Defensa del Trabajo Fin de Máster

UNIVERSIDAD COMPLUTENSE MADRID

FACULTAD DE ESTUDIOS ESTADÍSTICOS

Xinyuan Zheng

2024-2025

TEMA PRINCIPAL

MÁSTER EN CIENCIA DE DATOS E INTELIGENCIA DE NEGOCIOS

Análisis de la afluencia en estaciones de metro mediante modelos de Machine Learning basados en la interdependencia entre estaciones y factores espacio-temporales

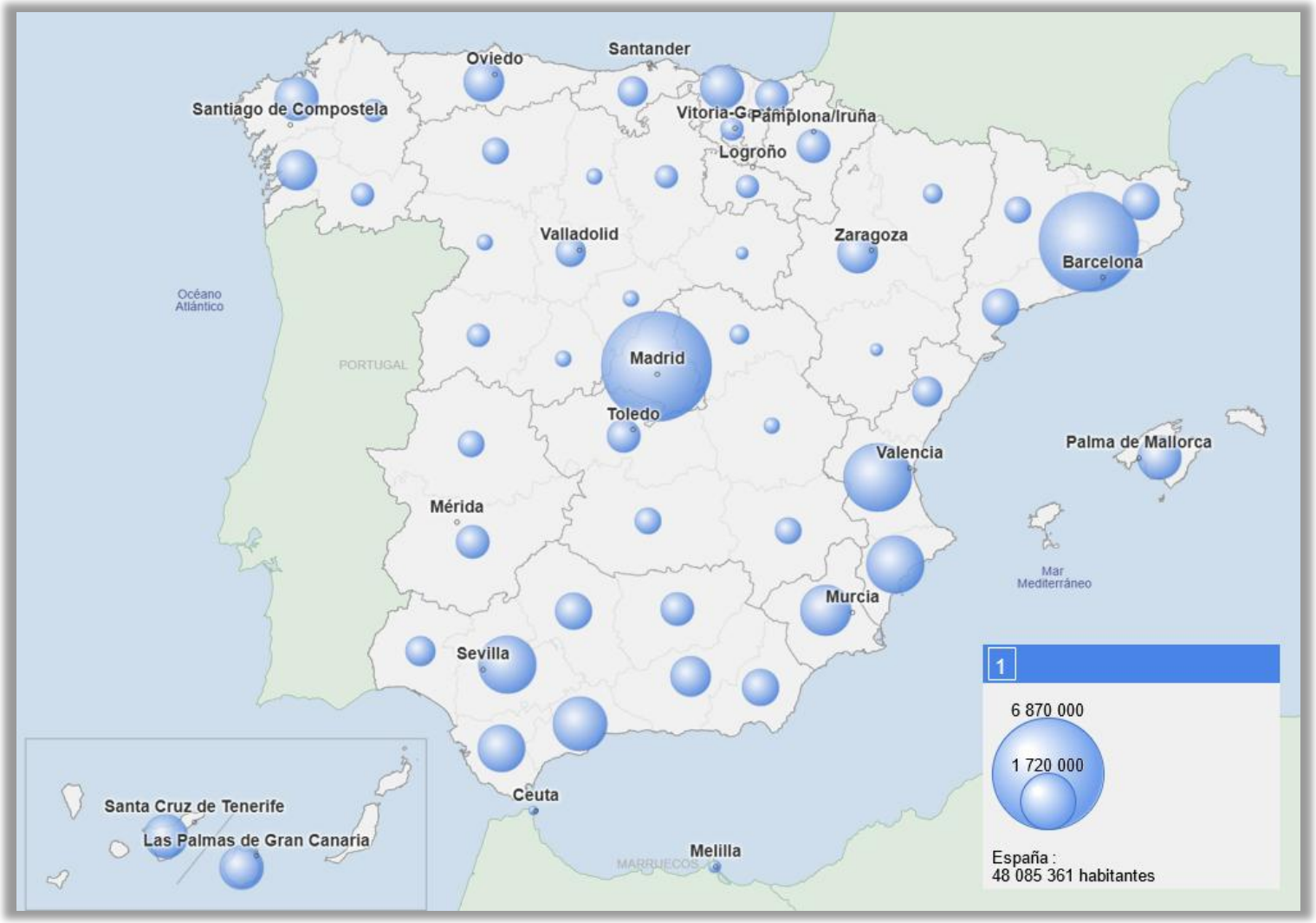
Alumno: Xinyuan Zheng

Tutores: Manuel Núñez García
Manuel Méndez Hurtado

ÍNDICE

1. INTRODUCCIÓN Y CONTEXTO

Fig. 1.1: Distribución de la población en España



En las últimas décadas, se puede observar una tendencia de movilidad de la población hacia las grandes ciudades.

Según el Ministerio de Vivienda, las grandes ciudades concentran el **69%** de la población española y el **76%** del empleo.

Esta concentración de la población trae nuevos retos en la movilidad urbana, especialmente como Madrid que cuenta con un casco urbano antiguo difícilmente modificado a gran escala. Por lo que el transporte subterráneo como el **metro** cobra cada vez más importancia.

Fuente: *Ministerio de Vivienda y Agenda Urbana, 2023*

2. OBJETIVOS

ESTADO DEL ARTE

Factores espaciales

- ❑ Conectividad
- ❑ Enrutamiento
- ❑ Direccionalidad

La mayoría de los estudios se centra en:

Factores externos

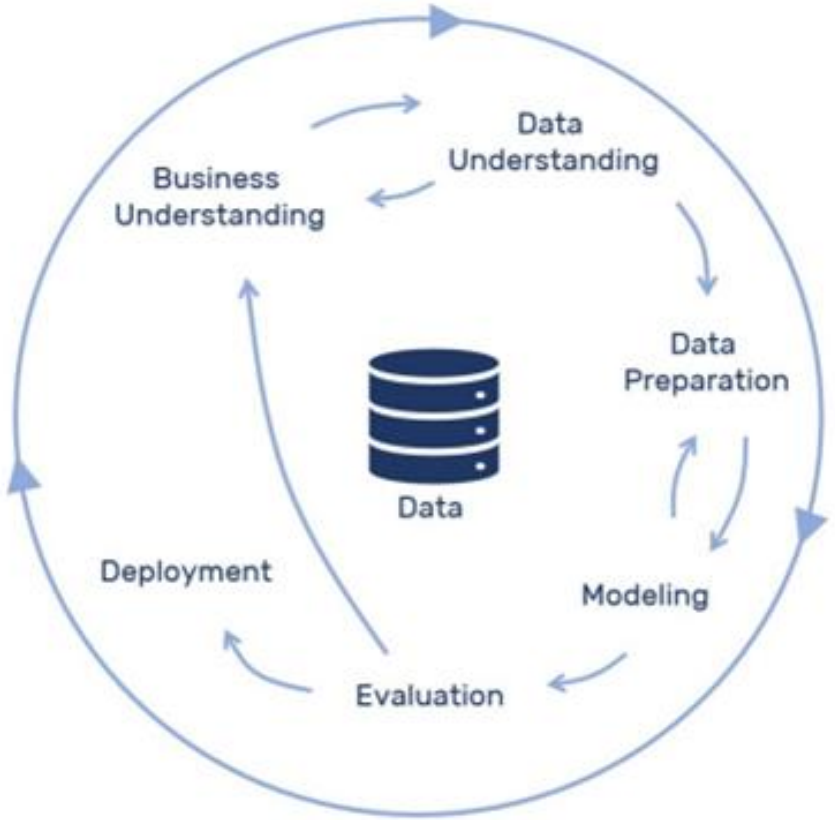
- ❑ Meteorología
- ❑ Hora del día
- ❑ Situación económica

- ❖ Limitación de conjunto de datos
- ❖ Metro Madrid solo dispone del número de viajeros que entra a cada estación, pero no la salida

OBJETIVOS	Aprovechar la cantidad de datos disponibles en cada estación mediante la aplicación de técnicas de Machine Learning para encontrar el mejor modelo que permita estimar la afluencia de una estación a partir de sus estaciones próximas, así como de **factores espacio-temporales y climatológicos**.

3. METODOLOGÍA

CRISP-DM
Cross Industry Standard Process for Data Mining



Fase CRISP-DM	Aplicación en el trabajo
1. Comprensión de negocio	- Introducción - Estado del arte - Objetivo
2. Comprensión de datos	- Orígenes y fuentes de datos - Integración de conjunto de datos - Selección espacial mediante distancia geográficas - Conjunto de datos inicial - Análisis Exploratorio de Datos (EDA) inicial
3. Preparación de datos	- Imputación de Valores NA's - Creación de Dummies y variables rezagadas (lags) - Estandarización de variables - Análisis Exploratorio de datos (EDA) post-preparación - Selección de variables
4. Modelado	- *Modelos individuales lineales:* · Regresión lineal · Lasso/Ridge - *Modelos de ensamblado:* · Random Forest · XGBoost · LightGBM - *Modelos no lineales:* · Redes Neuronales · SVM (SVR)
5. Evaluación	- Validaciones cruzadas - Métricas: R^2, RMSE
6. Despliegue	- Interpretación y muestreo del modelo final

Fuente: *Evolution Paths for Knowledge Discovery and Data Mining Process Models, 2020*

