

Trabalho Aprendizado Supervisionado

AUTHOR

Bruna, Gabriele, Luiza e Sara

Pacotes [🔗](#)

```
# Instalar se necessário:  
# install.packages("tidymodels")  
# install.packages("vip")
```

```
library(tidymodels)
```

— Attaching packages — tidymodels 1.3.0 —

✓ broom	1.0.8	✓ recipes	1.3.1
✓ dials	1.4.0	✓ rsample	1.3.0
✓ dplyr	1.1.4	✓ tibble	3.3.0
✓ ggplot2	3.5.2	✓ tidyr	1.3.1
✓ infer	1.0.9	✓ tune	1.3.0
✓ modeldata	1.4.0	✓ workflows	1.2.0
✓ parsnip	1.3.2	✓ workflowsets	1.1.1
✓ purrr	1.1.0	✓ yardstick	1.3.2

— Conflicts — tidymodels_conflicts() —

```
* purrr::discard() masks scales::discard()  
* dplyr::filter() masks stats::filter()  
* dplyr::lag() masks stats::lag()  
* recipes::step() masks stats::step()
```

```
library(readr)
```

Attaching package: 'readr'

The following object is masked from 'package:yardstick':

spec

The following object is masked from 'package:scales':

col_factor

```
library(janitor)
```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

chisq.test, fisher.test

```
library(dplyr)
library(vip)
```

Attaching package: 'vip'

The following object is masked from 'package:utils':

vi

Carregar Dados

```
# Caminho corrigido
dados <- read_csv("/Users/marlonzotty/Downloads/dadoscoletados-Página1.csv")
```

Rows: 200 Columns: 8

— Column specification —

Delimiter: ",",

chr (6): modelo, marca, estado_conservacao, nota_fiscal, fonte, preco_reais

dbl (2): ano_lancamento, armazenamento

- i Use `spec()` to retrieve the full column specification for this data.
- i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```
# Padronizar nomes
dados <- clean_names(dados)

# Corrigir tipos
dados <- dados %>%
  mutate(
    preco_reais = as.numeric(preco_reais),
    modelo = as.factor(modelo),
    marca = as.factor(marca),
    estado_conservacao = as.factor(estado_conservacao),
    nota_fiscal = as.factor(nota_fiscal),
    fonte = as.factor(fonte),
    ano_lancamento = as.integer(ano_lancamento),
    armazenamento = as.integer(armazenamento)
  ) %>%
  filter(!is.na(preco_reais))
```

Warning: There was 1 warning in `mutate()`.

i In argument: `preco_reais = as.numeric(preco_reais)`.

Caused by warning:

! NAs introduced by coercion

```
glimpse(dados)
```

Rows: 172

Columns: 8

\$ modelo	<fct> Poco X3 NFC 128GB, Moto G10 64GB, Galaxy A52 128GB,...
\$ marca	<fct> Xiaomi, Motorola, Samsung, Motorola, Xiaomi, Apple,...
\$ ano_lancamento	<int> 2020, 2021, 2021, 2021, 2021, 2020, 2021, 2020, 202...
\$ armazenamento	<int> 128, 64, 128, 128, 64, 128, 128, 128, 128, 64, 128,...
\$ estado_conservacao	<fct> Usado, Com avarias, Usado, Novo, Usado, Novo, Usado...
\$ nota_fiscal	<fct> Sim, Não, Não, Sim, Sim, Sim, Sim, Sim, Não, Não, N...
\$ fonte	<fct> OLX, OLX, OLX, OLX, OLX, OLX, OLX, OLX, OLX, OLX, O...
\$ preco_reais	<dbl> 926, 1630, 880, 2364, 2355, 2040, 2540, 2130, 2144,...

Divisão Treino/Teste

```
set.seed(123)

split <- initial_split(dados, prop = 0.8)
train <- training(split)
test <- testing(split)

nrow(train)
```

```
[1] 137
```

```
nrow(test)
```

```
[1] 35
```

Receita

```
rec <- recipe(
  preco_reais ~ modelo + marca + ano_lancamento +
    armazenamento + estado_conservacao + nota_fiscal + fonte,
  data = train
) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_normalize(all_numeric_predictors())

rec
```

— Recipe —

— Inputs

Number of variables by role

```
outcome: 1
predictor: 7
```

— Operations

- Dummy variables from: `all_nominal_predictors()`
 - Centering and scaling for: `all_numeric_predictors()`
-

Workflows

Linear Regression

