Trabajo Fin de Máster: Smart Bike - Sevilla.
Anexo 3. Modelos predictivos.

Máster en Data Science y Big Data - Universidad de Sevilla, 2016/2017.

Jerónimo Carranza Carranza

1 de marzo de 2018

Índice

Índice de cuadros

Índice de figuras

```
library(RPostgreSQL)
# library(tidyverse)
library(tidyr)
# library(dtplyr) library(dbplyr)
library(knitr)
library(dplyr)
library(sp)
library(sf)
library(ggplot2)
library(ggcorrplot)
library(ggspatial)
library(lubridate)
library(scales)
# library(factoextra) library(reshape2) library(igraph) library(ggraph)
# library(ggdendro)

library(glmnet)
```

```
Bp = readRDS(file = "../Bp.rds")
Bp = as.tibble(Bp)
```

```
load(file = "../Bp_co.RData")
Bp_co = as.tibble(Bp_co)
```

1. Introducción

Todos los modelos de regresión analizados consideran que para cada estación (i) en un momento determinado (t), el número de bicicletas disponibles (Y(i,t)) es una función lineal de:

- los valores de dicha variable en esa estación en momentos anteriores:

  $Y(i, t-15min), Y(i, t-30min), Y(i, t-1h), Y(i, t-4h), Y(i, t-8h), Y(i, y-24h)$

- los valores de dicha variable en la estación más cercana a ella (j) en momentos anteriores:

  $Y(j, t-15min), Y(j, t-30min), Y(j, t-1h), Y(j, t-4h), Y(j, t-8h), Y(j, y-24h)$

- el día de la semana que es t $DSEM$,
- la hora del día del momento t $HORA$,
- si es día festivo $FEST$,
- temperatura máxima del día $TMAX$,
- temperatura mínima del día $TMIN$
- precipitación total del día P

Dieciocho variables regresoras que, al convertir en binaria $DSEM$, con siete niveles, pasan a ser 24.

Para cada momento (t) se realizará la predicción para t, t+15min, t+30min, t+1h, t+4h, t+8h y t+24h.

2. Muestras de entrenamiento y test

Se toman para el modelado sólo los casos completos existentes en el conjunto de datos, lo que supone 52543 casos para 1834 variables (originales y retardadas). No se utilizan los datos de la estación 109, que sólo

dispone de datos durante los tres primeros meses de registro.

Se obtienen muestras relativamente pequeñas para training (14 %) y test (6 %) a partir de una división del conjunto de datos con fecha de corte 2016-09-15, que deja aproximadamente el 70 % de las observaciones a su izquierda (anteriores) y aproximadamente el 30 % a su derecha (posteriores). Se garantiza así que toda la muestra test sea posterior a la muestra de entrenamiento.

```
Bp_co$mie = ifelse(Bp_co$dsem == "mié", 1, 0)
Bp_co$sab = ifelse(Bp_co$dsem == "sáb", 1, 0)

Bp_co$fsof = ifelse(Bp_co$dsem %in% c("sáb", "dom") | Bp_co$fest == 1,
    1, 0)
```

```
set.seed(12345)
```

```
Bp_co.mini.train = Bp_co %>% filter(fecha < "2016-09-15") %>% sample_frac(size = 0.2)

Bp_co.mini.test = Bp_co %>% filter(fecha >= "2016-09-15") %>% sample_frac(size = 0.2)

Bp_co.mini.train = as.tibble(Bp_co.mini.train)
Bp_co.mini.test = as.tibble(Bp_co.mini.test)

Bp_co.mini.train$fsof = ifelse(Bp_co.mini.train$dsem %in% c("sáb", "dom") |
    Bp_co.mini.train$fest == 1, 1, 0)

Bp_co.mini.test$fsof = ifelse(Bp_co.mini.test$dsem %in% c("sáb", "dom") |
    Bp_co.mini.test$fest == 1, 1, 0)
```

```
modelos = tibble(id = 0, modelo = "", vi = "", vj = "", lambda = 0, a0 = 0,
    df = 0, r2 = 0, RMSE = 0, R2test = 0)
betas = tibble(id = 0, modelo = "", vi = "", vj = "", nom = as.list(NULL),
    beta = as.list(NULL))
residuos = tibble(id = 0, modelo = "", vi = "", residuo = as.list(NULL))
```

3. Modelo de Regresión con regularización. Elasticnet.

Se utilizan modelos con optimización de parámetros por validación cruzada según implementa el paquete de R *glmnet*.