4. EXPLORACIÓN INICIAL

4.1. SELECCIÓN POR ÁREAS

Fig. 4.1: Estaciones totales de Metro Madrid

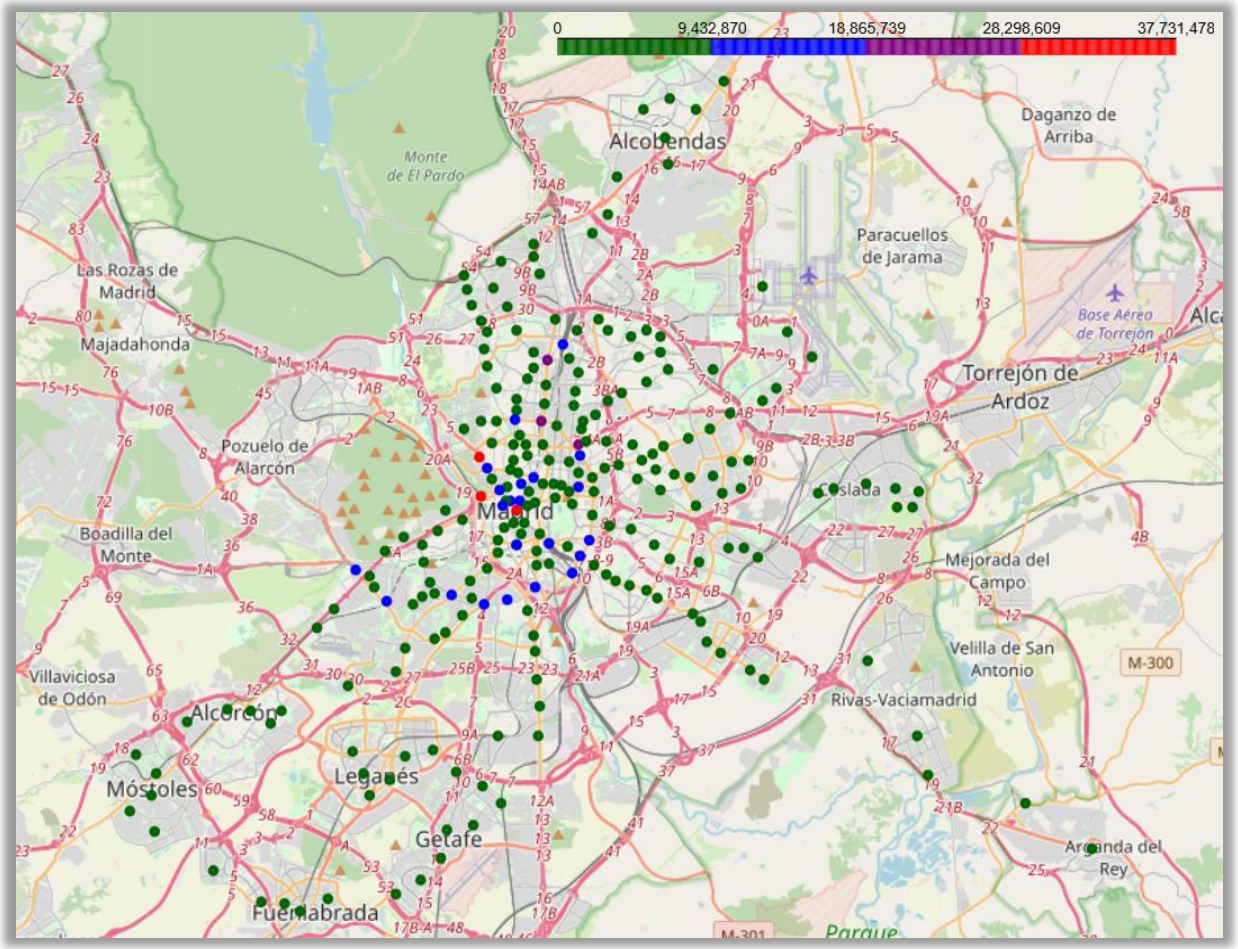
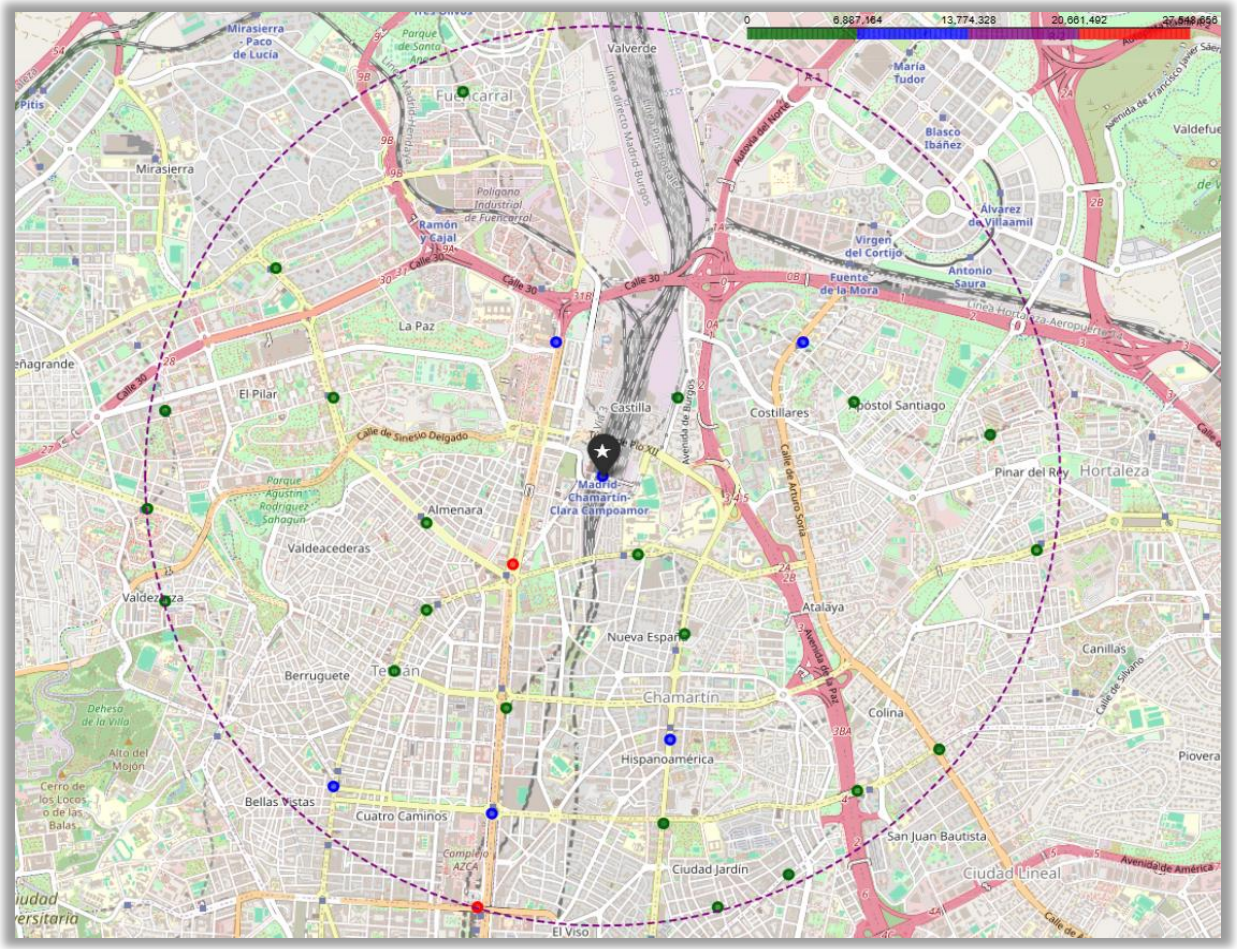


Fig. 4.2: Estaciones próximos a 3km de radio de Chamartín



❖ **VARIABLES ESPACIALES**

Variable Objetiva: Chamartín

Fuente: *Elaboración propia*

❖ VARIABLES TEMPORALES

Variable	Descripción
Dia_semana	Día de la semana (lunes a domingo)
tipo_dia	Tipo de día según calendario laboral (laborable / fin de semana / festivo)

❖ VARIABLES CLIMATOLÓGIAS

Variable	Descripción
tmed	Temperatura media diaria
prec	Precipitación diaria de 07 a 07
velmedia	Velocidad media del viento
sol	Duración de insolación
hrMedia	Humedad relativa media diaria

❖ Proporción de valores atípicos

Valores que cumple las 2 condiciones:

- Criterio de simetría (Z-score o MAD)
- Rango intercuartílico (IQR)

Variable	Proporción de valores atípicos
Santiago Bernabéu	0.00137

❖ Valores nulos (NA's)

Variable	N° total de NA's	Períodos afectados
prec	18	Patrón aleatorio no secuencial
velmedia	1	2022-07-09
sol	3	2022-06-23 / 2022-09-20 / 2023-06-07
Concha Espina	33	2023-08-05 hasta 2023-09-03
Cruz del Rayo	33	2023-08-05 hasta 2023-09-03
Pinar del Rey	104	2022-02-13 hasta 2022-05-27
Santiago Bernabéu	1	2022-05-04

5. PREPARACIÓN DE DATOS

5.1. IMPUTACIÓN DE VALORES NA'S

❖ NA puntuales (1)

- Santiago de Bernabéu
- velmedia
- sol

→ Media aritmética del día **anterior** y **posterior**

❖ NA's consecutivo modelado (33)

- Concha Espina
- Cruz del Rayo

→ Media aritmética del mismo día del mes **anterior** y **posterior**

❖ NA's consecutivo alto (104)

- Pinar del Rey

→ Variable descartada

❖ Caso especial prec (18)

- prec

→ Moda → valor 0

5.2. DUMMIES, LAGS Y ESTANDARIZACIÓN

❖ DUMMIES

- prec
  - **prec_baja**
  - prec_media
  - prec_alta

- Dia_semana
  - **Dia_semana_lunes**
  - Dia_semana_martes
  - Dia_semana_miercoles
  - Dia_semana_jueves
  - Dia_semana_viernes
  - Dia_semana_sabado
  - Dia_semana_domingo

- tipo_dia
  - **tipo_dia_festivo**
  - tipo_dia_fin_de_semana
  - tipo_dia_laborable

❖ Variables rezagadas (lags)

- t-1: corresponde a 1 día antes
- t-2: corresponde a 2 día antes
- t-7: corresponde a 7 día antes

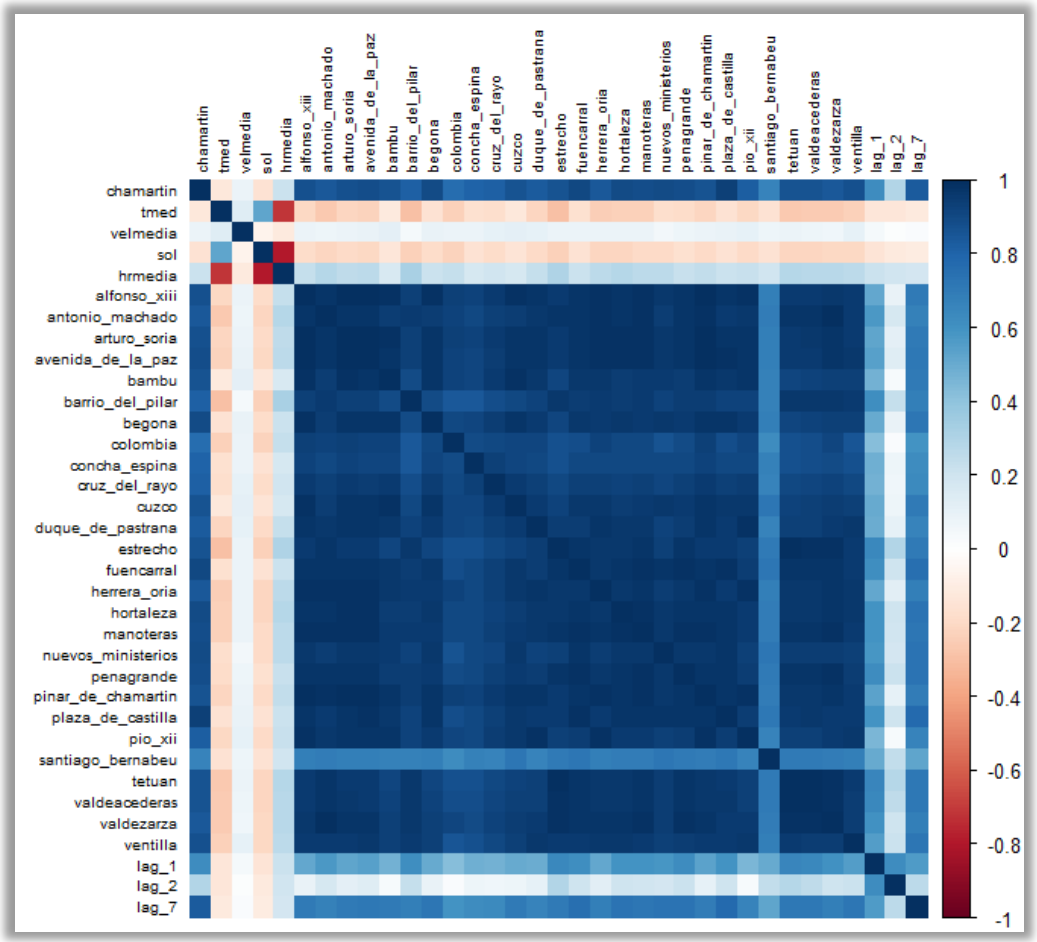
❖ Estandarización

Se procede a estandarizar el conjunto de datos a media 0 y desviación típica 1.