```
lm_spec <- linear_reg() %>%  
  set_engine("lm") %>%  
  set_mode("regression")  
  
wf_lm <- workflow() %>%  
  add_model(lm_spec) %>%  
  add_recipe(rec)
```

Random Forest

```
rf_spec <- rand_forest(  
  mtry = tune(),  
  trees = tune(),  
  min_n = tune()  
) %>%  
  set_engine("ranger", importance = "impurity") %>%  
  set_mode("regression")  
  
wf_rf <- workflow() %>%  
  add_model(rf_spec) %>%  
  add_recipe(rec)
```

Boosting (XGBoost)

```
boost_spec <- boost_tree(  
  trees = tune(),  
  learn_rate = tune(),  
  tree_depth = tune(),  
  min_n = tune()  
) %>%  
  set_engine("xgboost") %>%  
  set_mode("regression")  
  
wf_boost <- workflow() %>%  
  add_model(boost_spec) %>%  
  add_recipe(rec)
```

Validação Cruzada

```
foldes <- vfold_cv(train, v = 5)
foldes
```

```
# 5-fold cross-validation
# A tibble: 5 × 2
  splits      id
  <list>    <chr>
1 <split [109/28]> Fold1
2 <split [109/28]> Fold2
3 <split [110/27]> Fold3
4 <split [110/27]> Fold4
5 <split [110/27]> Fold5
```

Grades de Parâmetros

Random Forest

```
rf_grid <- grid_regular(
  mtry(range = c(2, 5)),
  trees(range = c(100, 500)),
  min_n(range = c(1, 5)),
  levels = 3
)

rf_grid
```

```
# A tibble: 27 × 3
  mtry trees min_n
  <int> <int> <int>
1     2   100     1
2     3   100     1
3     5   100     1
4     2   300     1
5     3   300     1
6     5   300     1
7     2   500     1
8     3   500     1
9     5   500     1
10    2   100     3
# i 17 more rows
```

Boosting

```
boost_grid <- grid_regular(
  trees(range = c(50, 300)),
  learn_rate(range = c(0.01, 0.3)),
```

```

tree_depth(range = c(2, 10)),
min_n(range = c(2, 20)),
levels = 5
)

boost_grid

```

A tibble: 625 × 4

	trees	learn_rate	tree_depth	min_n
	<int>	<dbl>	<int>	<int>
1	50	1.02	2	2
2	112	1.02	2	2
3	175	1.02	2	2
4	237	1.02	2	2
5	300	1.02	2	2
6	50	1.21	2	2
7	112	1.21	2	2
8	175	1.21	2	2
9	237	1.21	2	2
10	300	1.21	2	2

i 615 more rows

Tuning Random Forest

