# EST-25134: Aprendizaje Estadístico

**Profesor**: Alfredo Garbuno Iñigo — Primavera, 2023 — Interpretabilidad y explicabilidad.
**Objetivo**: Que veremos.
**Lectura recomendada**: [1] y [2].

## 1. INTRODUCCIÓN

Ejemplo tomado de: Tune random forests for #TidyTuesday IKEA prices.
Los datos que tenemos disponibles son los siguientes.

```
ikea_df ← ikea ▷
  select(price, name, category, depth, height, width) ▷
  mutate(price = log10(price)) ▷
  mutate_if(is.character, factor)

ikea_df ▷ print(n = 5)
```

```
# A tibble: 3,694 × 6
  price name                   category      depth height width
  <dbl> <fct>                  <fct>         <dbl>  <dbl> <dbl>
1  2.42 FREKVENS               Bar furniture    NA     99    51
2  3.00 NORDVIKEN              Bar furniture    NA    105    80
3  3.32 NORDVIKEN / NORDVIKEN  Bar furniture    NA     NA    NA
4  1.84 STIG                   Bar furniture    50    100    60
5  2.35 NORBERG                Bar furniture    60     43    74
# ... with 3,689 more rows
#  Use `print(n = ...)` to see more rows
```

### 1.1. Preparación de datos

```
set.seed(123)
ikea_split ← initial_split(ikea_df, strata = price)
ikea_train ← training(ikea_split)
ikea_test ← testing(ikea_split)

set.seed(234)
ikea_folds ← vfold_cv(ikea_train, strata = price)
```

```
library(textrecipes)
ranger_recipe ←
  recipe(formula = price ~ ., data = ikea_train) ▷
  step_other(name, category, threshold = 0.01) ▷
  step_clean_levels(name, category) ▷
  step_impute_knn(depth, height, width)
```

## 1.2. Especificación del modelo

```
ranger_spec ←
  rand_forest(trees = 1000) ▷
  set_mode("regression") ▷
  set_engine("ranger")

ranger_workflow ←
  workflow() ▷
  add_recipe(ranger_recipe) ▷
  add_model(ranger_spec)
```

```
all_cores ← parallel::detectCores(logical = TRUE) - 1
library(doParallel)
cl ← makePSOCKcluster(all_cores)
registerDoParallel(cl)
```

```
ikea_rf ← ranger_workflow ▷ fit(data = ikea_train)
ikea_rf
```

```
 Workflow [trained]
Preprocessor: Recipe
Model: rand_forest()

 Preprocessor
3 Recipe Steps•

 step_other()•
 step_clean_levels()•
 step_impute_knn()

 Model
Ranger result

Call:
 ranger::ranger(x = maybe_data_frame(x), y = y, num.trees = ~1000,       num.
    threads = 1, verbose = FALSE, seed = sample.int(10^5,          1))

Type:                             Regression
Number of trees:                  1000
Sample size:                      2770
Number of independent variables:  5
Mtry:                             2
Target node size:                 5
Variable importance mode:         none
Splitrule:                        variance
OOB prediction error (MSE):       0.1017
R squared (OOB):                  0.7576
```

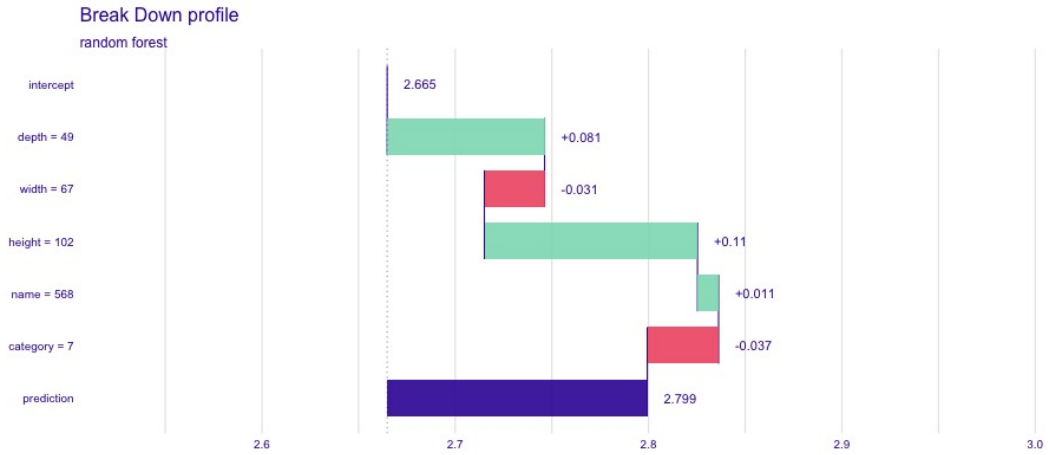
# 2. INTERPRETABILIDAD

```
library(DALEXtra)
```

