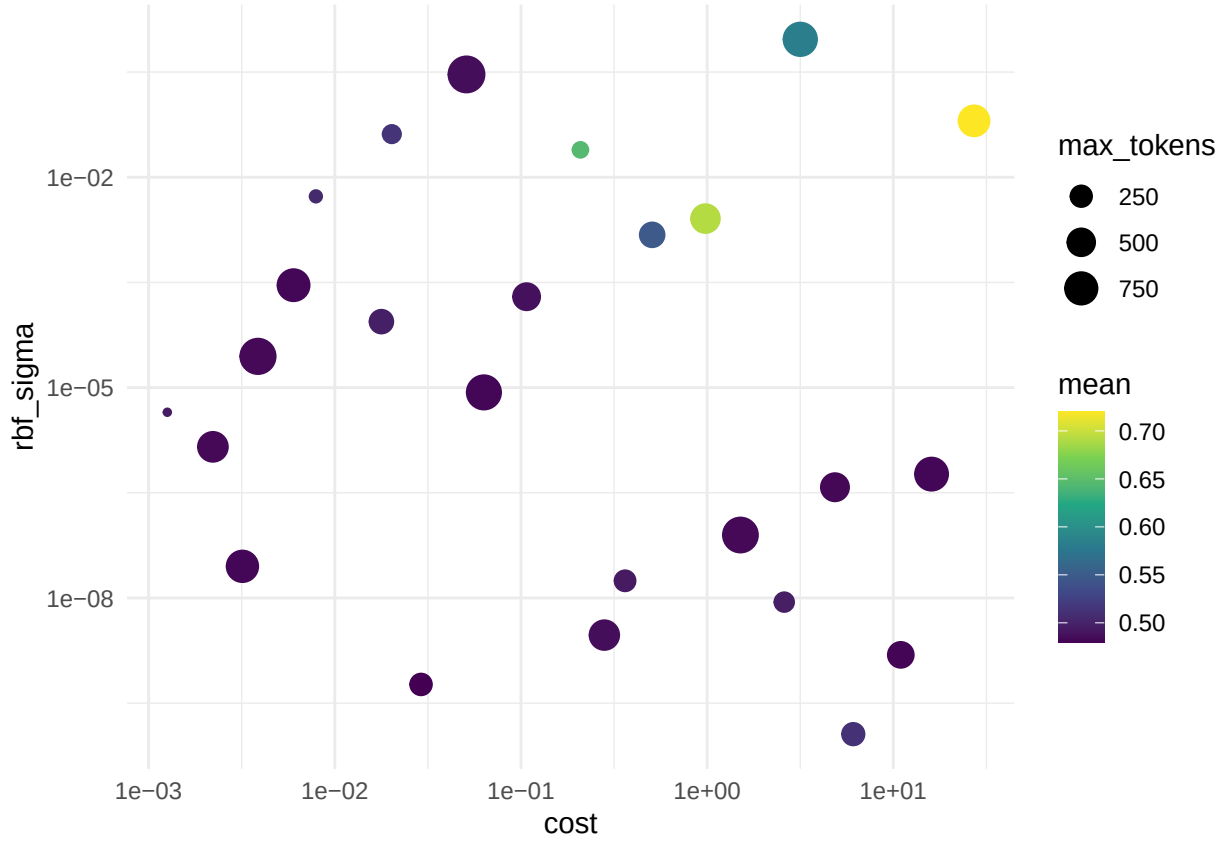
```
show_best(tune_res, "accuracy")
```

```
## # A tibble: 5 x 8
##     cost rbf_sigma max_tokens .metric  .estimator  mean     n std_err
##    <dbl>     <dbl>      <int> <chr>    <chr>      <dbl> <int>   <dbl>
## 1 27.1     0.0638         664 accuracy binary     0.719     5 0.0166
## 2  0.980   0.00257        550 accuracy binary     0.691     5 0.00720
## 3  0.209   0.0246          99 accuracy binary     0.645     5 0.0119
## 4  3.16    0.928          815 accuracy binary     0.583     5 0.0129
## 5  0.507   0.00151        368 accuracy binary     0.547     5 0.0385
```

