CU55_MODEL_DEVELOPMENT_01_XGBOOST

June 13, 2023

#

CU55_Modelo agregado de estimación del gasto medio por turista

1 IV. Model development

En este anexo se incluye el código utilizado durante el desarrollo de los modelos incluidos en el caso de uso.

1.1 Modelo XGBOOST

[1]:
```
Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

'es_ES.UTF-8/es_ES.UTF-8/es_ES.UTF-8/C/es_ES.UTF-8/C'

1.1.1 Paquetes

[2]:
```
library(readr)
library(dplyr)
library(tidyr)
library(stringr)
library(xgboost)
```

```
Attaching package: 'dplyr'


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

    filter, lag


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

    intersect, setdiff, setequal, union



Attaching package: 'xgboost'
```

```
The following object is masked from 'package:dplyr':

    slice
```

1.1.2 Datos

[3]:
```
gasto_municipio <- read_csv("CU_55_05_02_gasto_municipio.csv")

dm <- gasto_municipio |>
  mutate(nmes = factor(str_sub(mes, 6, 7)),
         pais_orig = factor(pais_orig)) |>
  select(nmes, pais_orig, turistas, gasto) |>
  filter(str_detect(pais_orig, "Total", negate = TRUE))

mm <- model.matrix( ~ ., dm)
```

```
Rows: 50294 Columns: 8
  Column specification

Delimiter: ","
chr (5): mes, pais_orig_cod, pais_orig, mun_dest, CMUN
dbl (3): mun_dest_cod, turistas, gasto

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

1.2 Splitting

[4]:
```
# mm <- Matrix::sparse.model.matrix(~., data = dm)

train <- sample(1:nrow(mm), round(0.8* nrow(mm)), replace = FALSE)
x.train <- mm[train, 1:(ncol(mm) - 1)]
y.train <- mm[train, ncol(mm)]

x.test <- mm[-train, 1:(ncol(mm) - 1)]
y.test <- mm[-train, ncol(mm)]
```