```

tune_rf <- tune_grid(
  wf_rf,
  resamples = folds,
  grid = rf_grid,
  metrics = metric_set(rmse, rsq)
)

```

→ A | warning: ! The following columns have zero variance so scaling cannot be used:

modelo_Black.Shark.3.128GB, modelo_Edge.20.Lite.128GB,
 modelo_Galaxy.A12.64GB, modelo_Galaxy.A51.128GB,
 modelo_Galaxy.A70.128GB,
 modelo_Galaxy.A73.128GB, modelo_Galaxy.F42.5G.128GB,
 modelo_Galaxy.M31.128GB,
 modelo_Galaxy.Note.10.Lite.128GB, modelo_Galaxy.Note.20.128GB,
 modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.S10.Lite.128GB,
 modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
 modelo_Galaxy.S22..256GB,
 modelo_Galaxy.S9.64GB, modelo_Galaxy.S9..128GB,
 modelo_Huawei.Nova.5T.128GB,
 ..., modelo_Zenfone.Max.Pro.M2.64GB, and marca_Huawei.

i Consider using ?step_zv (``?recipes::step_zv()``) to remove those
 columns
 before normalizing.

There were issues with some computations A: x1

→ B | warning: ! The following columns have zero variance so scaling cannot be used:
 modelo_Black.Shark.3.128GB, modelo_Galaxy.A51.128GB,
 modelo_Galaxy.A52.128GB,
 modelo_Galaxy.A73.128GB, modelo_Galaxy.F42.5G.128GB,
 modelo_Galaxy.M31.128GB,
 modelo_Galaxy.M52.5G.128GB, modelo_Galaxy.Note.20.Ultra.256GB,
 modelo_Galaxy.S10.Lite.128GB, modelo_Galaxy.S20..128GB,
 modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
 modelo_Galaxy.S9..128GB,
 modelo_Galaxy.Z.Fold2.256GB, modelo_Huawei.Nova.5T.128GB,
 modelo_Huawei.P30.128GB, modelo_iPhone.11.64GB,
 modelo_iPhone.11.Min.64GB,
 ..., modelo_Xiaomi.Mi.Note.10.128GB, and modelo_Xperia.1.II.256GB.
 i Consider using ?step_zv (`?recipes::step_zv()`) to remove those
 columns
 before normalizing.

There were issues with some computations A: x1
 There were issues with some computations A: x1 B: x1
 → C | warning: ! The following columns have zero variance so scaling cannot be used:
 modelo_Black.Shark.3.128GB, modelo_Galaxy.A12.128GB,
 modelo_Galaxy.A51.128GB,
 modelo_Galaxy.A71.128GB, modelo_Galaxy.A72.128GB,
 modelo_Galaxy.A73.128GB,
 modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.M31.128GB,
 modelo_Galaxy.M32.128GB,
 modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.Note.8.64GB,
 modelo_Galaxy.S10.128GB, modelo_Galaxy.S10.Lite.128GB,
 modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
 modelo_Galaxy.S9..128GB,
 modelo_Huawei.Nova.5T.128GB, modelo_Huawei.P30.128GB, ...,
 modelo_Xperia.1.II.256GB, and modelo_Xperia.10.II.128GB.
 i Consider using ?step_zv (`?recipes::step_zv()`) to remove those
 columns
 before normalizing.

There were issues with some computations A: x1 B: x1
 → D | warning: ! The following columns have zero variance so scaling cannot be used:
 modelo_Black.Shark.3.128GB, modelo_Edge.40.256GB,
 modelo_Galaxy.A04s.128GB,
 modelo_Galaxy.A13.64GB, modelo_Galaxy.A32.128GB,
 modelo_Galaxy.A33.128GB,
 modelo_Galaxy.A51.128GB, modelo_Galaxy.A53.128GB,
 modelo_Galaxy.A73.128GB,
 modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.M31.128GB,
 modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.Note.9.128GB,
 modelo_Galaxy.S10.Lite.128GB, modelo_Galaxy.S21.128GB,
 modelo_Galaxy.S21.256GB, modelo_Galaxy.S21.Ultra.256GB,
 modelo_Galaxy.S23.128GB, ..., modelo_Xperia.1.II.256GB, and
 modelo_Xperia.XA2.32GB.
 i Consider using ?step_zv (`?recipes::step_zv()`) to remove those
 columns
 before normalizing.

There were issues with some computations A: x1 B: x1
 There were issues with some computations A: x1 B: x1 C: x1 D: x1

```
→ E | warning: ! The following columns have zero variance so scaling cannot be used:
      modelo_Black.Shark.3.128GB, modelo_Galaxy.A51.128GB,
      modelo_Galaxy.A73.128GB,
      modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.J5.Prime.32GB,
      modelo_Galaxy.J7.Pro.64GB, modelo_Galaxy.M31.128GB,
      modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.S10.Lite.128GB,
      modelo_Galaxy.S10..512GB, modelo_Galaxy.S20.FE.128GB,
      modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
      modelo_Galaxy.S22.128GB,
      modelo_Galaxy.S9..128GB, modelo_Galaxy.Z.Flip3.256GB,
      modelo_Huawei.Nova.5T.128GB, modelo_Huawei.P30.128GB, ...,
      modelo_Xiaomi.Mi.Note.10.128GB, and modelo_Xperia.1.II.256GB.
i Consider using `step_zv(~?recipes::step_zv())` to remove those
columns
      before normalizing.
```

```
There were issues with some computations A: x1 B: x1 C: x1 D: x1
There were issues with some computations A: x1 B: x1 C: x1 D: x1 E: x1
There were issues with some computations A: x1 B: x1 C: x1 D: x1 E: x1
```