Además de estandarizar los nombres de las columnas para eliminar los espacios, caracteres, mayúscula manteniendo una homogeneidad en el nombramiento.

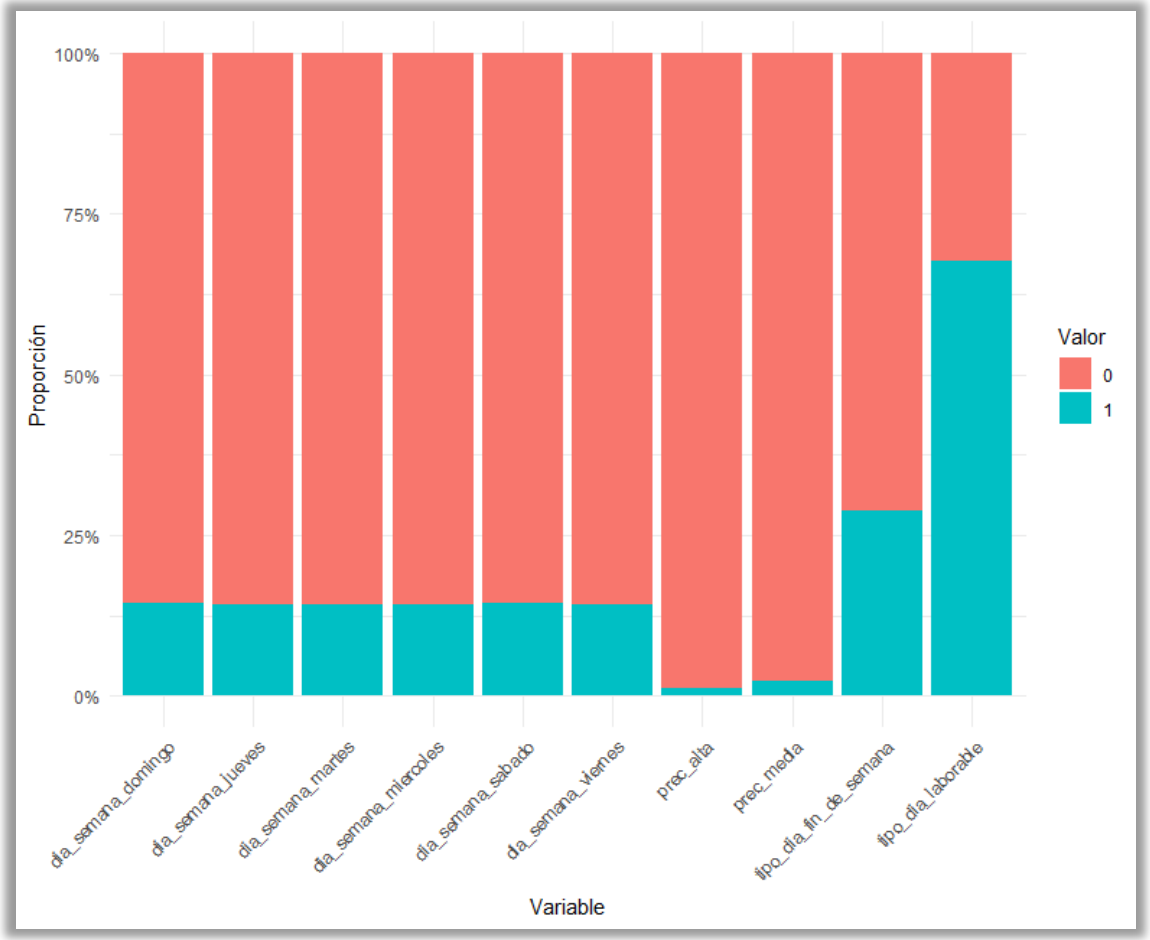
5.3. ANÁLISIS DE LA MULTICOLINEALIDAD Y DISTRIBUCIÓN DE DUMMIES

Fig. 5.1: Análisis de multicolinealidad mediante mapa de calor



- Presencia de multicolinealidad entra las variables predictoras

Fig. 5.2: Distribución de variables dummies



- Variables relevantes: tipo_dia_fin_de_semana y tipo_dia_laborable

Fuente: *Elaboración propia*

6. SELECCIÓN DE VARIABLES

Métodos basados en criterios de información

- AIC (Akaike Information Criterion)
- BIC(Bayesian Information Criterion)
- LEAPS

Métodos de selección por penalización (regularización)

- Lasso
- Ridge

Métodos basados en importancia de variables (embedded)

- Random Forest

Métodos de selección tipo wrapper

- RFE (Recursive Feature Elimination)
- SBF (Selection By Filtering)
- Boruta

Métodos basados en independencia condicional

- MMPC (Max-Min Parents and Children)
- SES(Statistically Equivalent Signatures)

❖ Procedimiento de selección

1. Se entrena con el conjunto de variables para obtener la **mejor combinación de variables** según cada método.

2. Se realiza un reentrenamiento del método usando la mejor combinación de las variables devuelta en el procedimiento previo con la **validación cruzadas repetidas** de 10 pliegues (folds) y 10 repeticiones; en total **100 iteracciones** para obtener un resultado lo suficientemente consistente para la comparativa.

6. SELECCIÓN DE VARIABLES

Fig. 6.1: Resultados de los métodos de selección de variables

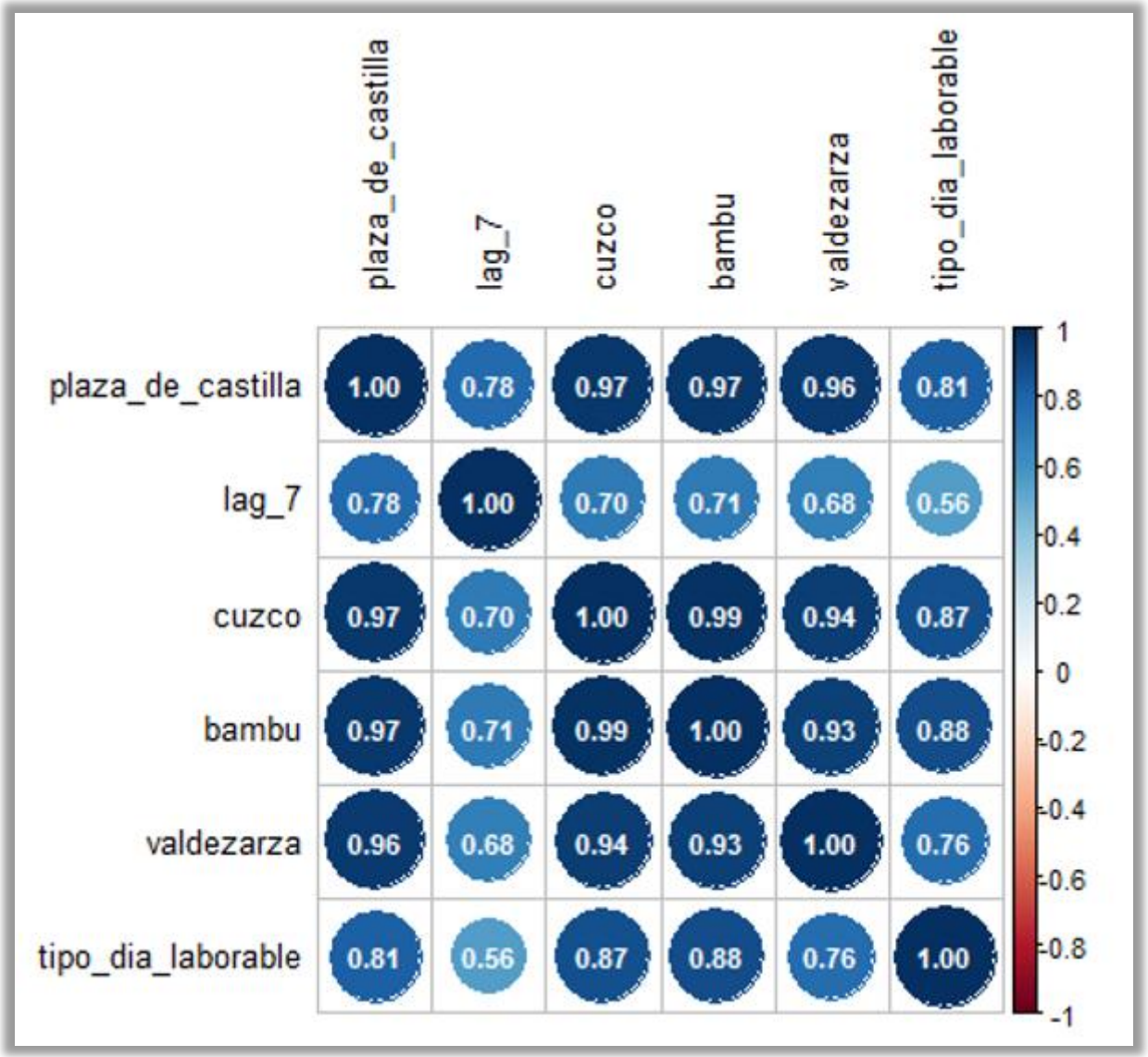
	modelo	n_variables	MSE	RMSE
	<fct>	<int>	<dbl>	<dbl>
1	AIC	26	0.0434	0.208
2	BIC	20	0.0438	0.209
3	LEAPS	21	0.0443	0.210
4	LASSO	42	0.0457	0.214
5	RFE	44	0.0458	0.214
6	RIDGE	44	0.0458	0.214
7	STEP_rep_AIC	18	0.0473	0.217
8	BORUTA	36	0.0508	0.225
9	STEP_rep_BIC	6	0.0515	0.227
10	RF	20	0.0547	0.234
11	MMPC	6	0.159	0.399
12	SES	4	0.181	0.425

La combinación de variables que **mejor resultados** ofrece con respecto a la **menor complejidad** posible es mediante el método de **Stepwise Repetido BIC.**