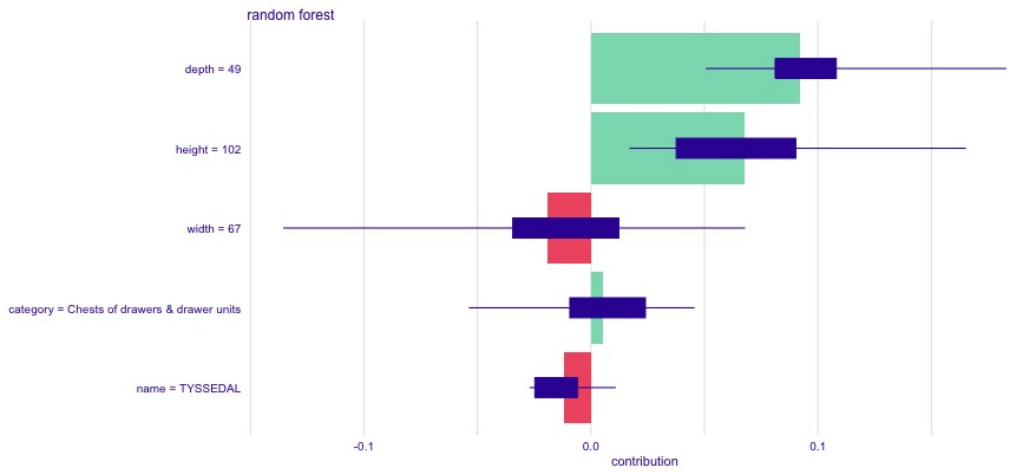
```
explainer_rf ←
  explain_tidymodels(
    ikea_rf,
    data = ikea_train ▷ select(-price),
    y    = ikea_train ▷ pull(price),
    label = "random forest",
    verbose = FALSE
  )
```

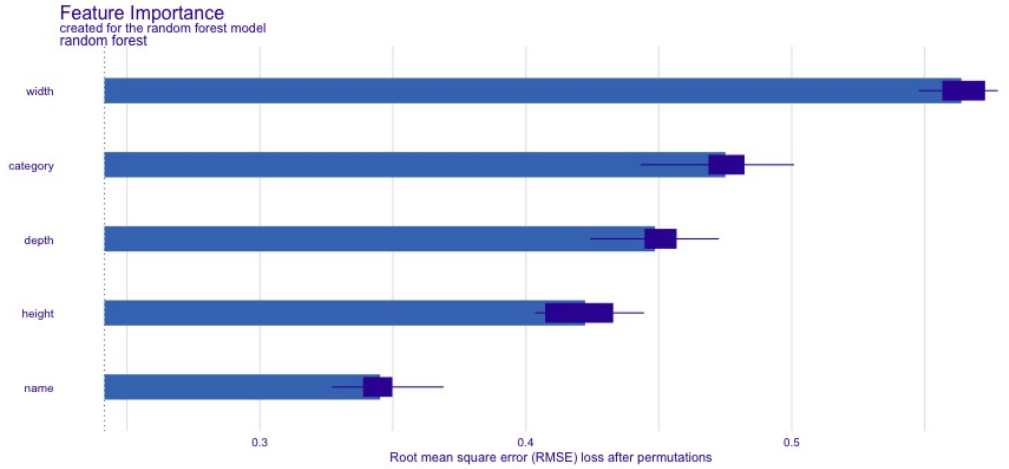
```
set.seed(123)
mueble ← ikea_test ▷ sample_n(1)
mueble
```

```
rf_breakdown ← predict_parts(explainer = explainer_rf, new_observation =
    mueble)
rf_breakdown
```

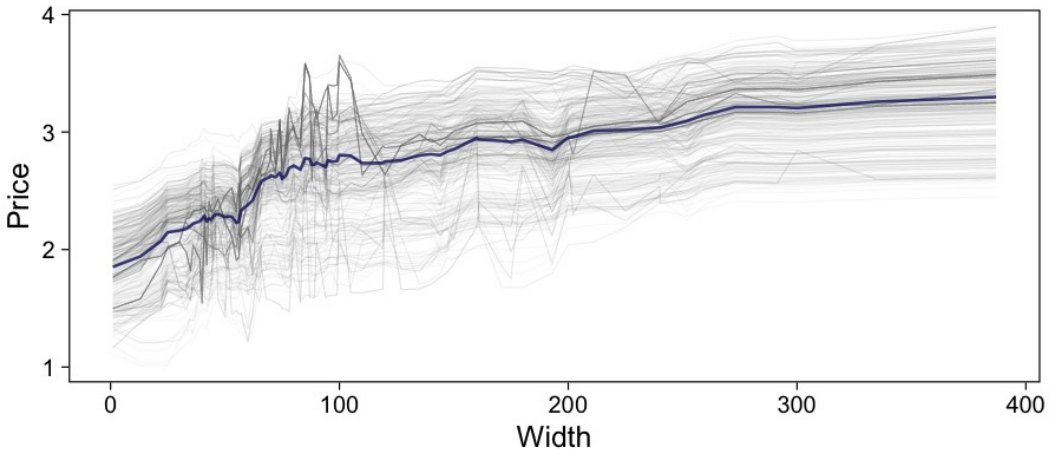



```
set.seed(1801)
shap_mueble ←
  predict_parts(
    explainer = explainer_rf,
    new_observation = mueble,
    type = "shap",
    B = 20
  )
```