```
tune_rf
```

```
# Tuning results
# 5-fold cross-validation
# A tibble: 5 × 4
  splits      id      .metrics      .notes
  <list>    <chr> <list>      <list>
1 <split [109/28]> Fold1 <tibble [54 × 7]> <tibble [1 × 3]>
2 <split [109/28]> Fold2 <tibble [54 × 7]> <tibble [1 × 3]>
3 <split [110/27]> Fold3 <tibble [54 × 7]> <tibble [1 × 3]>
4 <split [110/27]> Fold4 <tibble [54 × 7]> <tibble [1 × 3]>
5 <split [110/27]> Fold5 <tibble [54 × 7]> <tibble [1 × 3]>
```

There were issues with some computations:

```
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
```

Run `show_notes(.Last.tune.result)` for more information.

Tuning Boosting

```
tune_boost <- tune_grid(
  wf_boost,
```



```

resamples = folds,
grid = boost_grid,
metrics = metric_set(rmse, rsq)
)

```

→ A | warning: ! The following columns have zero variance so scaling cannot be used:

modelo_Black.Shark.3.128GB, modelo_Edge.20.Lite.128GB,
 modelo_Galaxy.A12.64GB, modelo_Galaxy.A51.128GB,
 modelo_Galaxy.A70.128GB,
 modelo_Galaxy.A73.128GB, modelo_Galaxy.F42.5G.128GB,
 modelo_Galaxy.M31.128GB,
 modelo_Galaxy.Note.10.Lite.128GB, modelo_Galaxy.Note.20.128GB,
 modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.S10.Lite.128GB,
 modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
 modelo_Galaxy.S22..256GB,
 modelo_Galaxy.S9.64GB, modelo_Galaxy.S9..128GB,
 modelo_Huawei.Nova.5T.128GB,
 ..., modelo_Zenfone.Max.Pro.M2.64GB, and marca_Huawei.

i Consider using `?step_zv`(`?recipes::step_zv())`` to remove those columns before normalizing.

→ B | warning: ! The following columns have zero variance so scaling cannot be used:

modelo_Black.Shark.3.128GB, modelo_Galaxy.A51.128GB,
 modelo_Galaxy.A52.128GB,
 modelo_Galaxy.A73.128GB, modelo_Galaxy.F42.5G.128GB,
 modelo_Galaxy.M31.128GB,
 modelo_Galaxy.M52.5G.128GB, modelo_Galaxy.Note.20.Ultra.256GB,
 modelo_Galaxy.S10.Lite.128GB, modelo_Galaxy.S20..128GB,
 modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
 modelo_Galaxy.S9..128GB,
 modelo_Galaxy.Z.Fold2.256GB, modelo_Huawei.Nova.5T.128GB,
 modelo_Huawei.P30.128GB, modelo_iPhone.11.64GB,
 modelo_iPhone.11.Min.64GB,
 ..., modelo_Xiaomi.Mi.Note.10.128GB, and modelo_Xperia.1.II.256GB.

i Consider using `?step_zv`(`?recipes::step_zv())`` to remove those columns before normalizing.

There were issues with some computations A: x1 B: x1

→ C | warning: ! The following columns have zero variance so scaling cannot be used:

modelo_Black.Shark.3.128GB, modelo_Galaxy.A12.128GB,
 modelo_Galaxy.A51.128GB,
 modelo_Galaxy.A71.128GB, modelo_Galaxy.A72.128GB,
 modelo_Galaxy.A73.128GB,
 modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.M31.128GB,
 modelo_Galaxy.M32.128GB,
 modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.Note.8.64GB,
 modelo_Galaxy.S10.128GB, modelo_Galaxy.S10.Lite.128GB,
 modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
 modelo_Galaxy.S9..128GB,
 modelo_Huawei.Nova.5T.128GB, modelo_Huawei.P30.128GB, ...,
 modelo_Xperia.1.II.256GB, and modelo_Xperia.10.II.128GB.

```
i Consider using ?step_zv (`?recipes::step_zv()`) to remove those
columns
before normalizing.
There were issues with some computations A: x1 B: x1
There were issues with some computations A: x1 B: x1 C: x1
→ D | warning: ! The following columns have zero variance so scaling cannot be used:
      modelo_Black.Shark.3.128GB, modelo_Edge.40.256GB,
modelo_Galaxy.A04s.128GB,
      modelo_Galaxy.A13.64GB, modelo_Galaxy.A32.128GB,
modelo_Galaxy.A33.128GB,
      modelo_Galaxy.A51.128GB, modelo_Galaxy.A53.128GB,
modelo_Galaxy.A73.128GB,
      modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.M31.128GB,
      modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.Note.9.128GB,
      modelo_Galaxy.S10.Lite.128GB, modelo_Galaxy.S21.128GB,
      modelo_Galaxy.S21.256GB, modelo_Galaxy.S21.Ultra.256GB,
      modelo_Galaxy.S23.128GB, ..., modelo_Xperia.1.II.256GB, and
      modelo_Xperia.XA2.32GB.
```