3.1. Modelo

```
varDTF <- Bp_co.mini.train %>% dplyr::select(
  one_of('hora','lun','mar','mie','jue','vie','sab','dom','fest','fsof',
         'p','tmax','tmin'))

testDTF <- Bp_co.mini.test %>% dplyr::select(
  one_of('hora','lun','mar','mie','jue','vie','sab','dom','fest','fsof',
         'p','tmax','tmin'))


for (i in c(1:108,110:260)){
  vari = paste0('b',i)
  varY = Bp_co.mini.train %>% dplyr::select(vari)
```

```
  testY = Bp_co.mini.test %>% dplyr::select(vari)

  varXi <- Bp_co.mini.train %>% dplyr::select(ends_with(vari)) %>%
                  dplyr::select(starts_with('l'))

  testXi <- Bp_co.mini.test %>% dplyr::select(ends_with(vari)) %>%
                  dplyr::select(starts_with('l'))

  j = seviesta_nearest[i,2]
  varj = paste0('b',j)
  varXj <- Bp_co.mini.train %>% dplyr::select(ends_with(varj)) %>%
                  dplyr::select(starts_with('l'))
  testXj <- Bp_co.mini.test %>% dplyr::select(ends_with(varj)) %>%
                  dplyr::select(starts_with('l'))

  varX <- bind_cols(varDTF,varXi,varXj)
  testX <- bind_cols(testDTF,testXi,testXj)

# Modelo completo
  model.glmnet = cv.glmnet(as.matrix.data.frame(varX),
                           as.matrix.data.frame(varY), alpha=0.5)
  idx.opt = which(model.glmnet$lambda==model.glmnet$lambda.1se)

  pred.glmnet = predict.cv.glmnet(model.glmnet,
                                  as.matrix.data.frame(testX))

  RMSE = sqrt(mean((testY - pred.glmnet)^2))
  R2test = cor(testY,pred.glmnet)^2

  id = nrow(modelos) + 1
  modelos <- add_row(modelos
    , id = id
    , modelo = 'GLMNET0'
    , vi = vari
    , vj = varj
    , lambda = model.glmnet$glmnet.fit$lambda[idx.opt]
    , a0 = model.glmnet$glmnet.fit$a0[idx.opt]

    , df = model.glmnet$glmnet.fit$df[idx.opt]
    , r2 = model.glmnet$glmnet.fit$dev.ratio[idx.opt]
    , RMSE = RMSE
    , R2test = as.numeric(R2test)
  )
  betas <- add_row(betas, id = id, modelo = 'GLMNET0',
                   vi = vari, vj = varj,
                   nom = names(model.glmnet$glmnet.fit$beta[,idx.opt]),
                   beta = model.glmnet$glmnet.fit$beta[,idx.opt]
  )
  residuos <- add_row(residuos, id = id, modelo = 'GLMNET0',
                   vi = vari, residuo = (testY - pred.glmnet)[,1]
  )

# Modelos parciales
```

```
  for (k in 0:5) {
      ki = 14+k #(l15mbi,l30mbi,l1hbi,l4hbi,l8hbi,l24hbi)
      kj = 20+k #(l15mbj,l30mbj,l1hbj,l4hbj,l8hbj,l24hbj)

      varXk = varX[,-c(14:ki,20:kj)]
      testXk = testX[,-c(14:ki,20:kj)]

      model.glmnet = cv.glmnet(as.matrix.data.frame(varXk),
                               as.matrix.data.frame(varY), alpha=0.5)
      idx.opt = which(model.glmnet$lambda==model.glmnet$lambda.1se)

      pred.glmnet = predict.cv.glmnet(model.glmnet,
                                      as.matrix.data.frame(testXk))

      RMSE = sqrt(mean((testY - pred.glmnet)^2))
      R2test   = cor(testY,pred.glmnet)^2

      id = nrow(modelos) + 1
      modelos <- add_row(modelos
        , id = id
        , modelo = paste0('GLMNET',k+1)
        , vi = vari
        , vj = varj
        , lambda = model.glmnet$glmnet.fit$lambda[idx.opt]
        , a0 = model.glmnet$glmnet.fit$a0[idx.opt]

        , df = model.glmnet$glmnet.fit$df[idx.opt]
        , r2 = model.glmnet$glmnet.fit$dev.ratio[idx.opt]
        , RMSE = RMSE
        , R2test = as.numeric(R2test)
      )
      betas <- add_row(betas, id = id, modelo = paste0('GLMNET',k+1),
                       vi = vari, vj = varj,
                       nom = names(model.glmnet$glmnet.fit$beta[,idx.opt]),
                       beta = model.glmnet$glmnet.fit$beta[,idx.opt]
      )
      residuos <- add_row(residuos, id = id, modelo = paste0('GLMNET',k+1),
                    vi = vari, residuo = (testY - pred.glmnet)[,1]
      )
  }
}
```