	Variable	AIC	BIC	STEP_rep_AIC	STEP_rep_BIC	LEAPS	RFE	BORUTA	MMPC	SES	LASSO	RIDGE	RF
1	tmed						X	X			X	X	
2	velmedia						X				X	X	
3	sol	X	X			X	X	X			X	X	
4	hrmedia						X	X	X	X	X	X	
5	alfonso_xiii						X	X				X	
6	antonio_machado	X	X			X	X	X			X	X	
7	arturo_soria						X	X			X	X	X
8	avenida_de_la_paz	X	X	X		X	X	X	X		X	X	
9	bambu	X	X	X	X	X	X	X			X	X	
10	barrio_del_pilar						X	X			X	X	X
11	begona	X		X			X	X			X	X	X
12	colombia	X	X	X		X	X	X			X	X	X
13	concha_espina						X	X			X	X	X
14	cruz_del_rayo						X	X			X	X	X
15	cuzco	X	X	X	X	X	X	X			X	X	
16	duque_de_pastrana						X	X			X	X	X
17	estrecho	X	X			X	X	X			X	X	X
18	fuencarral	X	X			X	X	X			X	X	X
19	herrera_oria						X	X			X	X	X
20	hortaleza	X					X	X			X	X	
21	manoteras	X					X	X			X	X	
22	nuevos_ministerios						X	X	X	X	X	X	X
23	penagrande	X	X			X	X	X	X		X	X	X
24	pinar_de_chamartin	X					X	X	X	X	X	X	
25	plaza_de_castilla	X	X	X	X	X	X	X			X	X	X
26	pio_xii	X		X		X	X	X			X	X	X
27	santiago_bernabeu						X	X			X	X	X
28	tetuan						X	X			X	X	
29	valdeacederas						X	X			X	X	
30	valdezarza	X		X	X		X	X			X	X	X
31	ventilla	X	X	X		X	X	X			X	X	X
32	prec_alta			X			X				X	X	
33	prec_media						X				X	X	
34	dia_semana_martes	X	X	X		X	X				X	X	
35	dia_semana_miercoles	X	X	X		X	X				X	X	
36	dia_semana_jueves	X	X	X		X	X				X	X	
37	dia_semana_viernes	X	X	X		X	X				X	X	
38	dia_semana_sabado	X	X			X	X	X			X	X	
39	dia_semana_domingo	X	X				X	X				X	
40	tipo_dia_laborable	X	X	X	X	X	X				X	X	
41	tipo_dia_fin_de_semana			X		X	X	X			X	X	
42	lag_1	X	X			X	X	X			X	X	X
43	lag_2			X		X	X	X	X	X	X	X	X
44	lag_7	X	X	X	X		X	X			X	X	X

Fuente: *Elaboración propia*

Fig. 6.2: Resultados de los métodos de selección de variables



❖ Factor de Inflación de la Varianza (VIF)

Variable	VIF
plaza_de_castilla	40.877544
lag_7	3.121079
cuzco	45.685197
bambu	52.314683
valdezarza	14.354910
tipo_dia_laborable	6.022752

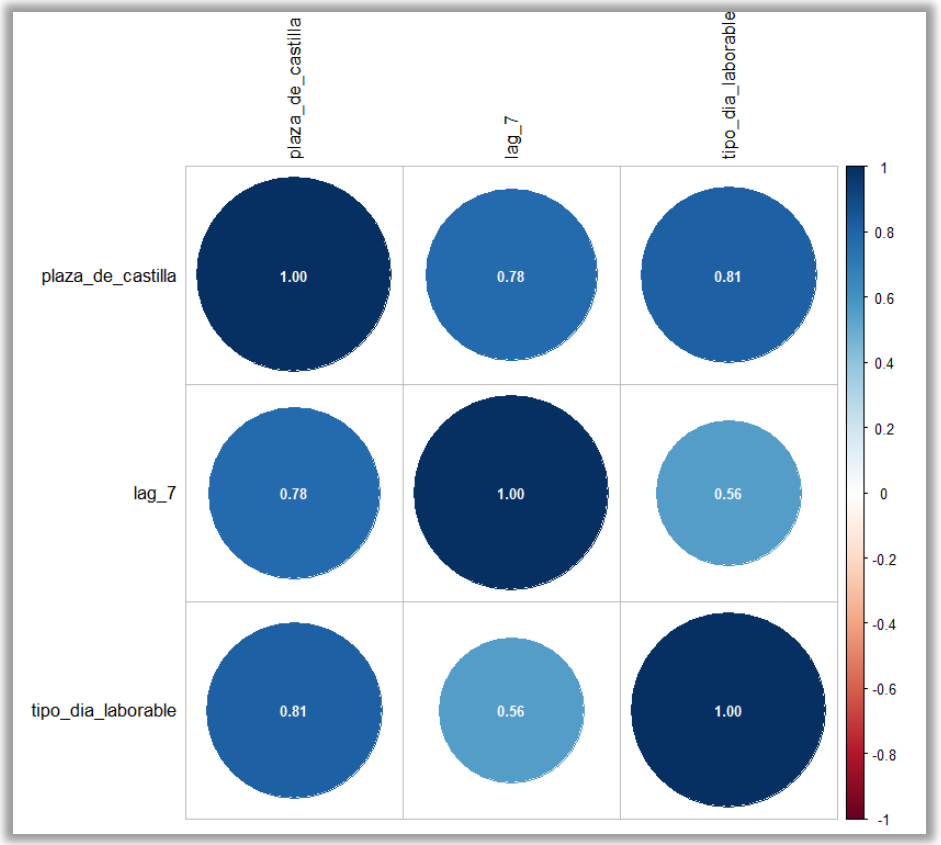
Según el mapa de calor, se puede percibir el problema de la **alta correlación** entre algunas variables explicativas.

Mediante el VIF se muestra que ciertas variables superan el umbral de VIF=10, por lo que se confirma la existencia de la **multicolinealidad**.

Fuente: *Elaboración propia*

6.2. REDUCCIÓN DE VARIABLES POR MULTICOLINEALIDAD

Fig. 6.3: Análisis de multicolinealidad tras el ajuste



El método Stepwise Repetido BIC ajustado, muestra una estructura mas sencilla de variables predictoras sin afectar mucho en el rendimiento.

❖ **Factor de Inflación de la Varianza (VIF) final**

Variable	VIF
plaza_de_castilla	5.295454
lag_7	2.627384
tipo_dia_laborable	3.036607

Tras descartar las variables con mayor valor VIF, la combinación resultante muestra una correlación más aceptable y un valor de VIF por debajo de 10.

Cuadro 6.1: Comparativa del método de selección de variable tras el ajuste

Modelo	N° variables	RMSE	R^2
STEP_rep_BIC	6	0.222	0.950
STEP_rep_BIC (ajustado)	3	0.280	0.922

Fuente: *Elaboración propia*

7. CONTRUCCIÓN DE MODELOS

Modelos individuales lineales:

- Regresión lineal
- Lasso
- Ridge

Modelos de ensamblado:

- Random Forest
- XGBoost
- LightGBM

Modelos no lineales:

- Redes Neuronales
- SVM(SVR)

❖ Procedimiento de construcción del modelo

1. A partir del conjunto de variables obtenidos mediante la selección de variables se construye el modelo con el objetivo de definir la **mejor combinación de hiperparámetro** mediante "**fine tuning**".

2. Se selecciona la mejor combinación de hiperparámetro teniendo en cuenta un balance entre el mejor rendimiento (**menor RMSE**) y la menor complejidad posible (**más simples**).

3. Se realiza un reentrenamiento del modelo usando la mejor combinación de los hiperparámetros mediante la **validación cruzada repetidas** de 10 pliegues (folds) y 10 repeticiones; en total **100 iteraciones** para obtener un resultado lo suficientemente consistente para la comparativa final.

❖ Regresión Lineal

Intercept	RMSE	R^2	MAE	RMSESD	R^2SD	MAESD
TRUE	0.280276	0.9221544	0.2122853	0.04670524	0.02462576	0.02009624

Lasso/ Ridge → **parámetro de regularización (λ)** que minimiza el error de validación

❖ Lasso ($\alpha = 1$)

$\lambda_{\text{óptimo}} = \mathbf{0.001843085}$

RMSE	R^2	MAE
0.2803189	0.9221134	0.2123291

❖ Ridge ($\alpha = 0$)

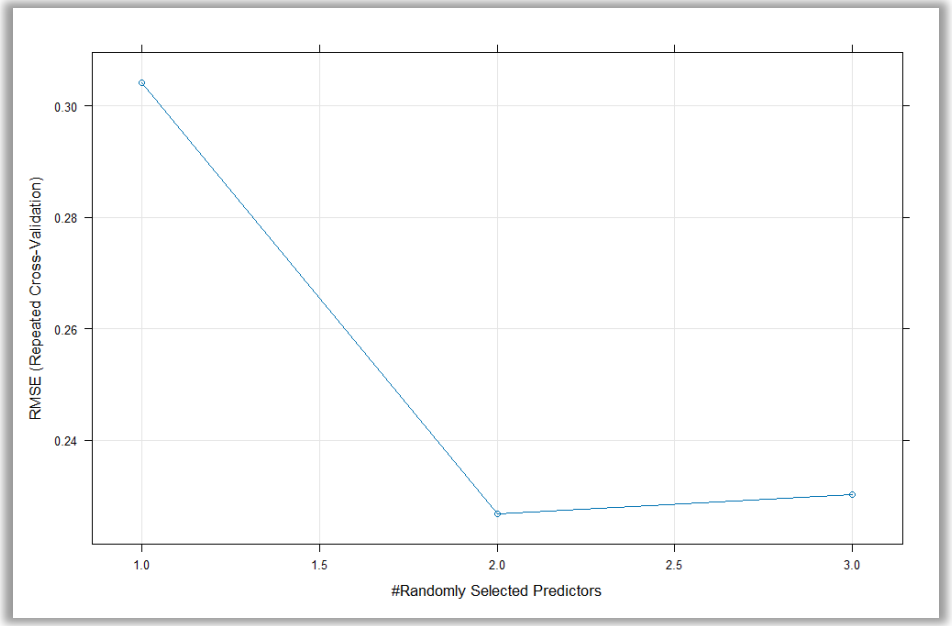
$\lambda_{\text{óptimo}} = \mathbf{0.09388932}$