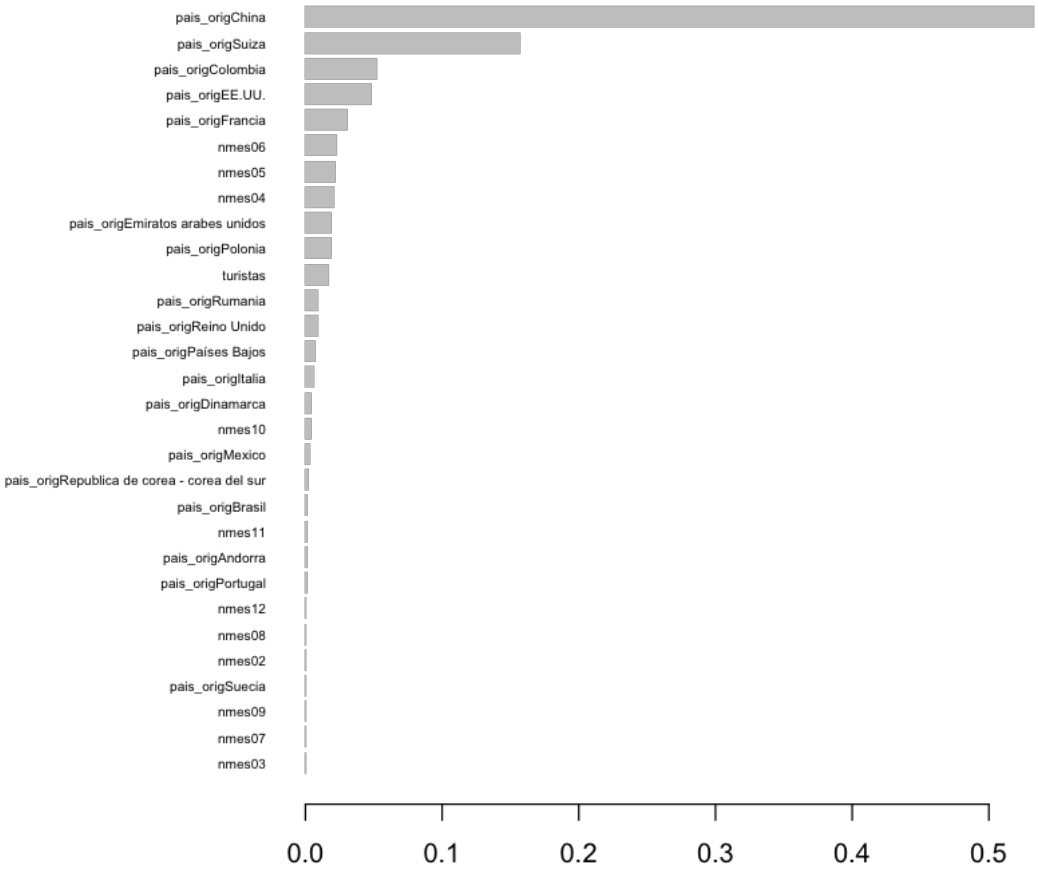
1.3 Modelo

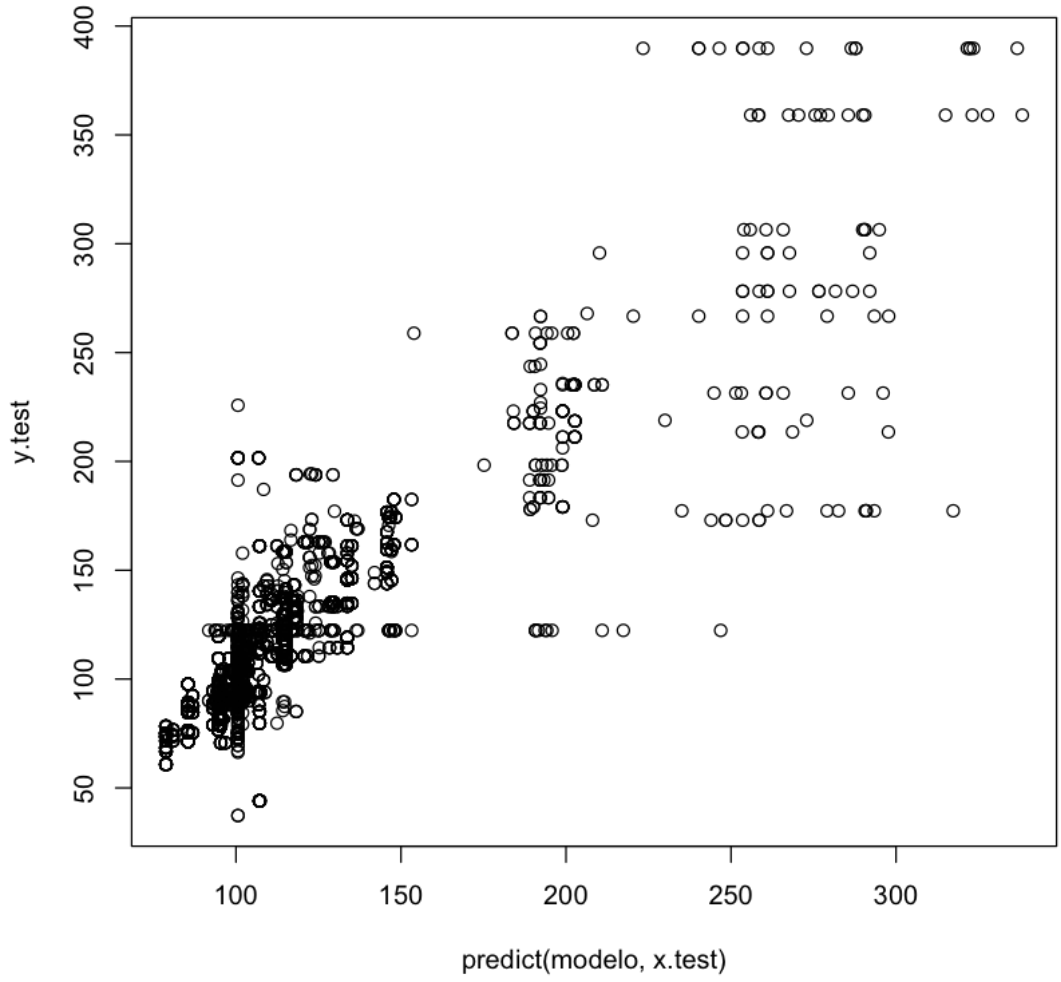
[5]:
```
modelo <- xgboost(data = x.train, label = y.train, nrounds = 10)

importance <- xgb.importance(feature_names = colnames(x.train), model = modelo)
head(importance)

xgb.plot.importance(importance_matrix = importance)

plot(predict(modelo, x.test), y.test)

write_rds(modelo, "modelo_xgb.rds")
```

```
[1]     train-rmse:83.689009
[2]     train-rmse:60.628673
[3]     train-rmse:45.010463
[4]     train-rmse:34.669155
[5]     train-rmse:27.991529
[6]     train-rmse:23.817382
[7]     train-rmse:21.389982
[8]     train-rmse:19.915366
[9]     train-rmse:18.972105
[10]    train-rmse:18.453314
```