3.2. Resultados

```
# modelos=modelos[-1,]
modelos = as.data.frame(modelos)
betas = as.data.frame(betas)
residuos = as.data.frame(residuos)
```

```
load(file = "modelos.RData")
load(file = "betas.RData")
load(file = "residuos.RData")
```

Los dataframes *modelos*, *betas* y *residuos* recogen la información derivada del ajuste, entrenamiento y testeo de los distintos modelos. Las cabeceras de dichos dataframes dan cuenta del modo en que se han organizado.

```
head(modelos)
```

```
##   id  modelo vi   vj    lambda         a0 df        r2      RMSE
## 1  2 GLMNET0 b1 b179 0.2931581  0.6973223  4 0.9777344  5.207738
## 2  3 GLMNET1 b1 b179 0.5555811  1.2567319  5 0.9540914  7.321472
## 3  4 GLMNET2 b1 b179 1.3720174  2.9205337  4 0.9043455 10.114414
## 4  5 GLMNET3 b1 b179 1.7024080  5.6792678  7 0.6711944 18.023582
## 5  6 GLMNET4 b1 b179 2.5113937  9.7058241  6 0.4913999 21.980414
## 6  7 GLMNET5 b1 b179 1.8018208 12.4920250  9 0.3386743 25.230830
##      R2test
## 1 0.9680338
## 2 0.9368288
## 3 0.8804563
## 4 0.6169176
## 5 0.4312904
## 6 0.2498336
```

```
head(betas)
```

```
##   id  modelo vi   vj  nom beta
## 1  2 GLMNET0 b1 b179 hora    0
## 2  2 GLMNET0 b1 b179  lun    0
## 3  2 GLMNET0 b1 b179  mar    0
## 4  2 GLMNET0 b1 b179  mie    0
## 5  2 GLMNET0 b1 b179  jue    0
## 6  2 GLMNET0 b1 b179  vie    0
```

```
head(residuos)
```

```
## # A tibble: 6 x 4
##      id modelo  vi    residuo
##   <dbl> <chr>   <chr> <list>
## 1  2.00 GLMNET0 b1    <dbl [1]>
## 2  2.00 GLMNET0 b1    <dbl [1]>
## 3  2.00 GLMNET0 b1    <dbl [1]>
## 4  2.00 GLMNET0 b1    <dbl [1]>
## 5  2.00 GLMNET0 b1    <dbl [1]>
## 6  2.00 GLMNET0 b1    <dbl [1]>
```

```
summary(modelos)
```

```
##        id          modelo              vi
##  Min.   :   2   Length:1813        Length:1813
##  1st Qu.: 455   Class :character   Class :character
##  Median : 908   Mode  :character   Mode  :character
##  Mean   : 908
##  3rd Qu.:1361
##  Max.   :1814
##
##       vj                lambda              a0
##  Length:1813        Min.   :0.03992   Min.   :-33.2090
##  Class :character   1st Qu.:0.49386   1st Qu.:  0.9763
##  Mode  :character   Median :1.02690   Median :  4.1605
##                     Mean   :1.15605   Mean   : 13.1218
```

```
##                     3rd Qu.:1.68416   3rd Qu.: 19.9352
##                     Max.   :4.54239   Max.   :116.1219
##
##        df               r2              RMSE
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.2289
##  1st Qu.: 5.000   1st Qu.:0.3363   1st Qu.: 7.3979
##  Median : 7.000   Median :0.7843   Median :15.5110
##  Mean   : 7.334   Mean   :0.6482   Mean   :17.5913
##  3rd Qu.: 9.000   3rd Qu.:0.9519   3rd Qu.:26.6529
##  Max.   :18.000   Max.   :0.9990   Max.   :69.3278
##
##      R2test
##  Min.   :0.000005
##  1st Qu.:0.117016
##  Median :0.677502
##  Mean   :0.548565
##  3rd Qu.:0.926816
##  Max.   :0.998700
##  NA's   :6
```

La tabla *modelos* incorpora tres indicadores de la bondad de ajuste, uno referido a la etapa de entrenamiento (r2) y otros dos (RMSE y R2test) calculados en base al contraste de las predicciones con los valores reales en el conjunto test.

