# CU25_MODEL_DEVELOPMENT_01_XGBOOST

June 14, 2023

#

CU25_Modelo de gestión de Lista de Espera Quirúrgica

## 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]:
```
# https://www.business-science.io/code-tools/2020/06/29/introducing-modeltime.
↪html

## ESTE MODELO SE GUARDA EN MAESTROS PARA DESPUÉS CARGARLO Y HACER PREDICCIÓN
## Para la predicción solo hace falta el horizonte


library(tidymodels)
library(modeltime)
library(timetk)
library(lubridate)
library(tidyverse)
```

```
  Attaching packages                      tidymodels
1.0.0

  broom        1.0.3      recipes
1.0.4
  dials        1.1.0      rsample
1.1.1
  dplyr        1.1.0      tibble
3.2.1
```

```
 ggplot2      3.4.2      tidyr
1.3.0
 infer        1.0.4      tune
1.0.1
 modeldata    1.1.0      workflows
1.1.3
 parsnip      1.0.4      workflowsets
1.0.0
 purrr        1.0.1      yardstick
1.1.0

  Conflicts
tidymodels_conflicts()
 purrr::discard() masks
scales::discard()
 dplyr::filter()  masks stats::filter()
 dplyr::lag()     masks stats::lag()
 recipes::step()  masks stats::step()
• Search for functions across packages at
https://www.tidymodels.org/find/


Attaching package: 'lubridate'


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

    date, intersect, setdiff, union


  Attaching packages                     tidyverse
1.3.2
 readr   2.1.3      forcats 0.5.2
 stringr 1.5.0
  Conflicts
tidyverse_conflicts()
 lubridate::as.difftime() masks
base::as.difftime()
 readr::col_factor()      masks
scales::col_factor()
 lubridate::date()        masks
base::date()
 purrr::discard()         masks
scales::discard()
 dplyr::filter()          masks
stats::filter()
 stringr::fixed()         masks
recipes::fixed()
```

```
  lubridate::intersect()   masks
base::intersect()
  dplyr::lag()             masks
stats::lag()
  lubridate::setdiff()     masks
base::setdiff()
  readr::spec()            masks
yardstick::spec()
  lubridate::union()       masks
base::union()
```

### 1.1.2 Datos

[3]:
```
indicadores <- read_csv("CU_25_05_06_indicadores_area.csv")
capacidad <- read_csv("CU_25_05_07_01_capacidad.csv")
lista <- read_csv("CU_25_05_07_02_lista_espera.csv")
hospitales <- read_csv("CU_25_05_05_01_hospitales.csv")

lista <- lista |>
  mutate(fecha = as.Date(parse_date_time(paste(ano, semana, 1, sep="/"),'Y/W/
 ↪w')))
```

```
Rows: 11 Columns: 17
  Column specification

Delimiter: ","
chr  (2): id_area, nombre_area
dbl (15): t3_1, t1_1, t2_1, t2_2, t4_1, t4_2, t4_3, t5_1, t6_1, t7_1,
t8_1, …

  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.
Rows: 160 Columns: 4
  Column specification

Delimiter: ","
chr (3): id_area, nombre_area, Especialidad
dbl (1): capacidad

  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.
Rows: 55680 Columns: 12
  Column specification
```

```
Delimiter: ","
chr (4): Hospital, Especialidad, id_area, nombre_area
dbl (8): total_pacientes, media_tiempo_dias, ano, semana, CODCNH,
pacientes,…

 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.
Rows: 87 Columns: 24
 Column specification

Delimiter: ","
chr  (6): Hospital, id_area, nombre_area, Municipio, Clase,
Dependencia
dbl (18): CODCNH, cmunicipio, CAMAS, TAC, RM, GAM, HEM, ASD, LIT, BCO,
ALI, …

 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.
```