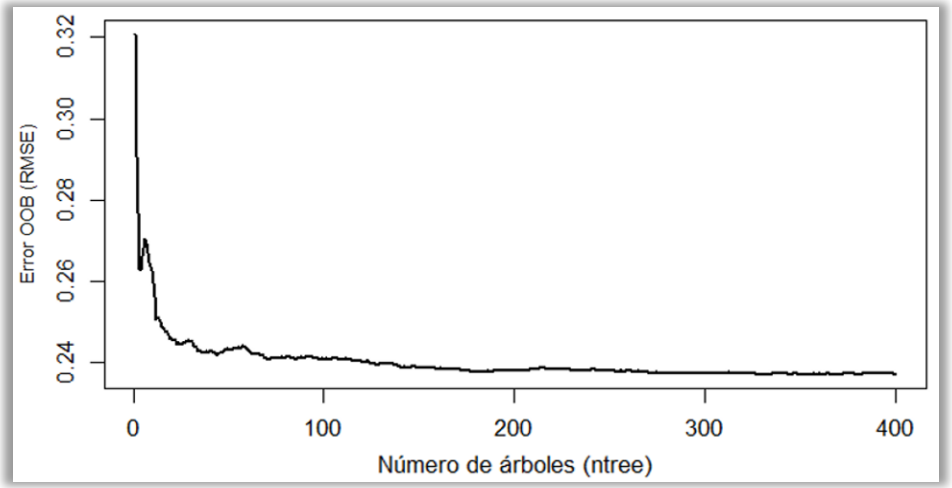
RMSE	R^2	MAE
0.3015469	0.9123395	0.2279584

Fuente: *Elaboración propia*

❖ Random Forest

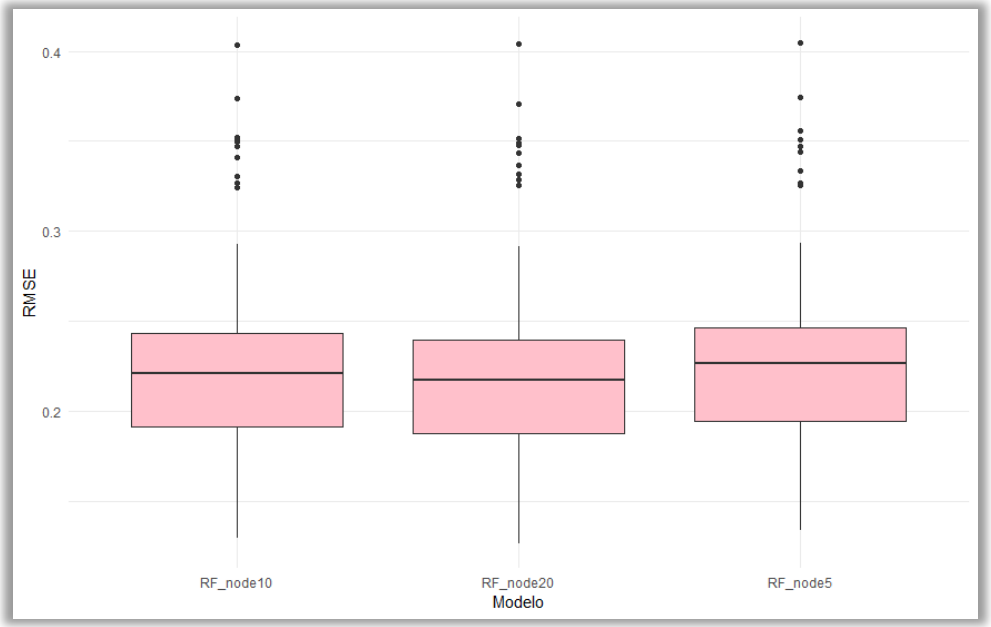
Fig. 7.1: Fine tuning del mtry y ntree





mtry: {1, **2**, 3} # nº var. usada en cada nodo
ntree: {100, **200**, 300, 400} # nº árboles generados
nodesize: {5, 10, **20**} # nº mín. obs. en cada nodo terminal

Fig. 7.2: Fine tuning del nodesize



mtry	RMSE	R^2	MAE	RMSESD	R^2SD	MAESD
2	0.2251605	0.9487059	0.150004	0.05283123	0.02378044	0.01935885

Fuente: *Elaboración propia*

❖ XGBoost

min_child_weight: {3, **5**} # nº mín. obs. en nodo hijo
eta : {0.1, **0.01**, 0.001} # learing rate, contribución cada árbol
max_depth : {**3**, 5} # profundidad max. cada árbol
nrounds : {300, **500**, 1000} # nº iteraciones/arbol

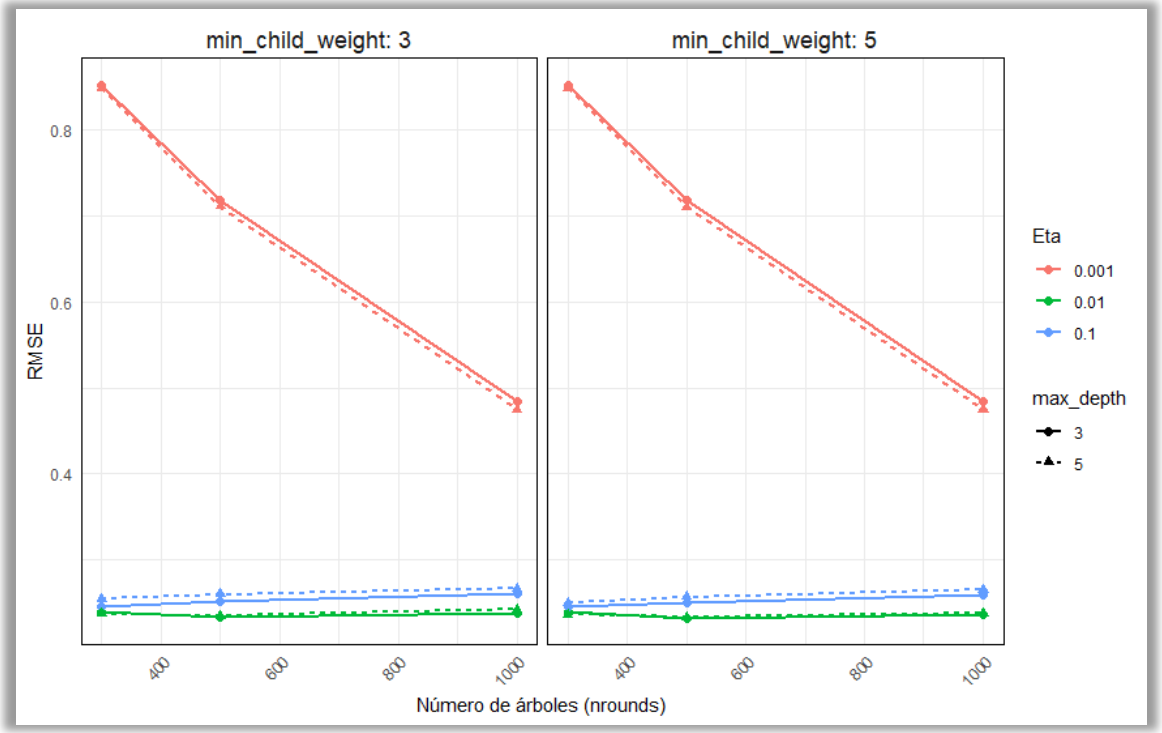


Fig. 7.3: Fine tuning de hiperparámetros de XGBoost

nrounds	eta	max_depth	gamma	colsample_bytree	min_child_weight
500	0.01	3	0	1	5

subsample	RMSE	R^2	MAE	RMSESD	R^2SD	MAESD
1	0.2314668	0.946073	0.1589347	0.05127528	0.02360878	0.01975624

❖ LightGBM

min_data_in_leaf: {3, **5**} # nº mín. obs. en hoja terminal
learning_rate: {**0.1**, 0.01, 0.001} # tasa aprendizaje
num_leaves: {**3**, 5} # nº máx. hojas por árbol
nrounds: {300, **500**, 1000} # nº iteraciones/árbol