En la tablas siguientes se muestran los indicadores señalados tanto para el conjunto de todos los modelos como por tipo de modelo.

```
modelos %>% dplyr::select("r2", "R2test", "RMSE") %>% summarise_all(funs(median,
    mean), na.rm = TRUE) %>% kable(caption = "Bondad de ajuste global de los modelos.",
    digits = 4)
```

Cuadro 1: Bondad de ajuste global de los modelos.

r2_median	R2test_median	RMSE_median	r2_mean	R2test_mean	RMSE_mean
0.7843	0.6775	15.511	0.6482	0.5486	17.5913

```
modelos %>% dplyr::select("modelo", "r2", "R2test", "RMSE") %>% group_by(modelo) %>%
    summarise_all(funs(median, mean), na.rm = TRUE) %>% kable(caption = "Bondad de ajuste por tipo de m
    digits = 4)
```

Cuadro 2: Bondad de ajuste por tipo de modelo.

modelo	r2_median	R2test_median	RMSE_median	r2_mean	R2test_mean	RMSE_mean
GLMNET0	0.9774	0.9677	5.1183	0.9741	0.9623	5.0735
GLMNET1	0.9546	0.9320	7.3979	0.9490	0.9234	7.2812
GLMNET2	0.9066	0.8551	10.7559	0.8992	0.8463	10.4202
GLMNET3	0.6584	0.4512	20.2921	0.6613	0.4976	19.3752
GLMNET4	0.4778	0.2180	24.7588	0.5115	0.3212	23.1190
GLMNET5	0.3419	0.1132	26.5877	0.4090	0.2304	25.3272
GLMNET6	0.0971	0.0262	31.2985	0.1329	0.0472	32.5427

Los resultados muestran un buen ajuste para el conjunto de los modelos, con R2test que para más de la mitad de ellos superan el .67. Pero más importante es que RMSE, raíz del error medio cuadrático, tiene un

valor muy bajo, 17.59 en media y mediana de 15.51. Hay que tener en cuenta que RMSE se mide en las mismas unidades que la variable objetivo de predicción, en nuestro caso porcentaje de bicicletas.

La bondad de los modelos por tipo es tambien bastante buena, si bien, como era esperable con un notable incremento del error a medida que se dispone de menor información para la predicción. En los modelos con disponibilidad de información muy reciente (15min), la bondad del ajuste, se dispara con R2 muy próximo a 1 y RMSE entorno a 5. Para un horizonte de predicción de 4h, RMSE se sitúa entorno a 20, con 24h en 26 y para horizontes más lejanos, el RMSE está entorno a 32.

La figura siguiente muestra el progresivo incremento de RMSE a medida que se alarga el horizonte de predicción y han de utilizarse modelos con menor información reciente.

```
modelos %>% dplyr::select("modelo", "r2", "R2test", "RMSE") %>% ggplot() +
    geom_boxplot(aes(modelo, RMSE, group = modelo), colour = "orange") +
    annotate("text", x = 1, y = -2, label = "15min", size = 3, color = "blue") +
    annotate("text", x = 2, y = -2, label = "30min", size = 3, color = "blue") +
    annotate("text", x = 3, y = -2, label = "1h", size = 3, color = "blue") +
    annotate("text", x = 4, y = -2, label = "4h", size = 3, color = "blue") +
    annotate("text", x = 5, y = -2, label = "8h", size = 3, color = "blue") +
    annotate("text", x = 6, y = -2, label = "24h", size = 3, color = "blue") +
    labs(x = "Tipo", y = "RMSE")
```