### 1.1.3 Spot checking

[4]:
```
h <- "HOSPITAL UNIVERSITARIO LA PAZ"
a <- "05"
e <- "Angiología y Cirugía Vascular"

## Valores perdidos: fill (solo hay una semana)

lzona_esp_1 <- lista |>
  left_join(hospitales) |>
  filter(Especialidad == e,
         id_area == a) |>
  fill(total_pacientes, media_tiempo_dias) |>
  group_by(nombre_area, Especialidad, fecha) |>
  summarise(total_pacientes = sum(total_pacientes, na.rm = TRUE),
            media_tiempo_dias = mean(media_tiempo_dias, na.rm = TRUE))

lzona_esp_1 |> plot_time_series(fecha, total_pacientes)

## XBGoost con series temporales
```

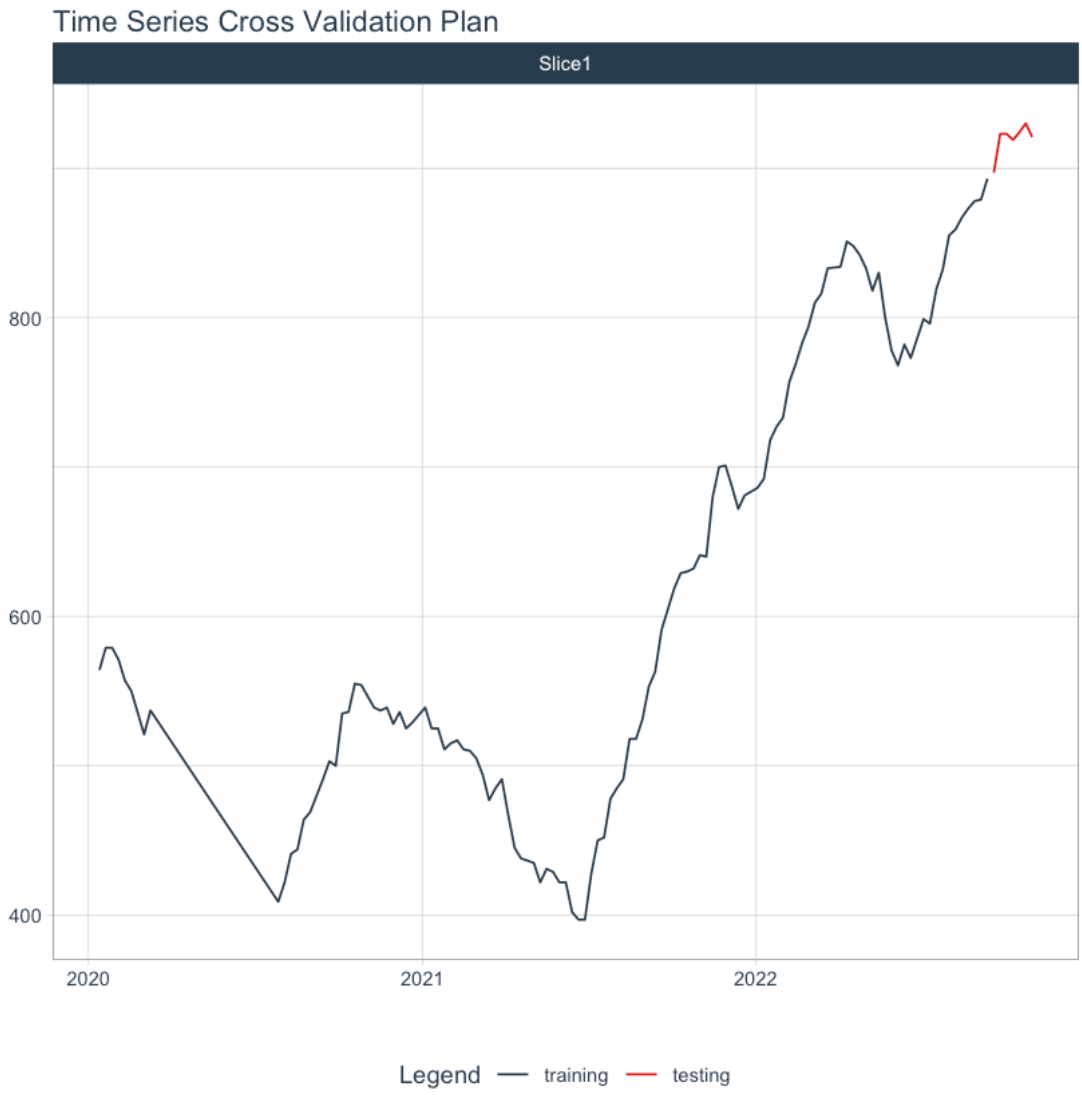
```
Joining with `by = join_by(Hospital, CODCNH, id_area, nombre_area)`
`summarise()` has grouped output by 'nombre_area', 'Especialidad'. You
can
override using the `.groups` argument.
```

```
HTML widgets cannot be represented in plain text (need html)
```