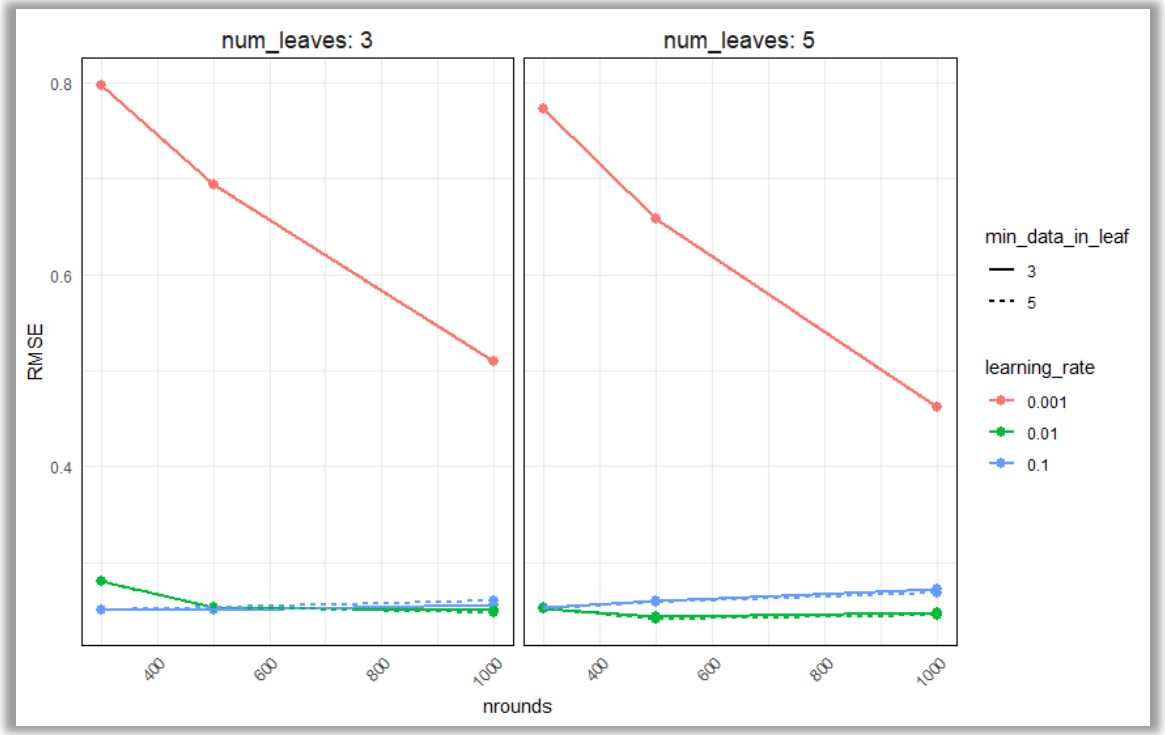


Fig. 7.4: Fine tuning de hiperparámetros de LightGBM

Modelo	RMSE	R^2	MAE
LGBM	0.250832	0.9371698	0.1678948

Fuente: *Elaboración propia*

❖ Red Neuronal

$$\frac{Nº\ observaciones}{Nºobs/parámetro} = parám.máx.recomendados$$

$$\frac{723}{20} = 36{,}15\ parámetros\ máx.recomendados$$

$$Nº\ parámetros\ = h(k\ +1) + h + 1$$

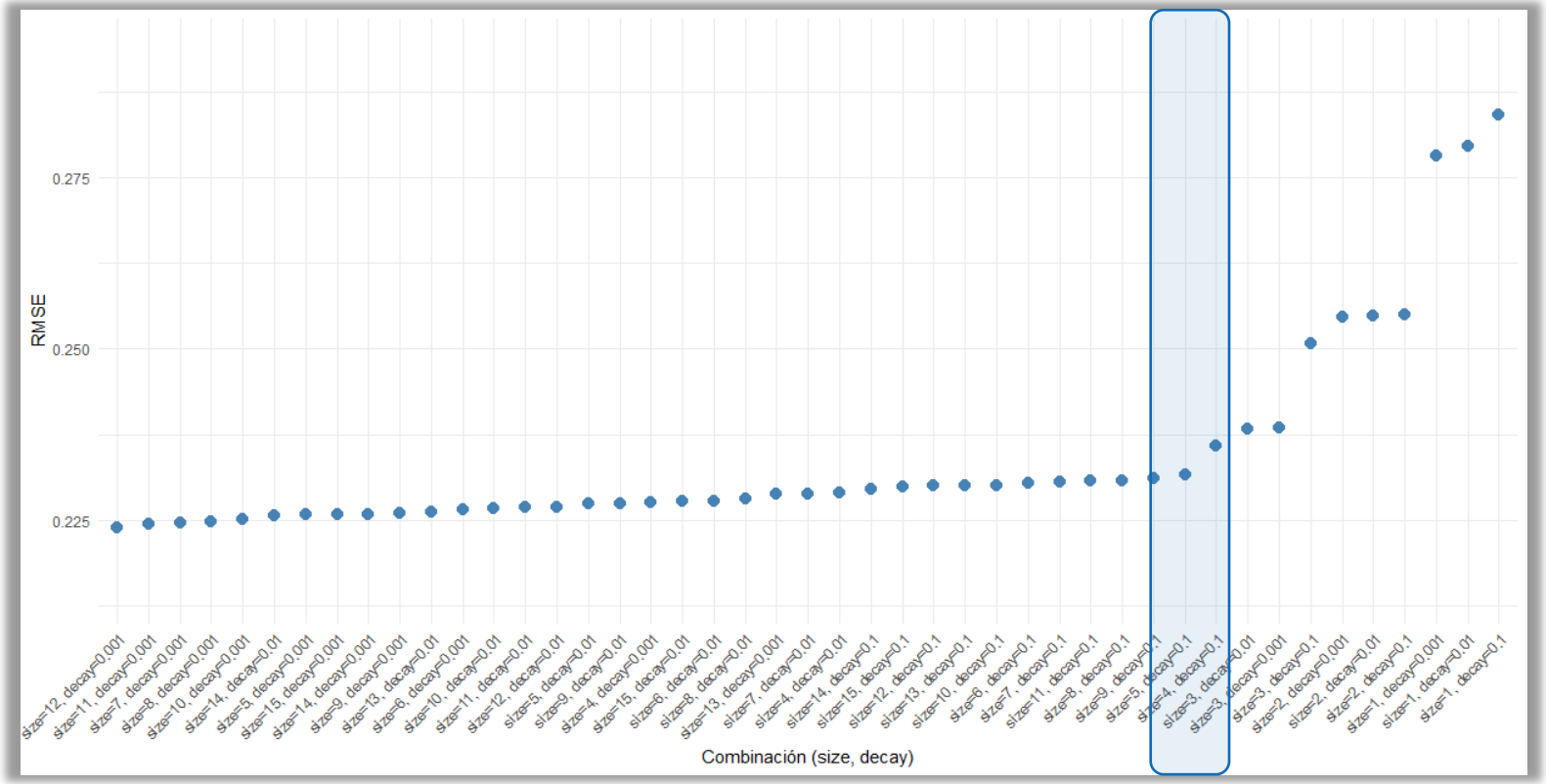
$$36{,}15 = h(3+1) + h + 1$$

$$36{,}15 = 5h + 1$$

$$h = \mathbf{7{,}03\ nodos\ máx.}$$

h: número de nodos ocultos
k: número de variables explicativas empleadas en el modelo.

Fig. 7.5: Fine tuning simple de la Red neuronal con nodos 1 a 15



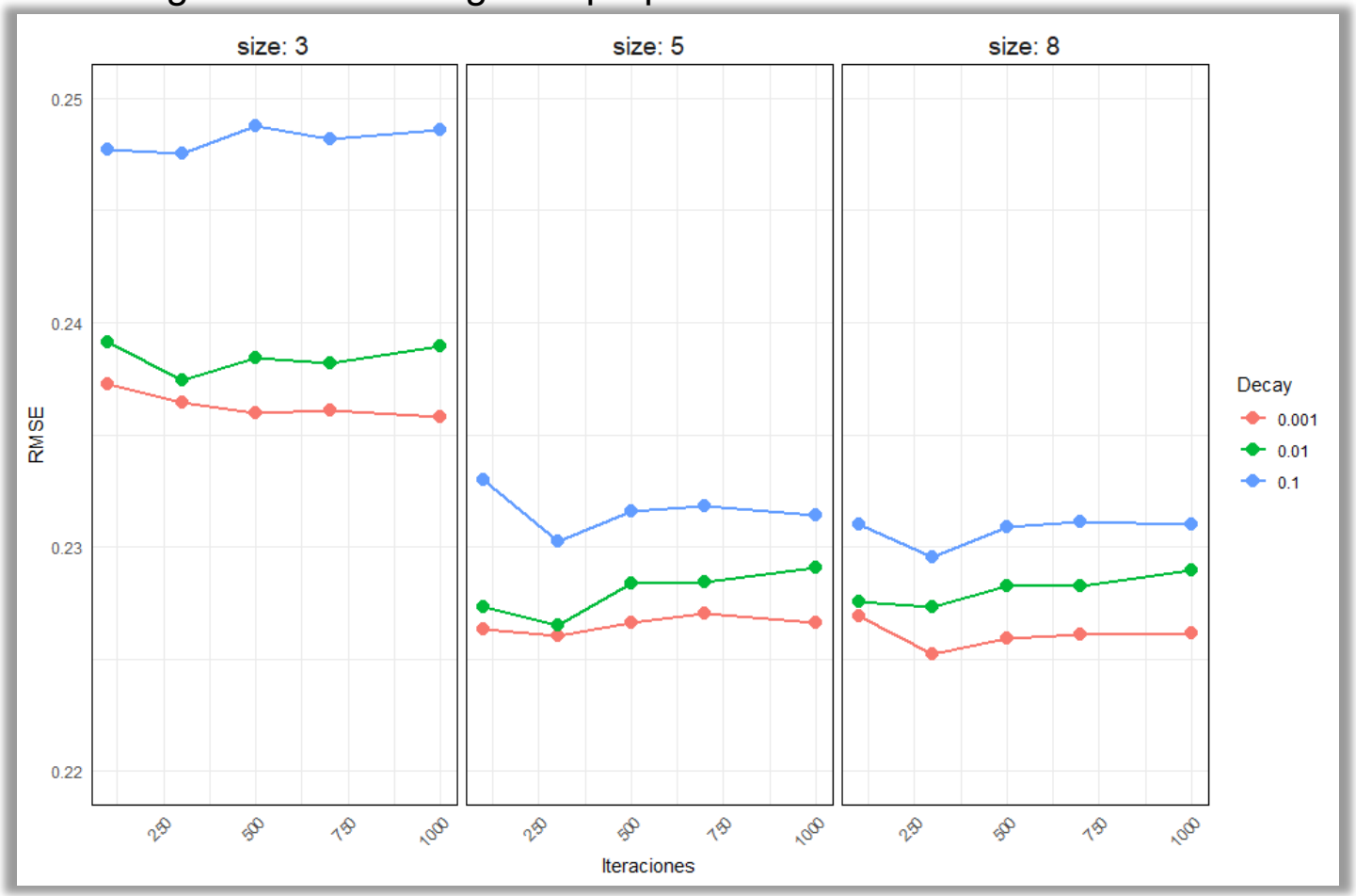
El **límite máximo** recomendado por la función empírica es de no superar los **7 nodos**, mientras que mediante la fine tuning simple, se puede observar que, a partir de **3 a 4 nodos**, el RMSE empieza a estabilizarse, estableciendo como posibles **límites mínimos**.

Fuente: *Elaboración propia*

❖ Red Neuronal

size : {3, **5**, 8} # nº nodos en capa oculta
decay : {**0.1**, 0.01, 0.001} # par. regularización
iter: {100, **300**, 500, 700, 1000} # nº máx. iteraciones

Fig. 7.6: Fine tuning de hiperparámetros de la Red neuronal



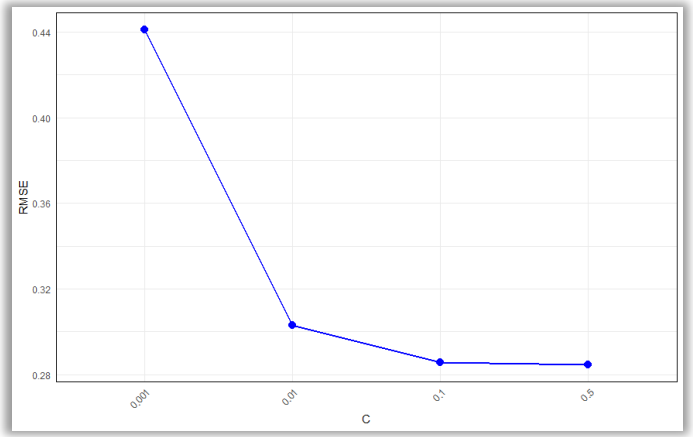
size	decay	bag	RMSE	R^2	MAE	RMSESD	R^2SD	MAESD
5	0.1	FALSE	0.2308	0.9465	0.1602	0.0488	0.0226	0.0176