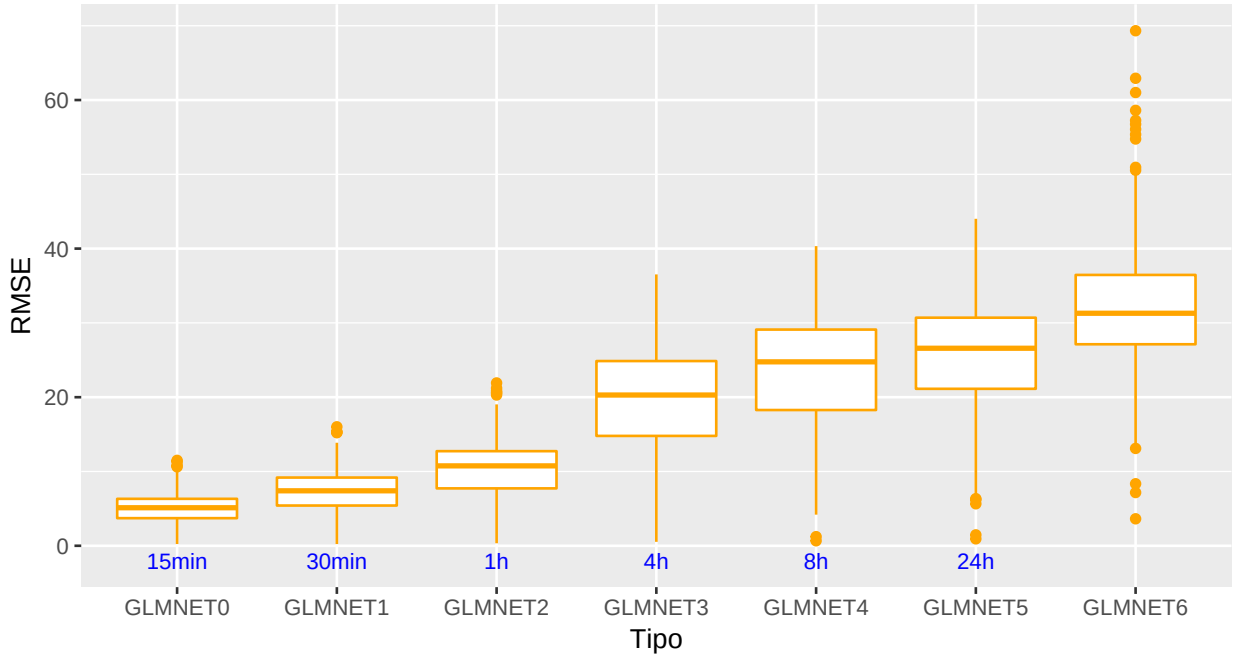

Figura 1: Bondad de ajuste. Raíz del error cuadrático medio (RMSE) por tipo de modelo.

Para cada estación se han estimado siete modelos, la figura siguiente muestra para cada uno de ellos su error (RMSE).

```
modelos = modelos %>% mutate(num = substr(vi, 2, 10))

modelos %>% group_by(modelo, num) %>% summarise(RMSE = mean(RMSE), na.rm = TRUE) %>%
    ggplot() + geom_tile(aes(x = as.numeric(num), y = modelo, fill = RMSE)) +
    scale_fill_gradientn(colors = c("cyan", "green", "yellow", "red")) +
```

```
    # scale_x_discrete(breaks = c(50,100,200,250))+
labs(y = "Tipo de modelo", x = "Estación")
```

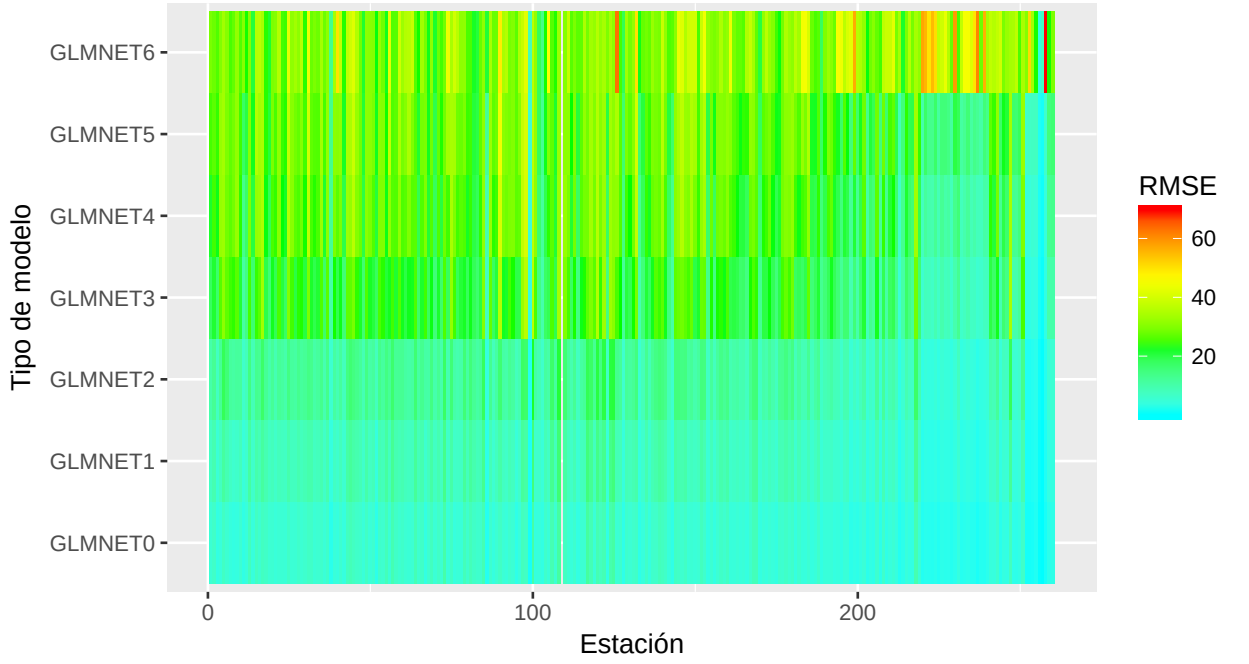

Figura 2: Errores (RMSE) por estación y tipo de modelo

La figura siguiente muestra la distribución de residuos por tipo de modelo.

```
# residuos_sample = as.data.frame(residuos %>% group_by(modelo) %>%
# sample_n(2000))

ggplot(residuos) + geom_histogram(aes(as.numeric(residuo)), color = "orange",
    alpha = 0.2) + labs(x = "Residuo", y = "Frecuencia") + facet_grid(~modelo)
```