[5]:
```r
## XBGoost con series temporales

## División conjuntos de datos
splits <- lzona_esp_1 %>%
  time_series_split(assess = "3 months", cumulative = TRUE)

## Visualización
splits %>%
  tk_time_series_cv_plan() %>%
  plot_time_series_cv_plan(fecha, total_pacientes, .interactive = FALSE)

## Tidymodels workflow

recipe_spec <- recipe(total_pacientes ~ fecha, training(splits)) %>%
  step_timeseries_signature(fecha) %>%
  step_rm(
    # contains("am.pm"), contains("hour"), contains("minute"),
    #       contains("second"),
    contains("week"),
    contains("xts")) %>%
  step_fourier(fecha, period = 365, K = 5) %>%
  step_dummy(all_nominal())  |>
  step_zv()

recipe_spec %>% prep() %>% juice()
```

```
Groups detected. Removing groups.

Using date_var: fecha
```

| fecha <date> | total_pacientes <dbl> | fecha_index.num <dbl> | fecha_year <int> | fecha_year.iso <int> | fecha_half <int> |
|---|---|---|---|---|---|
| 2020-01-13 | 564 | 1578873600 | 2020 | 2020 | 1 |
| 2020-01-20 | 579 | 1579478400 | 2020 | 2020 | 1 |
| 2020-01-27 | 579 | 1580083200 | 2020 | 2020 | 1 |
| 2020-02-03 | 571 | 1580688000 | 2020 | 2020 | 1 |
| 2020-02-10 | 557 | 1581292800 | 2020 | 2020 | 1 |
| 2020-02-17 | 550 | 1581897600 | 2020 | 2020 | 1 |
| 2020-03-02 | 521 | 1583107200 | 2020 | 2020 | 1 |
| 2020-03-09 | 537 | 1583712000 | 2020 | 2020 | 1 |
| 2020-07-27 | 409 | 1595808000 | 2020 | 2020 | 2 |
| 2020-08-03 | 422 | 1596412800 | 2020 | 2020 | 2 |
| 2020-08-10 | 441 | 1597017600 | 2020 | 2020 | 2 |
| 2020-08-17 | 444 | 1597622400 | 2020 | 2020 | 2 |
| 2020-08-24 | 464 | 1598227200 | 2020 | 2020 | 2 |
| 2020-08-31 | 469 | 1598832000 | 2020 | 2020 | 2 |
| 2020-09-07 | 480 | 1599436800 | 2020 | 2020 | 2 |
| 2020-09-14 | 491 | 1600041600 | 2020 | 2020 | 2 |
| 2020-09-21 | 503 | 1600646400 | 2020 | 2020 | 2 |
| 2020-09-28 | 500 | 1601251200 | 2020 | 2020 | 2 |
| 2020-10-05 | 535 | 1601856000 | 2020 | 2020 | 2 |
| 2020-10-12 | 536 | 1602460800 | 2020 | 2020 | 2 |
| 2020-10-19 | 555 | 1603065600 | 2020 | 2020 | 2 |
| 2020-10-26 | 554 | 1603670400 | 2020 | 2020 | 2 |
| 2020-11-09 | 539 | 1604880000 | 2020 | 2020 | 2 |
| 2020-11-16 | 537 | 1605484800 | 2020 | 2020 | 2 |
| 2020-11-23 | 539 | 1606089600 | 2020 | 2020 | 2 |
| 2020-11-30 | 528 | 1606694400 | 2020 | 2020 | 2 |
| 2020-12-07 | 536 | 1607299200 | 2020 | 2020 | 2 |
| 2020-12-14 | 525 | 1607904000 | 2020 | 2020 | 2 |
| 2020-12-21 | 529 | 1608508800 | 2020 | 2020 | 2 |
| A tibble: 113 x 46 2021-01-04 | 539 | 1609718400 | 2021 | 2021 | 1 |
| ... | ... | ... | ... | ... | ... |
| 2022-02-07 | 757 | 1644192000 | 2022 | 2022 | 1 |
| 2022-02-14 | 769 | 1644796800 | 2022 | 2022 | 1 |
| 2022-02-21 | 783 | 1645401600 | 2022 | 2022 | 1 |
| 2022-02-28 | 794 | 1646006400 | 2022 | 2022 | 1 |
| 2022-03-07 | 810 | 1646611200 | 2022 | 2022 | 1 |
| 2022-03-14 | 816 | 1647216000 | 2022 | 2022 | 1 |
| 2022-03-21 | 833 | 1647820800 | 2022 | 2022 | 1 |
| 2022-04-04 | 834 | 1649030400 | 2022 | 2022 | 1 |
| 2022-04-11 | 851 | 1649635200 | 2022 | 2022 | 1 |
| 2022-04-18 | 848 | 1650240000 | 2022 | 2022 | 1 |
| 2022-04-25 | 842 | 1650844800 | 2022 | 2022 | 1 |
| 2022-05-02 | 833 | 1651449600 | 2022 | 2022 | 1 |
| 2022-05-09 | 818 | 1652054400 | 2022 | 2022 | 1 |
| 2022-05-16 | 830 | 1652659200 | 2022 | 2022 | 1 |
| 2022-05-23 | 800 | 1653264000 | 2022 | 2022 | 1 |
| 2022-05-30 | 778 | 1653868800 | 2022 | 2022 | 1 |
| 2022-06-06 | 768 | 1654473600 | 2022 | 2022 | 1 |
| 2022-06-13 | 782 | 1655078400 | 2022 | 2022 | 1 |
| 2022-06-20 | 773 | 1655683200 | 2022 | 2022 | 1 |
| 2022-07-04 | 799 | 1656892800 | 2022 | 2022 | 2 |