```
i Consider using ?step_zv (`?recipes::step_zv()`) to remove those
columns
before normalizing.
There were issues with some computations A: x1 B: x1 C: x1
There were issues with some computations A: x1 B: x1 C: x1 D: x1
→ E | warning: ! The following columns have zero variance so scaling cannot be used:
      modelo_Black.Shark.3.128GB, modelo_Galaxy.A51.128GB,
modelo_Galaxy.A73.128GB,
      modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.J5.Prime.32GB,
      modelo_Galaxy.J7.Pro.64GB, modelo_Galaxy.M31.128GB,
      modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.S10.Lite.128GB,
      modelo_Galaxy.S10..512GB, modelo_Galaxy.S20.FE.128GB,
      modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB,
modelo_Galaxy.S22.128GB,
      modelo_Galaxy.S9..128GB, modelo_Galaxy.Z.Flip3.256GB,
      modelo_Huawei.Nova.5T.128GB, modelo_Huawei.P30.128GB, ...,
      modelo_Xiaomi.Mi.Note.10.128GB, and modelo_Xperia.1.II.256GB.
```

```
i Consider using ?step_zv (`?recipes::step_zv()`) to remove those
columns
before normalizing.
There were issues with some computations A: x1 B: x1 C: x1 D: x1
There were issues with some computations A: x1 B: x1 C: x1 D: x1 E: x1
There were issues with some computations A: x1 B: x1 C: x1 D: x1 E: x1
```

```
tune_boost
```

```
# Tuning results
# 5-fold cross-validation
# A tibble: 5 × 4
  splits      id  .metrics      .notes
  <list>    <chr> <list>      <list>
1 <split [109/28]> Fold1 <tibble [1,250 × 8]> <tibble [1 × 3]>
2 <split [109/28]> Fold2 <tibble [1,250 × 8]> <tibble [1 × 3]>
3 <split [110/27]> Fold3 <tibble [1,250 × 8]> <tibble [1 × 3]>
4 <split [110/27]> Fold4 <tibble [1,250 × 8]> <tibble [1 × 3]>
```

```
5 <split [110/27]> Fold5 <tibble [1,250 × 8]> <tibble [1 × 3]>
```

There were issues with some computations:

```
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
- Warning(s) x1: ! The following columns have zero variance so scaling cannot be
...
```

Run ``show_notes(.Last.tune.result)`` for more information.

Linear Model (sem tuning)

```
fit_lm <- fit(wf_lm, data = train)
```

Warning: ! The following columns have zero variance so scaling cannot be used:
modelo_Black.Shark.3.128GB, modelo_Galaxy.A51.128GB, modelo_Galaxy.A73.128GB,
modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.M31.128GB,
modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.S10.Lite.128GB,
modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB, modelo_Galaxy.S9..128GB,
modelo_Huawei.Nova.5T.128GB, modelo_Huawei.P30.128GB, modelo_iPhone.11.64GB,
modelo_iPhone.11.Min.64GB, modelo_iPhone.12.Min.128GB,
modelo_iPhone.13.128GB, modelo_iPhone.15.Pro.256GB,
modelo_iPhone.7.Plus.128GB, ..., modelo_Xiaomi.Mi.Note.10.128GB, and
modelo_Xperia.1.II.256GB.
i Consider using `?step_zv`(`recipes::step_zv())`` to remove those columns
before normalizing.

```
lm_preds <- predict(fit_lm, new_data = test) %>%  
  bind_cols(test)
```

Warning in `predict.lm(object = object$fit, newdata = new_data, type = "response", : prediction from rank-deficient fit; consider predict(., rankdeficient="NA")`

```
lm_metrics <- metrics(lm_preds, truth = preco_reais, estimate = .pred)  
  
print("Resultados Linear Model:")
```

```
[1] "Resultados Linear Model:"
```

```
print(lm_metrics)
```

```
# A tibble: 3 × 3
  .metric .estimator .estimate
  <chr>   <chr>         <dbl>
1 rmse    standard      1100.
2 rsq     standard        0.0125
3 mae     standard        790.
```

Melhor Random Forest