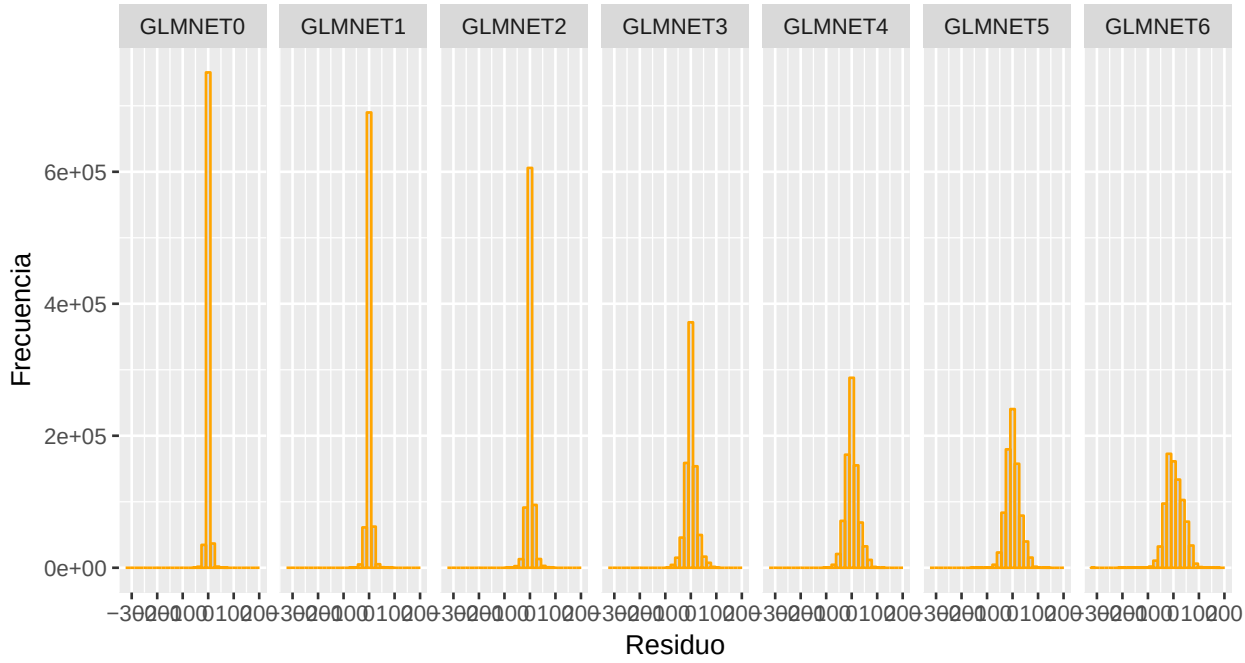

Figura 3: Distribución de residuos por tipo de modelo.

La importancia relativa de cada uno de los regresores en el conjunto de modelos estimados se muestra en la figura siguiente, en la que se representa el número de modelos en los que dicho regresor aparece como significativo, según tipo de modelo.

```
# Creamos la variable regres que contiene el nombre genérico del
# regresor en vez del específico (nom) que incluye la estación objetivo
# o la proxy.

# betas$regres = ifelse(endsWith(betas$nom,betas$vi),
# paste0(strsplit(betas$nom,betas$vi)[[1]],'i'),
# ifelse(endsWith(betas$nom,betas$vj),
# paste0(strsplit(betas$nom,betas$vj)[[1]],'j'), betas$nom ) )

frnom = function(a, b, c) {
    ifelse(endsWith(c, a), return(paste0(strsplit(c, a)[[1]], "i")), ifelse(endsWith(c,
        b), return(paste0(strsplit(c, b)[[1]], "j")), return(c)))
}
```

```
betas$regres = mapply(frnom, betas$vi, betas$vj, betas$nom)
```

```
data_regres = as.data.frame(betas %>% filter(beta != 0) %>% group_by(regres,
    modelo) %>% count())

# data_max0 = data_regres %>% filter(modelo == 'GLMNET0') %>%
# group_by(regres) %>% summarise(max=max(n))

# data_regres$regres = factor(data_regres$regres, levels =
# (data_regres$regres)[order(data_max0$max,data_max0$regres)])
```

```
# regres_list = as.list(data_max0 %>% arrange(max,regres) %>%
# select(regres))

ggplot(data_regres) + geom_tile(aes(x = modelo, y = regres, fill = n)) +
    geom_text(aes(x = modelo, y = regres, label = n), color = "blue", size = 2.5,
        alpha = 0.7) + scale_y_discrete(limits = c("fest", "p", "jue",
    "l30mj", "sab", "vie", "mie", "mar", "lun", "l1hj", "dom", "l8hj",
    "fsof", "tmax", "tmin", "l24hj", "l4hj", "hora", "l15mj", "l8hi", "l4hi",
    "l1hi", "l24hi", "l30mi", "l15mi")) + scale_fill_gradientn(colors = c("cyan",
    "green", "yellow", "red")) + labs(x = "Tipo de modelo", y = "Regresor")
```