[6]:
```
model_spec_glmnet <- linear_reg(penalty = 0.01, mixture = 0.5) %>%
  set_engine("glmnet")
```

[7]:
```
workflow_fit_glmnet <- workflow() %>%
  add_model(model_spec_glmnet) %>%
  add_recipe(recipe_spec %>% step_rm(fecha)) %>%
  fit(training(splits))
```

### 1.1.4 Model fitting

[8]:
```
lzona_esp <- lista |>
  left_join(hospitales) |>
  # filter(Especialidad == e,
  #        id_area == a) |>
  fill(total_pacientes, media_tiempo_dias) |>
  group_by(nombre_area, Especialidad, fecha) |>
  summarise(total_pacientes = sum(total_pacientes, na.rm = TRUE),
            media_tiempo_dias = mean(media_tiempo_dias, na.rm = TRUE))

dfs <- split(lzona_esp, ~nombre_area + Especialidad)
```

```
Joining with `by = join_by(Hospital, CODCNH, id_area, nombre_area)`
`summarise()` has grouped output by 'nombre_area', 'Especialidad'. You
can
override using the `.groups` argument.
```

[9]:
```
res_pacientes <- map(dfs, function(x){
  recipe_spec <- recipe(total_pacientes ~ fecha, x) %>%
    step_timeseries_signature(fecha) %>%
    step_rm(
      # contains("am.pm"), contains("hour"), contains("minute"),
      #       contains("second"),
      contains("week"),
      contains("xts")) %>%
    step_fourier(fecha, period = 365, K = 5) %>%
    step_dummy(all_nominal())  |>
    step_zv()
  recipe_spec %>% prep() %>% juice()
  model_spec_prophet_boost <- prophet_boost(seasonality_yearly = TRUE,
                                            seasonality_daily = FALSE,
                                            seasonality_weekly = TRUE) %>%
    set_engine("prophet_xgboost")
  workflow_fit_prophet_boost <- workflow() %>%
    add_model(model_spec_prophet_boost) %>%
    add_recipe(recipe_spec) %>%
    fit(x)
  model_table <- modeltime_table(
    workflow_fit_prophet_boost
  )
  calibration_table <- model_table %>%
    modeltime_calibrate(x)
})
```