Fuente: *Elaboración propia*

❖ SVR Lineal

C : {0.5, **0.1**, 0.01, 0.001} # par. regularización

Fig. 7.7: Fine tuning de hiperparámetros de SVR Lineal

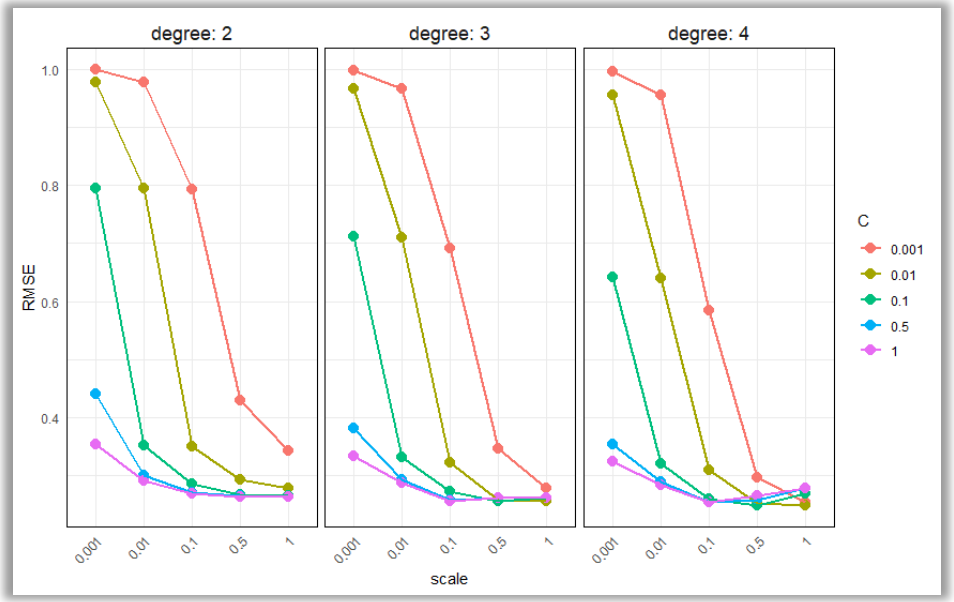


RMSE	R^2	MAE
0.2858566	0.9204151	0.207427

❖ SVR Polinomial

degree : {**2**, 3, 4 } # grado polinomial
scale : {1, 0.5, **0.1**, 0.01, 0.001 } # escala entrada
C : {1, 0.5, **0.1**, 0.01, 0.001} # par. regularización

Fig. 7.8: Fine tuning de hiperparámetros de SVR Polinomial

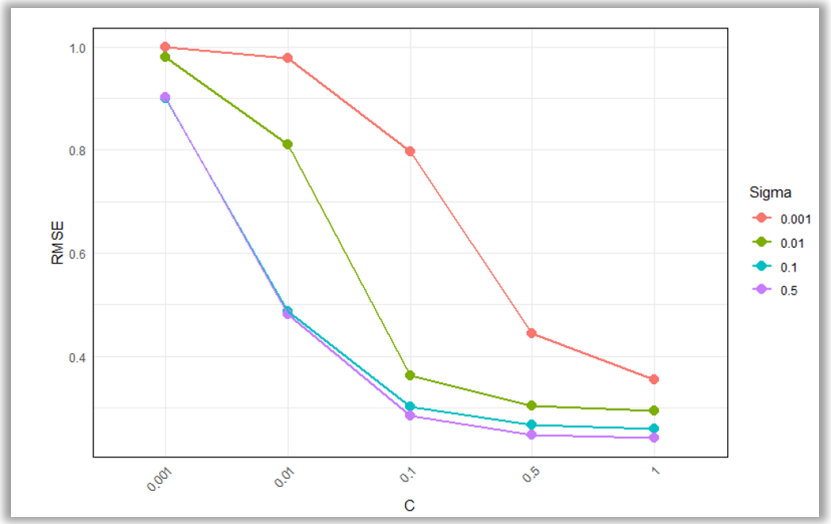


RMSE	R^2	MAE
0.2864518	0.9202174	0.2039351

❖ SVR Radial (RBF)

sigma : {0.001, 0.01, **0.1**, 0.5} # amplitud función de la base radial
C : {1, 0.5, **0.1**, 0.01, 0.001} # par. regularización

Fig. 7.9: Fine tuning de hiperparámetros de SVR Radial (RBF)



RMSE	R^2	MAE
0.3009869	0.9132075	0.2115539

Fuente: *Elaboración propia*

8. EVALUACIÓN DE RESULTADOS

8.1. SELECCIÓN DEL MODELO ÓPTIMO

Fig. 8.1: Comparación de modelos finales según R^2

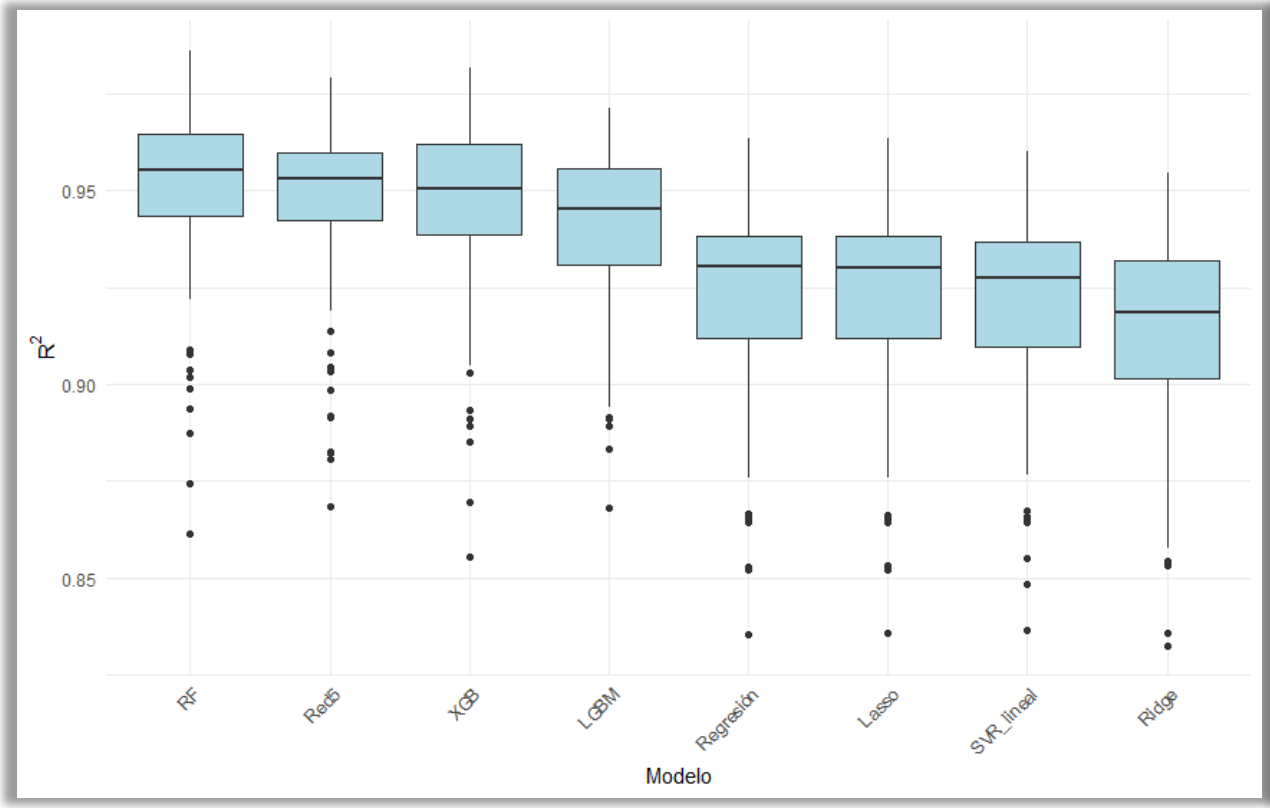
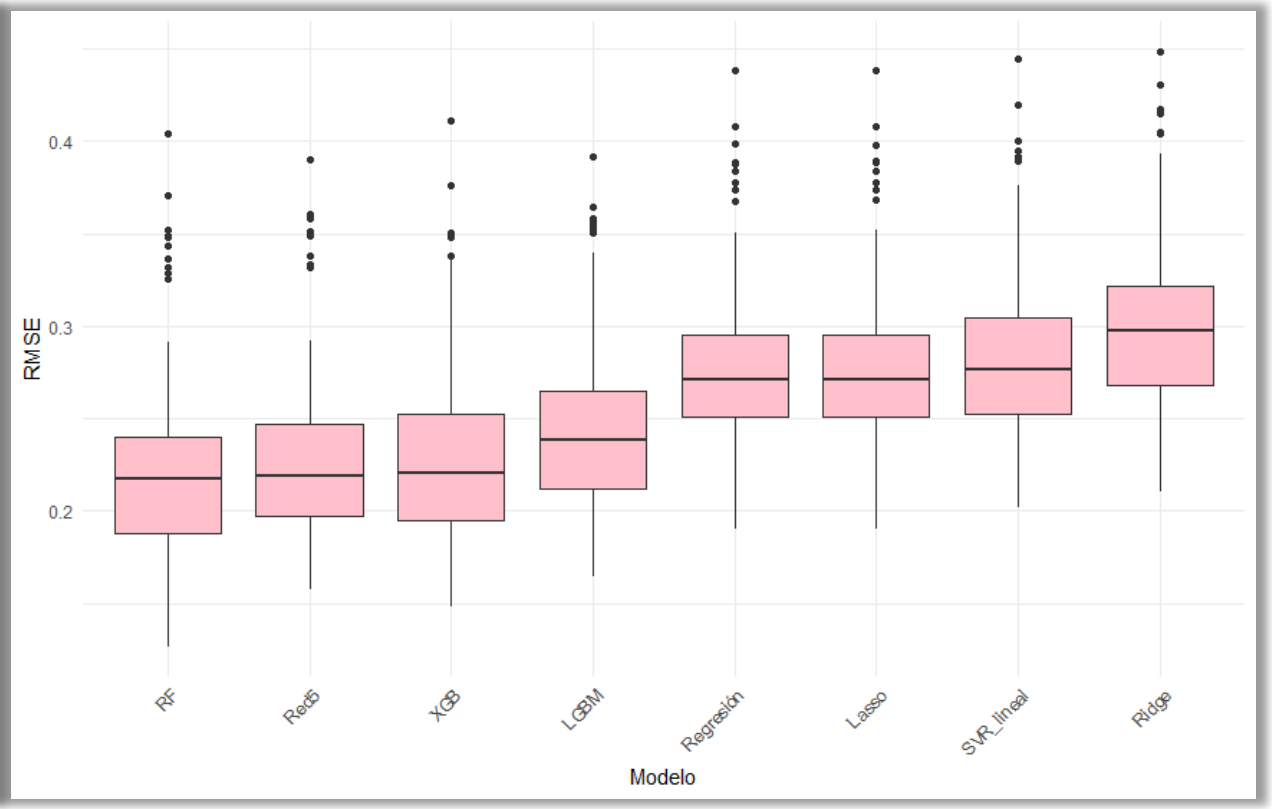


Fig. 8.2: Comparación de modelos finales según $RMSE$



Comparando los resultados de los modelos según R^2 y $RMSE$, se confirma que el modelo óptimo es el **Random Forest**, obteniendo el mayor R^2 y el menor $RMSE$ entre todos los modelos de la comparativa.