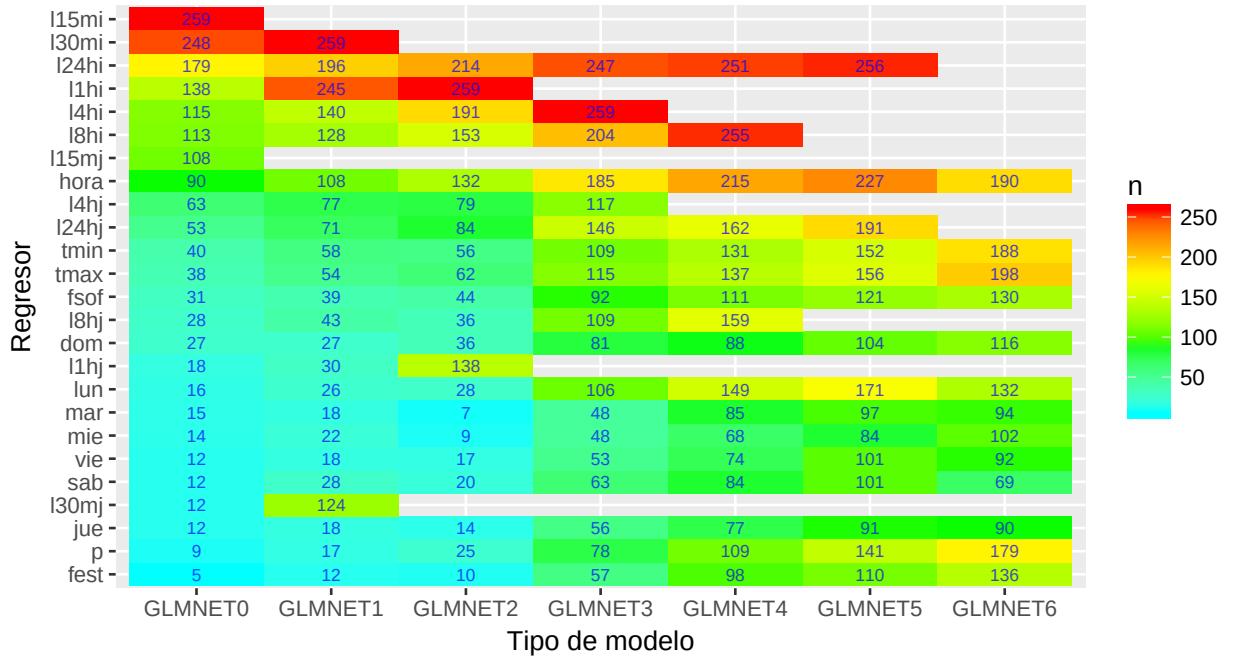

Figura 4: Frequencia de modelos con regresor significativo por tipo de modelo.

Vemos que para cada uno de los tipos de modelo, los regresores de retardo propios (de la misma estación) más recientes disponibles son los que más importancia tienen, apareciendo en la totalidad de estaciones. Esto es cierto para horizontes de predicción de hasta 4h (GLMNET3). En el caso de un horizonte de 8h (GLMNET4), tiene más importancia el retardo propio de 24h que el mismo de 8h.

El retardo propio de 24h tiene una gran importancia apareciendo en tercer lugar con 179 estaciones para modelos completos (GLMNET0).

Los regresores de retardo de vecino más próximo (j) tienen una importancia más limitada, el orden de importancia de los mismos es: 15min,4h,24h,8h y 30min.

La variable no retardada que aparece como más importante en conjunto es *hora*, alcanzando su máxima representación en los modelos con horizonte de 24h (GLMNET5), le siguen temperaturas; *tmin*, *tmax*.