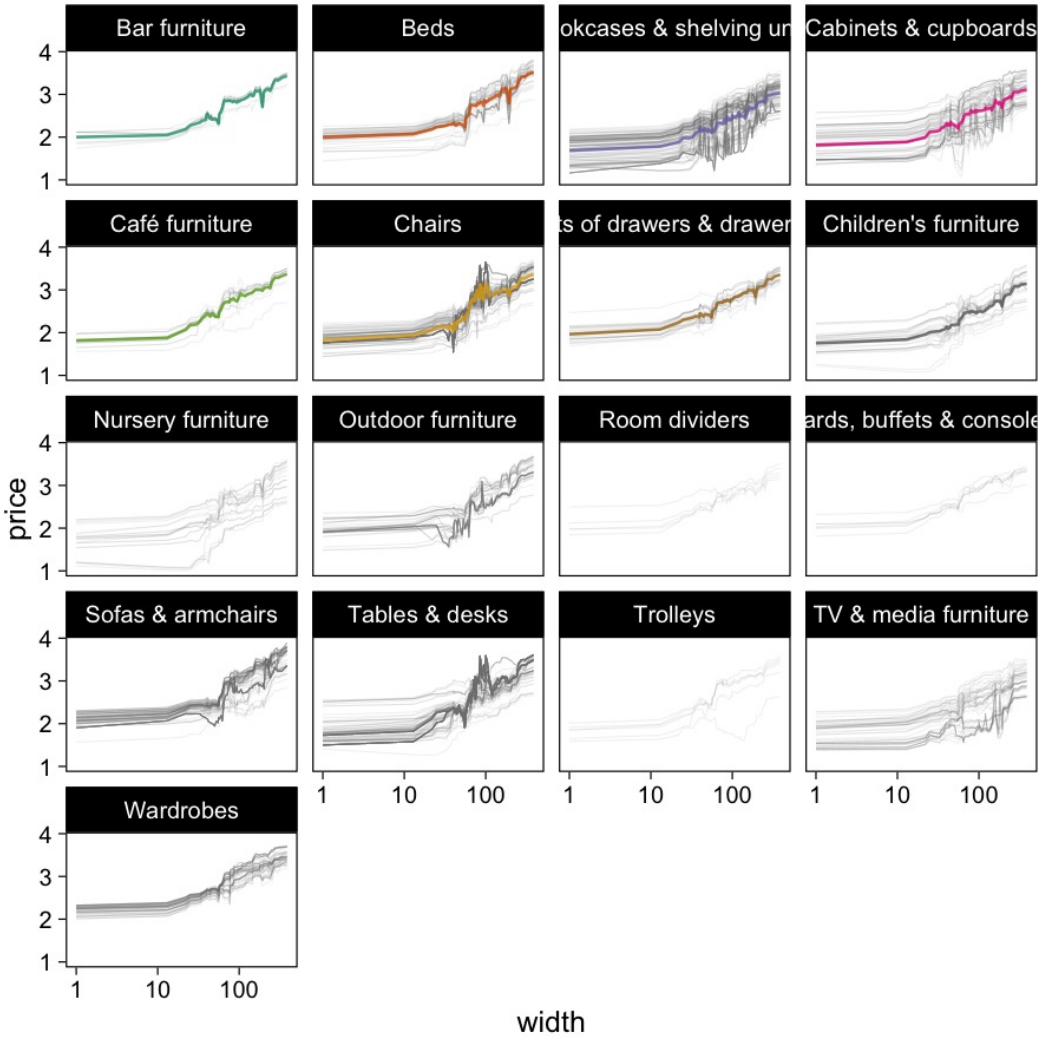



```
set.seed(1805)
pdp_width ← model_profile(explainer_rf, N = 500, variables = "width")
```




```
set.seed(1806)
pdp_wcat ← model_profile(explainer_rf, N = 1000,
                         variables = "width",
                         groups = "category")
```




## REFERENCIAS

[1] P. Biecek and T. Burzykowski. *Explanatory Model Analysis: Explore, Explain, and Examine Predictive Models*. Chapman & Hall/CRC Data Science Series. CRC Press, Boca Raton, first edition, 2021. ISBN 978-0-367-13559-1. 1

[2] C. Molnar. *Interpretable Machine Learning*. Lean Pub, 2020. 1