A data.table: 6 x 4

Feature <chr>	Gain <dbl>	Cover <dbl>	Frequency <dbl>
pais_origChina	0.53268228	0.14526697	0.02821317
pais_origSuiza	0.15703532	0.09873650	0.01880878
pais_origColombia	0.05277161	0.07341269	0.01567398
pais_origEE.UU.	0.04806993	0.07826957	0.01567398
pais_origFrancia	0.03114864	0.05373272	0.02821317
nmes06	0.02257595	0.05999225	0.06269592


pais_origChina
pais_origSuiza
pais_origColombia
pais_origEE.UU.
pais_origFrancia
nmes06
nmes05
nmes04
pais_origEmiratos arabes unidos
pais_origPolonia
turistas
pais_origRumania
pais_origReino Unido
pais_origPaíses Bajos
pais_origItalia
pais_origDinamarca
nmes10
pais_origMexico
pais_origRepublica de corea - corea del sur
pais_origBrasil
nmes11
pais_origAndorra
pais_origPortugal
nmes12
nmes08
nmes02
pais_origSuecia
nmes09
nmes07
nmes03
0.0
0.1
0.2
0.3
0.4
0.5



1.4 Predicción

[6]:
```
### predicción
## 1. tipo escenario origen

escenario <- read_csv("ESCENARIO_ORIGEN.csv")

escenario.x <- escenario |>
  mutate(nmes = factor(str_sub(mes, 6, 7), levels = levels(dm$nmes)),
         pais_orig = factor(pais_orig, levels = levels(dm$pais_orig))) |>
  select(nmes, pais_orig, turistas) |>
  model.matrix(~., data = _)
```

```r
predict(modelo, escenario.x)
## 1. tipo escenario destino

escenario <- read_csv("ESCENARIO_DESTINO.csv")

escenario.x <- escenario |>
  mutate(nmes = factor(str_sub(mes, 6, 7), levels = levels(dm$nmes)),
         pais_orig = factor(pais_orig, levels = levels(dm$pais_orig))) |>
  select(nmes, pais_orig, turistas) |>
  model.matrix(~., data = _)

predict(modelo, escenario.x)
```

```
Rows: 30 Columns: 4
  Column specification

Delimiter: ","
chr (3): mes, pais_orig, mun_dest
dbl (1): turistas

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

1. 85.4694519042969 2. 85.4694519042969 3. 85.4694519042969 4. 85.4694519042969
5. 85.4694519042969 6. 85.4694519042969 7. 85.4694519042969 8. 85.4694519042969
9. 85.4694519042969 10. 85.4694519042969 11. 85.4694519042969 12. 85.4694519042969
13. 85.4694519042969 14. 85.4694519042969 15. 85.4694519042969 16. 85.4694519042969
17. 85.4694519042969 18. 85.4694519042969 19. 85.4694519042969 20. 85.4694519042969
21. 85.4694519042969 22. 86.8655319213867 23. 86.8655319213867 24. 86.8655319213867
25. 86.8655319213867 26. 86.8655319213867 27. 86.8655319213867 28. 86.8655319213867
29. 86.8655319213867 30. 86.8655319213867

```
Rows: 24 Columns: 4
  Column specification

Delimiter: ","
chr (3): mes, pais_orig, mun_dest
dbl (1): turistas

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

1. 100.598556518555 2. 100.598556518555 3. 100.598556518555 4. 95.3761825561523
5. 85.4694519042969 6. 100.598556518555 7. 100.598556518555 8. 94.6509704589844
9. 100.598556518555 10. 93.0525054931641 11. 78.7714691162109 12. 102.962203979492

13. 112.434280395508 14. 100.598556518555 15. 115.144866943359 16. 100.598556518555 17. 194.728652954102 18. 100.598556518555 19. 100.598556518555 20. 100.598556518555 21. 100.598556518555 22. 133.715316772461 23. 145.716323852539 24. 258.420227050781