```
best_rf <- select_best(tune_rf, metric = "rmse")

rf_final_wf <- finalize_workflow(wf_rf, best_rf)

rf_fit <- fit(rf_final_wf, data = train)
```

Warning: ! The following columns have zero variance so scaling cannot be used: modelo_Black.Shark.3.128GB, modelo_Galaxy.A51.128GB, modelo_Galaxy.A73.128GB, modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.M31.128GB, modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.S10.Lite.128GB, modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB, modelo_Galaxy.S9..128GB, modelo_Huawei.Nova.5T.128GB, modelo_Huawei.P30.128GB, modelo_iPhone.11.64GB, modelo_iPhone.11.Mini.64GB, modelo_iPhone.12.Mini.128GB, modelo_iPhone.13.128GB, modelo_iPhone.15.Pro.256GB, modelo_iPhone.7.Plus.128GB, ..., modelo_Xiaomi.Mi.Note.10.128GB, and modelo_Xperia.1.II.256GB.

i Consider using `?step_zv` (`?recipes::step_zv()`) to remove those columns before normalizing.

```
rf_preds <- predict(rf_fit, new_data = test) %>%
  bind_cols(test)

rf_metrics <- metrics(rf_preds, truth = preco_reais, estimate = .pred)

print("Resultados Random Forest:")
```

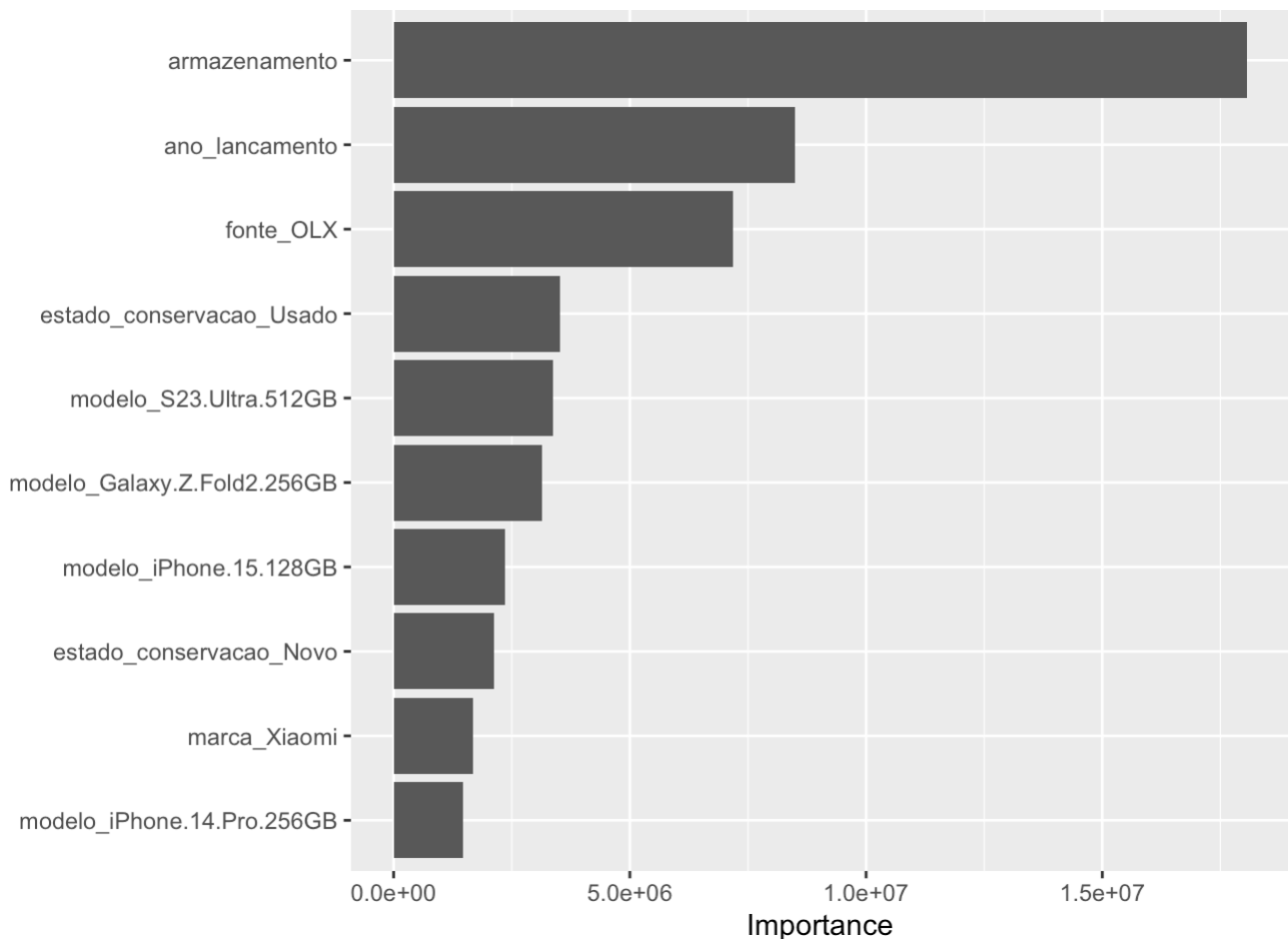
```
[1] "Resultados Random Forest:"
```