En los modelos sin variables retardadas (GLMNET6) el orden de importancia de las variables, en los términos aquí considerados, es el siguiente: tmax > hora > tmin > p > > fest > lun > fsof > ...

```
betas$beta = unlist(betas$beta)
class(betas$beta)
```

```
## [1] "numeric"
```

```
betas$betabs = abs(betas$beta)
dim(betas)
```

```
## [1] 34447     8
```

```
betas[1:10, 1:8]
```

```
##    id  modelo vi   vj  nom       beta regres     betabs
## 1   2 GLMNET0 b1 b179 hora 0.00000000   hora 0.00000000
## 2   2 GLMNET0 b1 b179  lun 0.00000000    lun 0.00000000
## 3   2 GLMNET0 b1 b179  mar 0.00000000    mar 0.00000000
## 4   2 GLMNET0 b1 b179  mie 0.00000000    mie 0.00000000
## 5   2 GLMNET0 b1 b179  jue 0.00000000    jue 0.00000000
## 6   2 GLMNET0 b1 b179  vie 0.00000000    vie 0.00000000
## 7   2 GLMNET0 b1 b179  sab 0.00000000    sab 0.00000000
## 8   2 GLMNET0 b1 b179  dom 0.01126332    dom 0.01126332
## 9   2 GLMNET0 b1 b179 fest 0.00000000   fest 0.00000000
## 10  2 GLMNET0 b1 b179 fsof 0.00000000   fsof 0.00000000
```

La figura siguiente muestra la media de los coeficientes significativos en valor absoluto de cada regresor por tipo de modelo.

```
data_coefs = betas %>%
                        filter(beta!=0) %>%
                        group_by(regres, modelo) %>%
                        summarise(mean=round(mean(betabs),4))

ggplot(data_coefs)+
    geom_tile(aes(x=modelo, y=regres, fill=mean))+
    geom_text(aes(x=modelo, y=regres, label=mean),
              color='black', size=2.5, alpha = 1)+
    scale_y_discrete(name='',
        limits=c("dom","lun","mar","mie","jue","vie","sab","fest","fsof","hora",
                 "p", "tmax", "tmin",
                 "l15mi","l30mi","l1hi","l4hi","l8hi","l24hi",
                 "l15mj","l30mj","l1hj","l4hj","l8hj","l24hj"))+
    scale_fill_gradientn(
      colors = c('grey100','cyan1','cyan','green','yellow','red')
      # , limits=c(0,1)
      # , breaks=c(0,1,6.65)
    )+
    labs(x="Tipo de modelo", y="Regresor")
```

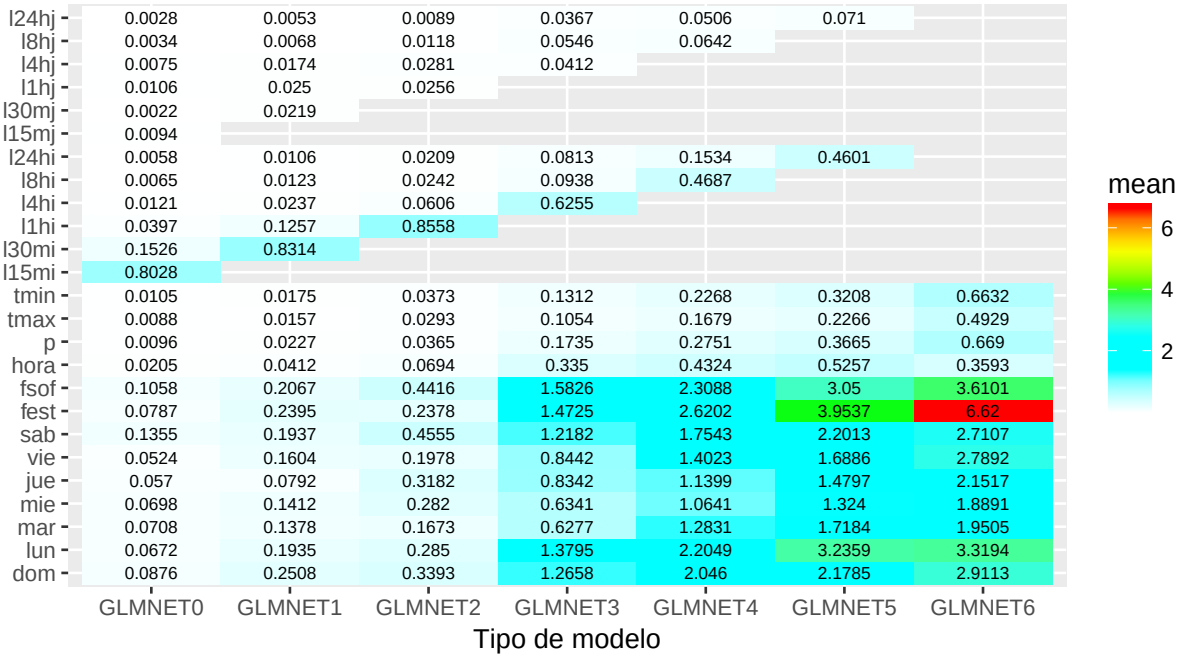

Figura 5: Media |coeficientes sig.| de regresores por tipo de modelo.