Acurácia

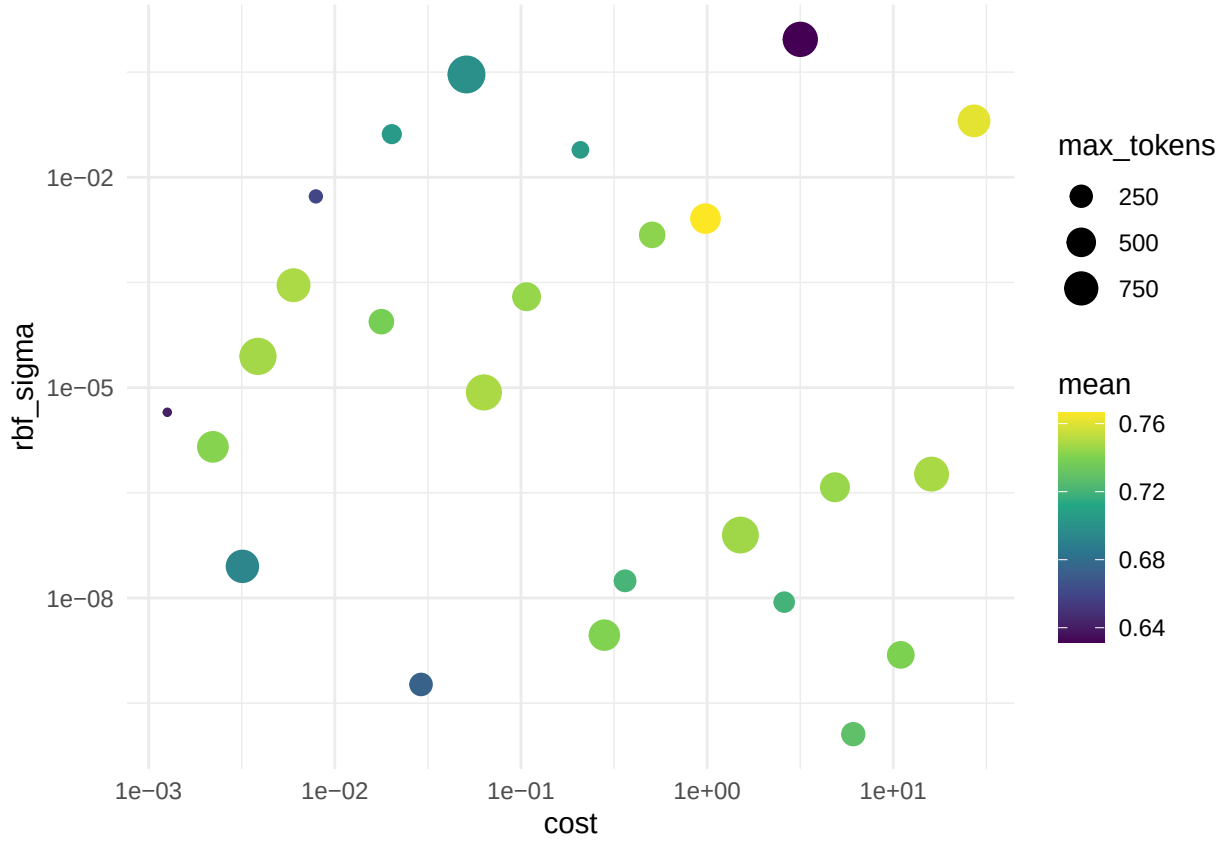
```
collect_metrics(tune_res) %>%
  filter(.metric == "accuracy") %>%
  ggplot(aes(cost, rbf_sigma, size = max_tokens, color = mean)) +
  geom_point() +
  scale_y_log10() +
```

```
  scale_x_log10() +
  scale_color_viridis_c()
```



Curva ROC

```
collect_metrics(tune_res) %>%
  filter(.metric == "roc_auc") %>%
  ggplot(aes(cost, rbf_sigma, size = max_tokens, color = mean)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() +
  scale_color_viridis_c()
```




```
best_accuracy <- select_best(tune_res, "accuracy")

final_wf <- finalize_workflow(
  letras_wf,
  best_accuracy
)

final_wf
```

```
## == Workflow ====================================================================================
## Preprocessor: Recipe
## Model: svm_rbf()
##
## -- Preprocessor ----------------------------------------------------------------------------
## 4 Recipe Steps
##
## * step_tokenize()
## * step_stopwords()
## * step_tokenfilter()
## * step_tf()
##
## -- Model -----------------------------------------------------------------------------------
## Radial Basis Function Support Vector Machine Specification (classification)
##
## Main Arguments:
```

```
##   cost = 27.1427682985615
##   rbf_sigma = 0.063797168553877
##
## Computational engine: kernlab
```

Finalizando o modelo

```
final_res <- final_wf %>%
  last_fit(letras_split)
```

```
final_res %>%
  collect_metrics()
```