[10]:
```
res_tiempo <- map(dfs, function(x){
  recipe_spec <- recipe(media_tiempo_dias ~ fecha, x) %>%
    step_timeseries_signature(fecha) %>%
```

```
    step_rm(
      # contains("am.pm"), contains("hour"), contains("minute"),
      #       contains("second"),
      contains("week"),
      contains("xts")) %>%
    step_fourier(fecha, period = 365, K = 5) %>%
    step_dummy(all_nominal())  |>
    step_zv()
  recipe_spec %>% prep() %>% juice()
  model_spec_prophet_boost <- prophet_boost(seasonality_yearly = TRUE,
                                            seasonality_daily = FALSE,
                                            seasonality_weekly = TRUE) %>%
    set_engine("prophet_xgboost")
  workflow_fit_prophet_boost <- workflow() %>%
    add_model(model_spec_prophet_boost) %>%
    add_recipe(recipe_spec) %>%
    fit(x)
  model_table <- modeltime_table(
    workflow_fit_prophet_boost
  )
  calibration_table <- model_table %>%
    modeltime_calibrate(x)
})
```

[11]:
```
model_spec_prophet_boost <- prophet_boost(seasonality_yearly = TRUE,
                                          seasonality_weekly = TRUE,
                                          seasonality_daily = TRUE) %>%
  set_engine("prophet_xgboost")
```

[13]:
```
workflow_fit_prophet_boost <- workflow() %>%
  add_model(model_spec_prophet_boost) %>%
  add_recipe(recipe_spec) %>%
  fit(training(splits))

#workflow_fit_prophet_boost
```

[15]:
```
model_table <- modeltime_table(
  # model_fit_arima,
  # model_fit_prophet,
  workflow_fit_glmnet,
  # workflow_fit_rf,
  workflow_fit_prophet_boost
)
glimpse(model_table)
```