```
print(rf_metrics)
```

```
# A tibble: 3 × 3
  .metric .estimator .estimate
  <chr>   <chr>         <dbl>
1 rmse    standard        835.
2 rsq     standard        0.514
3 mae     standard        719.
```

Importância das Variáveis (Random Forest)

```
rf_fit %>%
  extract_fit_parsnip() %>%
  vip()
```



Melhor Boosting

```
best_boost <- select_best(tune_boost, metric = "rmse")

boost_final_wf <- finalize_workflow(wf_boost, best_boost)

boost_fit <- fit(boost_final_wf, data = train)
```

Warning: ! The following columns have zero variance so scaling cannot be used:
 modelo_Black.Shark.3.128GB, modelo_Galaxy.A51.128GB, modelo_Galaxy.A73.128GB,
 modelo_Galaxy.F42.5G.128GB, modelo_Galaxy.M31.128GB,
 modelo_Galaxy.Note.20.Ultra.256GB, modelo_Galaxy.S10.Lite.128GB,
 modelo_Galaxy.S21.128GB, modelo_Galaxy.S21.256GB, modelo_Galaxy.S9..128GB,
 modelo_Huawei.Nova.5T.128GB, modelo_Huawei.P30.128GB, modelo_iPhone.11.64GB,
 modelo_iPhone.11.Mini.64GB, modelo_iPhone.12.Mini.128GB,
 modelo_iPhone.13.128GB, modelo_iPhone.15.Pro.256GB,
 modelo_iPhone.7.Plus.128GB, ..., modelo_Xiaomi.Mi.Note.10.128GB, and
 modelo_Xperia.1.II.256GB.

i Consider using `?step_zv` (``?recipes::step_zv()``) to remove those columns before normalizing.

```
boost_preds <- predict(boost_fit, new_data = test) %>%
  bind_cols(test)

boost_metrics <- metrics(boost_preds, truth = preco_reais, estimate = .pred)

print("Resultados Boosting:")
```

```
[1] "Resultados Boosting:"
```

```
print(boost_metrics)
```

```
# A tibble: 3 × 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rmse    standard      756.
2 rsq     standard       0.466
3 mae     standard      580.
```

Comparação Final

```
cat("\n===== Comparação =====\n")
```

```
===== Comparação =====
```

```
cat("\nLinear Model:\n")
```

Linear Model:

```
print(lm_metrics)
```

```
# A tibble: 3 × 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rmse    standard     1100.
2 rsq     standard       0.0125
3 mae     standard       790.
```

```
cat("\nRandom Forest:\n")
```

Random Forest:

```
print(rf_metrics)
```

```
# A tibble: 3 × 3
  .metric .estimator .estimate
```

	<chr>	<chr>	<dbl>
1	rmse	standard	835.
2	rsq	standard	0.514
3	mae	standard	719.

```
cat("\nBoosting:\n")
```

Boosting:









```
print(boost_metrics)
```

```
# A tibble: 3 × 3  
  .metric .estimator .estimate  
  <chr>    <chr>        <dbl>  
1 rmse    standard      756.  
2 rsq     standard      0.466  
3 mae     standard      580.
```

Salvar Ambiente

```
save.image("/Users/marlonzotty/Downloads/meu_projeto_final.RData")
```

RESUMO

-  Dados lidos corretamente
 -  Divisão treino/teste
 -  Receita criada
 -  Workflows configurados
 -  Random Forest e Boosting com tuning
 -  Métricas calculadas
 -  Comparação final
 -  Ambiente salvo
-