Fuente: *Elaboración propia*

9. EVALUACIÓN DE RESULTADOS CON SAS

Fig. 9.1: Resumen de errores del train=100% con SAS

Modelo seleccionado	Nodo predecesor	Nodo del modelo	Descripción del modelo	Variable target	Train: Average Squared Error ▲	Train: Root Mean Squared Error
Y	Boost	Boost	Gradient Boosting	chamartin	0.035759	.
	Neural	Neural	Red neuronal 5	chamartin	0.048359	0.22397
	Reg	Reg	Regresión	chamartin	0.079729	0.283148

Fig. 9.1: Resumen de errores del train/validation (70/30) con SAS

Modelo seleccionado	Nodo predecesor	Nodo del modelo	Descripción del modelo	Variable target	Valid: Average Squared Error ▲	Valid: Root Mean Square Error	Valid: Root Average Squared Error
Y	Neural2	Neural2	Red neuronal 5	chamartin	0.058238	0.241327	0.241327
	Boost2	Boost2	Gradient Boosting	chamartin	0.067493	.	0.259795
	Reg2	Reg2	Regresión	chamartin	0.090725	0.301205	0.301205

Empleando el **100%** del conjunto de datos para el entrenamiento (**train**) se muestra que los modelos ensamblados tipo **Gradient Boosting** y los modelos tipo Red Neuronal consiguen mejorar el error cometido por la regresión lineal.

Realizando una partición en **train/validation** al **70/30**, se puede observar un resultado similar, aunque esta vez el modelo de **Red Neuronal** muestra un menor error en la parte del conjunto de datos para la validación con respecto al Gradient Boosting.

El resultado obtenido mediante SAS respalda las conclusiones tomadas en la diapositiva, los modelos de Machine Learning tipo Red Neuronal y modelos ensamblados como Random Forest, XGBoost o LightGBM ofrece mejores resultados que la regresión lineal para este conjunto de datos.

Fuente: *Elaboración propia*

10. INTERPRETACIÓN DEL MODELO ÓPTIMO

10.1. RANDOM FOREST

❖ **SHAP**

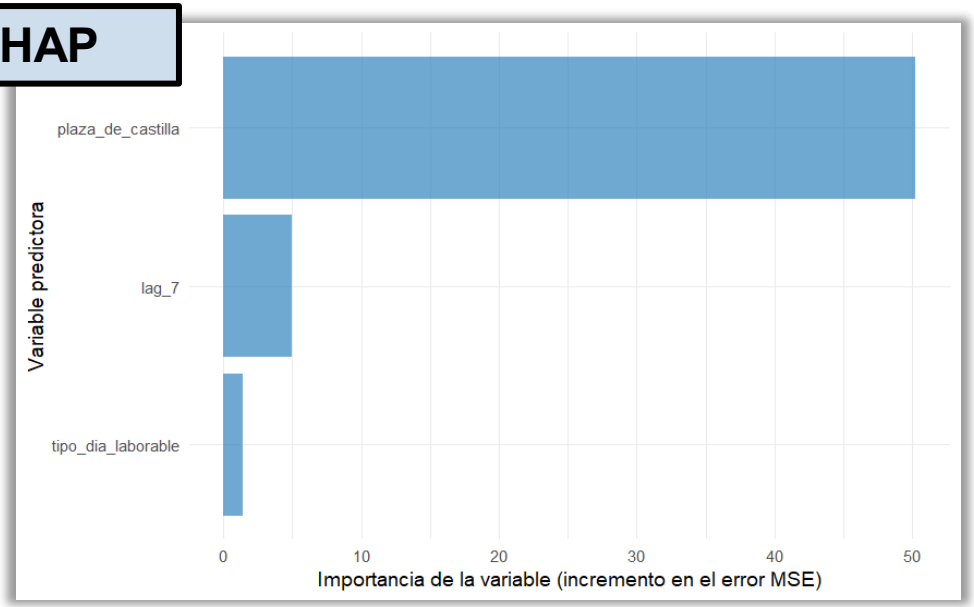


Fig. 10.1: Importancia de variables con SHAP

❖ **LIME**

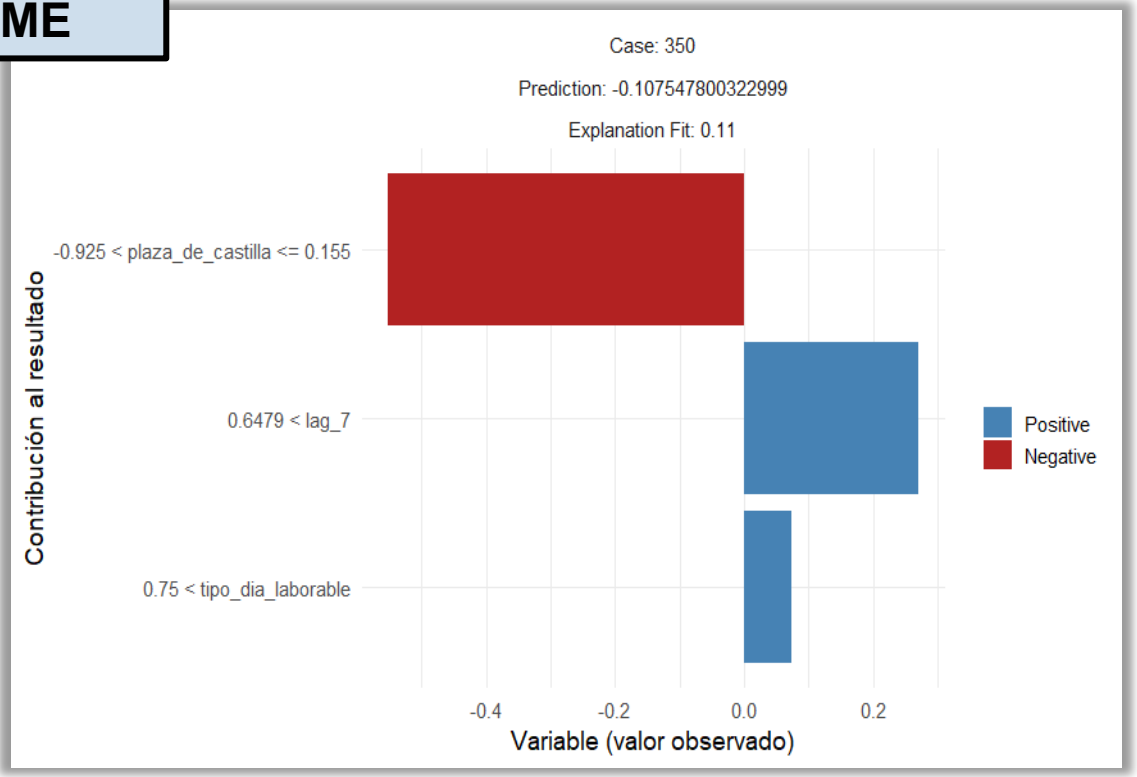


Fig. 10.2: Explicabilidad local mediante LIME

❖ **ÁRBOL**

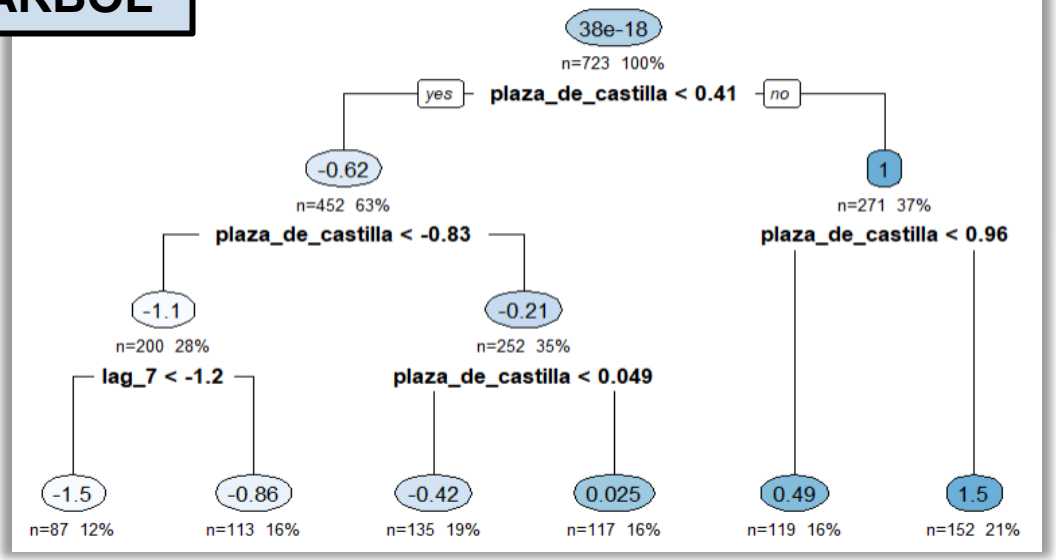


Fig. 10.3: Árbol representativo del Random Forest

La estación **Plaza de Castilla** es la variable que más contribuye en la estimación de la afluencia que entra al Chamartín.

Fuente: *Elaboración propia*

11.1. REGRESIÓN LINEAL

Fig. 11.1: Coeficientes del modelo de regresión lineal

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        0.23769    0.02854   8.328 4.13e-16 ***
plaza_de_castilla  0.88303    0.02423  36.440  < 2e-16 ***
lag_7              0.24348    0.01707  14.264  < 2e-16 ***
tipo_dia_laborable -0.35142    0.03922  -8.961  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

❖ ECUACIÓN

$$Chamartín = 0.23769 + 0.88303 * plaza\_de\_castilla + 0.24348 * lag\_7 - 035142 * tipo\_dia\_laborable$$

La mejora de rendimiento del Random Forest con respecto a la regresión lineal es bastante reducido con respecto a la complejidad del modelo que conlleva. A la hora de tener que interpretar la influencia de las variables explicativas sobre la variable objetiva, el **modelo de regresión lineal** sigue siendo la más adecuado para este trabajo.

Fuente: *Elaboración propia*

12. CONCLUSIÓN Y FUTURAS LÍNEAS DE TRABAJO

El modelo óptimo obtenido del proceso de modelado es el **Random Forest**.

Sin embargo, para un proyecto real con necesidad de llevar el modelo a la fase de producción, la pequeña mejora del rendimiento y error que muestra el Random Forest no justifica su implementación debido a que se trata de un modelo más complejo y complicado de interpretar por su mecanismo de funcionamiento.

Mientras tanto, el modelo de **Regresión Lineal** se presentará como la **mejor opción** gracia a su estructura mas sencilla y de ser mas fácil de interpretar la influencia de cada variable sobre la objetiva mediante el valor de los coeficientes, tal como el objetivo del presente trabajo.

Una de las principales limitaciones del presente proyecto, es la **escasez de observaciones** en el conjunto de datos y la **escasez de variables predictoras** de calidad acorde al objetivo de investigación del presente trabajo.

Como recomendación de futuras líneas de investigación, sería recomendable trabajar con un conjunto de datos más extenso y con mayor diversidad de variables explicativas para poder desarrollar modelos predictivos más avanzados, así como **técnicas de ensamblados** para combinar diferentes modelos de Machine Learning, así como modelos tradicionales tipo ARIMA para los conjuntos de datos tipo series temporales, como el que se ha utilizado en el presente trabajo.

GRACIAS

Lenguaje de programación	Entorno de desarrollo (IDE)	Versión
Python	PyCharm 2024.2.3	Python 3.12
R	RStudio 2024.12.0+467	R 4.4.1
SAS	SAS Enterprise Miner Workstation 14.1	SAS 9.4