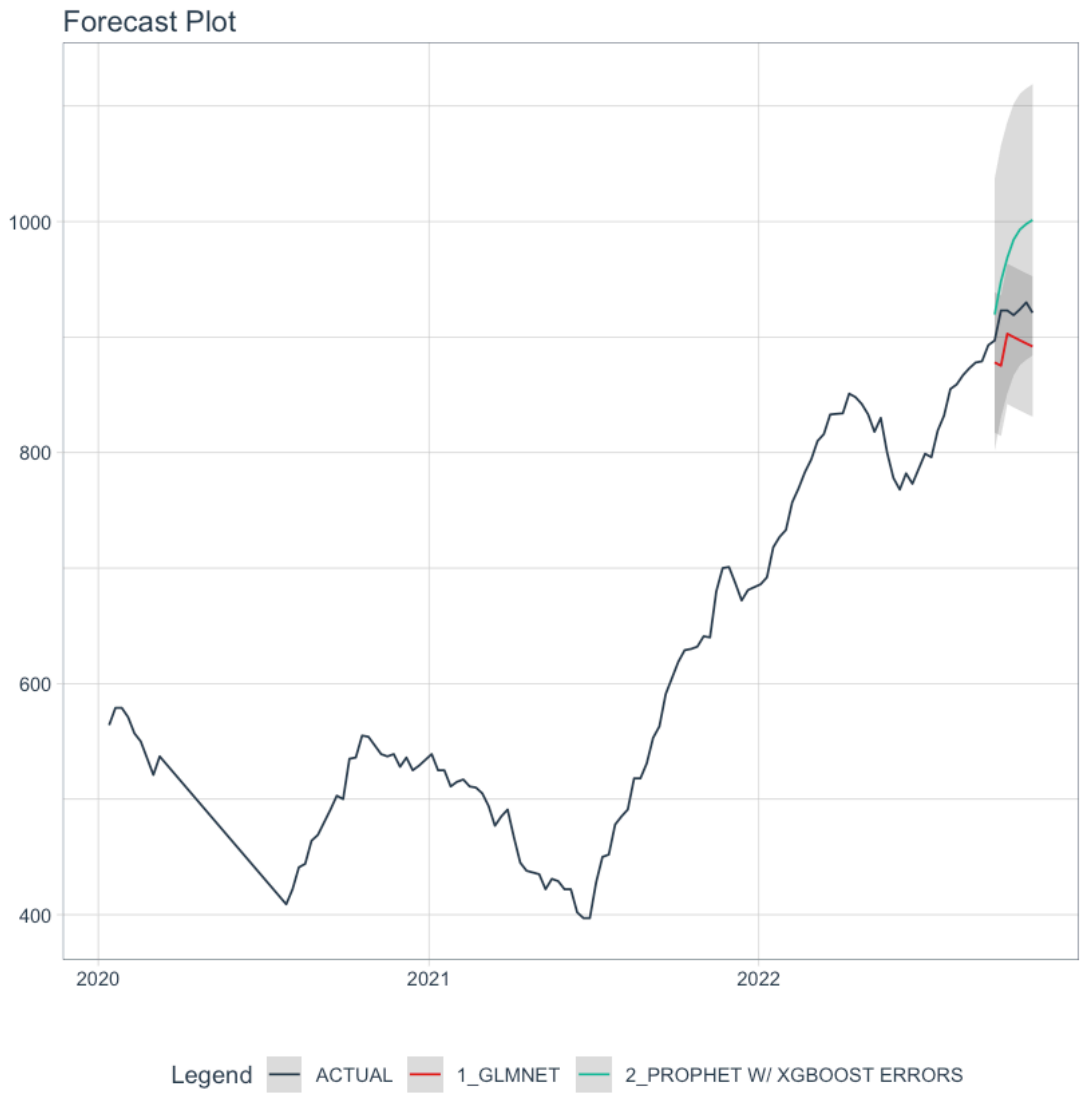
```
Rows: 2
Columns: 3
$ .model_id   <int> 1, 2
```

```
$ .model      <list> [fecha, total_pacientes, date, double,
numeric, predictor,…
$ .model_desc <chr> "GLMNET", "PROPHET W/ XGBOOST ERRORS"
```

[22]:
```
calibration_table <- model_table %>%
  modeltime_calibrate(testing(splits))

calibration_table |> glimpse()

calibration_table %>%
  modeltime_forecast(actual_data = lzona_esp_1) %>%
  plot_modeltime_forecast(.interactive = FALSE)
```

```
Rows: 2
Columns: 5
$ .model_id         <int> 1, 2
$ .model            <list> [fecha, total_pacientes, date,
double, numeric, pred…
$ .model_desc       <chr> "GLMNET", "PROPHET W/ XGBOOST
ERRORS"
$ .type             <chr> "Test", "Test"
$ .calibration_data <list> [<tbl_df[7 x 4]>], [<tbl_df[7 x
4]>]
```

```
Using '.calibration_data' to forecast.

Warning message in max(ids, na.rm = TRUE):
"ningun argumento finito para max; retornando -Inf"
```