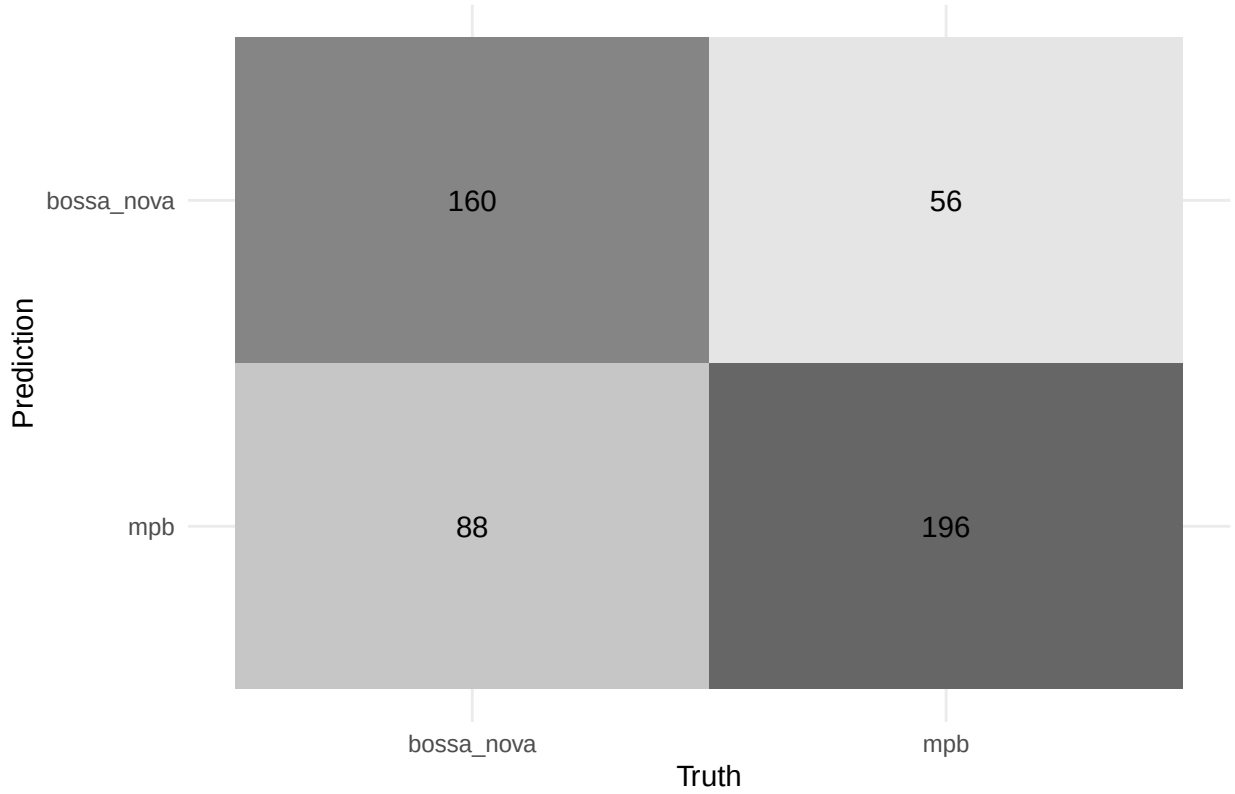
```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.712
## 2 roc_auc  binary         0.741
```

```
final_res %>%
  unnest(.predictions)
```

```
## # A tibble: 500 x 10
##    splits id    .metrics .notes .pred_bossa_nova .pred_mpb  .row .pred_class
##    <list> <chr> <list>   <list>            <dbl>     <dbl> <int> <fct>
##  1 <spli~ trai~ <tibble~ <tibb~            0.882     0.118     7 bossa_nova
##  2 <spli~ trai~ <tibble~ <tibb~            0.288     0.712     9 mpb
##  3 <spli~ trai~ <tibble~ <tibb~            0.882     0.118    10 bossa_nova
##  4 <spli~ trai~ <tibble~ <tibb~            0.299     0.701    15 mpb
##  5 <spli~ trai~ <tibble~ <tibb~            0.436     0.564    19 mpb
##  6 <spli~ trai~ <tibble~ <tibb~            0.873     0.127    20 bossa_nova
##  7 <spli~ trai~ <tibble~ <tibb~            0.308     0.692    26 mpb
##  8 <spli~ trai~ <tibble~ <tibb~            0.882     0.118    29 bossa_nova
##  9 <spli~ trai~ <tibble~ <tibb~            0.533     0.467    31 bossa_nova
## 10 <spli~ trai~ <tibble~ <tibb~            0.756     0.244    36 bossa_nova
## # ... with 490 more rows, and 2 more variables: genero <fct>, .workflow <list>
```

Matriz de confusão

```
final_res %>%
  unnest(.predictions) %>%
  conf_mat(truth = genero, .pred_class) %>%
  autoplot(type = "heatmap") +
  labs(title = "Matriz de confusão: Classificação de gênero musical")
```


Matriz de confusão: Classificação de gênero musical
bossa_nova
mpb
Prediction
160
56
88
196
bossa_nova
mpb
Truth