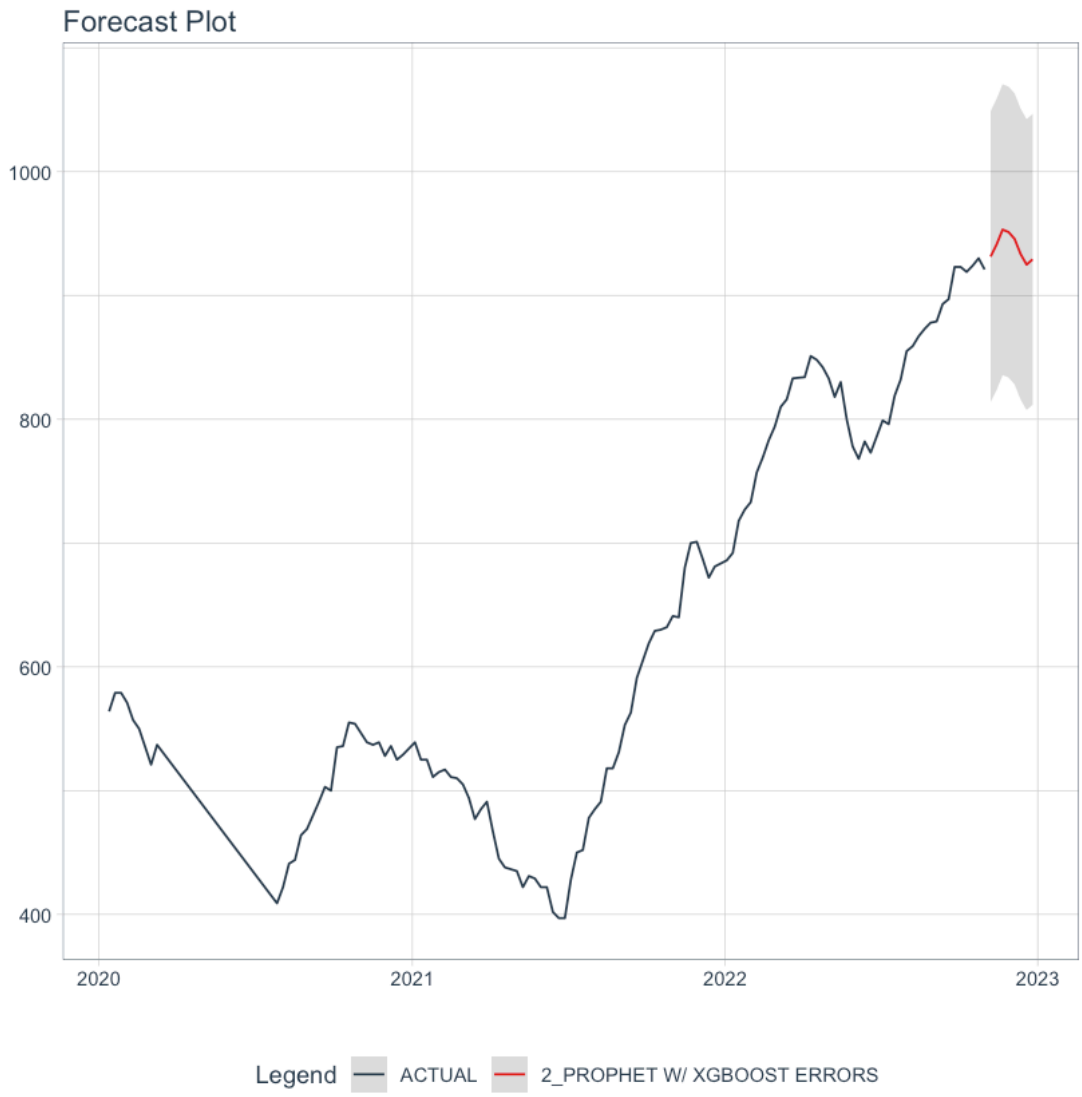



[23]:
```
calibration_table %>%
  modeltime_accuracy() %>%
  table_modeltime_accuracy(.interactive = FALSE) |>
  glimpse()


## Refit and forecast
calibration_table |>
  filter(.model_id == 2) |>
  modeltime_refit(lzona_esp_1) %>%
  modeltime_forecast(h = "8 weeks", actual_data = lzona_esp_1) %>%
  plot_modeltime_forecast(.interactive = FALSE)
```

```
List of 17
 $ _data         : tibble [2 × 9] (S3: tbl_df/tbl/data.frame)
  ..$ .model_id  : int [1:2] 1 2
  ..$ .model_desc: chr [1:2] "GLMNET" "PROPHET W/ XGBOOST ERRORS"
  ..$ .type      : chr [1:2] "Test" "Test"
  ..$ mae        : num [1:2] 28.2 53.7
  ..$ mape       : num [1:2] 3.06 5.83
  ..$ mase       : num [1:2] 3.38 6.44
  ..$ smape      : num [1:2] 3.12 5.64
  ..$ rmse       : num [1:2] 29.9 57.7
  ..$ rsq        : num [1:2] 0.25 0.61
 $ _boxhead      : tibble [9 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ var         : chr [1:9] ".model_id" ".model_desc" ".type" "mae" …
  ..$ type        : chr [1:9] "default" "default" "default" "default" …
  ..$ column_label:List of 9
  ..$ column_align: chr [1:9] "right" "left" "left" "right" …
  ..$ column_width:List of 9
  ..$ hidden_px   :List of 9
 $ _stub_df      : tibble [2 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ rownum_i         : int [1:2] 1 2
  ..$ row_id           : chr [1:2] NA NA
  ..$ group_id         : chr [1:2] NA NA
  ..$ group_label      :List of 2
  ..$ indent           : chr [1:2] NA NA
  ..$ built_group_label: chr [1:2] NA NA
 $ _row_groups   : chr(0)
 $ _heading      :List of 3
  ..$ title    : chr "Accuracy Table"
  ..$ subtitle : NULL
  ..$ preheader: NULL
 $ _spanners     : tibble [0 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ vars         : list()
  ..$ spanner_label: list()
  ..$ spanner_id   : chr(0)
  ..$ spanner_level: int(0)
  ..$ gather       : logi(0)
  ..$ built        : chr(0)
 $ _stubhead     :List of 1
  ..$ label: NULL
 $ _footnotes    : tibble [0 × 8] (S3: tbl_df/tbl/data.frame)
  ..$ locname  : chr(0)
  ..$ grpname  : chr(0)
  ..$ colname  : chr(0)
  ..$ locnum   : num(0)
  ..$ rownum   : int(0)
  ..$ colnum   : int(0)
  ..$ footnotes: list()
  ..$ placement: chr(0)
```

```
 $ _source_notes : list()
 $ _formats      : list()
 $ _substitutions: list()
 $ _styles       : tibble [0 × 7] (S3: tbl_df/tbl/data.frame)
  ..$ locname: chr(0)
  ..$ grpname: chr(0)
  ..$ colname: chr(0)
  ..$ locnum : num(0)
  ..$ rownum : int(0)
  ..$ colnum : int(0)
  ..$ styles : list()
 $ _summary      : list()
 $ _options      : tibble [189 × 5] (S3: tbl_df/tbl/data.frame)
  ..$ parameter: chr [1:189] "table_id" "table_caption" "table_width"
"table_layout" …
  ..$ scss     : logi [1:189] FALSE FALSE TRUE TRUE TRUE TRUE …
  ..$ category : chr [1:189] "table" "table" "table" "table" …
  ..$ type     : chr [1:189] "value" "value" "px" "value" …
  ..$ value    :List of 189
 $ _transforms   : list()
 $ _locale       :List of 1
  ..$ locale: NULL
 $ _has_built    : logi FALSE
 - attr(*, "class")= chr [1:2] "gt_tbl" "list"
```

```
Warning message in max(ids, na.rm = TRUE):
"ningun argumento finito para max; retornando -Inf"
```




### 1.1.5 Predicción

[24]:
```
## Predicción
prediccion <- res_tiempo$`Centro-Norte.Angiología y Cirugía Vascular` |>
  modeltime_forecast(h = 8, actual_data = dfs$`Centro-Norte.Angiología y␣
 ↪Cirugía Vascular`)

## Visualización
dfs$`Centro-Norte.Angiología y Cirugía Vascular` |>
  plot_time_series(fecha, total_pacientes)
prediccion |> plot_modeltime_forecast()

## Guardar modelos
```

```r
write_rds(res_pacientes, "modelos_pacientes_xgboost.rds")
write_rds(res_tiempo, "modelos_tiempo_xgboost.rds")
```

```
HTML widgets cannot be represented in plain text (need html)
```

```
HTML widgets cannot be represented in plain text (need html)
```