

FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN CIENCIA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2024/2025

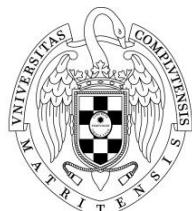
Trabajo de Fin de Máster

TÍTULO: 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

Septiembre de 2025



UNIVERSIDAD COMPLUTENSE
MADRID

Declaración Responsable sobre Autoría y Uso Ético de
Herramientas de Inteligencia Artificial (IA)

Yo, ZHENG XINYUAN

Con DNI/NIE/PASAPORTE: X4472845N

declaro de manera responsable que el/la presente:

- Trabajo de Fin de Grado (TFG)
- Trabajo de Fin de Máster (TFM)
- Tesis Doctoral

Titulado/a

MÁSTER EN CIENCIA DE DATOS E INTELIGENCIA DE NEGOCIOS

es el resultado de mi trabajo intelectual personal y creativo, y ha sido elaborado de acuerdo con los principios éticos y las normas de integridad vigentes en la comunidad académica y, más específicamente, en la Universidad Complutense de Madrid.

Soy, pues, autor del material aquí incluido y, cuando no ha sido así y he tomado el material de otra fuente, lo he citado o bien he declarado su procedencia de forma clara -incluidas, en su caso, herramientas de inteligencia artificial-. Las ideas y aportaciones principales incluidas en este trabajo, y que acreditan la adquisición de competencias, son mías y no proceden de otras fuentes o han sido reescritas usando material de otras fuentes.

Asimismo, aseguro que los datos y recursos utilizados son legítimos, verificables y han sido obtenidos de fuentes confiables y autorizadas. Además, he tomado medidas para garantizar la confidencialidad y privacidad de los datos utilizados, evitando cualquier tipo de sesgo o discriminación injusta en el tratamiento de la información.

En Madrid a 27 de junio de 2025

FIRMA



Índice general

Índice	I
Índice de Figuras	IV
Índice de Cuadros	VI
Resumen	VII
Abstract	VIII
1. Introducción	1
1.1. Contexto	1
1.2. Planteamiento del problema	3
1.3. Justificación del trabajo	4
2. Estado del arte	5
3. Objetivos	7
4. Metodología	8
5. Fundamentos teóricos	11
5.1. Modelos individuales lineales	12
5.1.1. Regresión lineal	12
5.1.2. Lasso	12
5.1.3. Ridge	13
5.2. Modelos basados en técnicas de ensamblado	13
5.2.1. Random Forest	14
5.2.2. XGBoost	14
5.2.3. LightGBM	15
5.3. Modelos no lineales complejas y tipo kernel	15
5.3.1. Redes neuronales	15
5.3.2. SVR	17
5.4. Métodos de evaluación	17
5.4.1. Validación cruzada repetida	17
5.4.2. Coeficiente de determinación (R^2)	18
5.4.3. Raíz del error cuadrático medio (RMSE)	18
6. Entorno de desarrollo	20

7. Exploración inicial del conjunto de datos	22
7.1. Orígenes y fuentes de datos	22
7.2. Integración del conjunto de datos	23
7.3. Selección espacial mediante distancias geográficas	23
7.4. Conjunto de datos inicial	25
7.5. Análisis exploratorio de datos (EDA) inicial	25
7.5.1. Recuento de valores atípicos (outliers)	26
7.5.2. Recuento de Valores Nulos (NA's)	26
8. Análisis y preparación de datos	28
8.1. Imputación de valores NA's	28
8.2. Generación de dummies	29
8.3. Creación de variables rezagadas (lags)	29
8.4. Estandarización de variables	30
8.5. Análisis exploratorio de datos (EDA) post- preparación	30
8.5.1. Evolución temporal de la variable objetivo	30
8.5.2. Análisis de la correlación (gráfica de dispersión)	30
8.5.3. Análisis de la multicolinealidad (mapa de calor)	32
8.5.4. Distribución de variables dummies	33
9. Selección de variables	34
9.1. Métodos de selección	34
9.1.1. Métodos basados en criterios de información	34
9.1.2. Métodos de selección por penalización (regularización)	35
9.1.3. Métodos basados en importancia de variables (embedded) . .	35
9.1.4. Métodos de selección tipo wrapper	35
9.1.5. Métodos basados en independencia condicional	36
9.2. Conjunto de variables seleccionadas	36
9.3. Resultado obtenido	37
9.4. Revisión de la multicolinealidad	38
10. Construcción del modelo y evaluación de resultados	42
10.1. Red neuronal	42
10.2. Regresión lineal	45
10.3. Ridge	45
10.4. Lasso	45
10.5. Random forest	46
10.6. XGBoost	48
10.7. LightGBM	49
10.8. SVR	51
10.8.1. SVR Lineal	51
10.8.2. SVR polinomial	52
10.8.3. SVR radial (RBF)	53
10.8.4. Mejor modelo SVR	54

11. Selección del modelo óptimo	55
11.1. Evaluación de los modelos finales	55
11.2. Evaluación complementaria con SAS	56
11.3. Interpretabilidad del modelo ganador: <i>Random Forest</i>	58
11.3.1. Importancia de las variables	58
11.3.2. Explicación local de la predicción (LIME)	59
11.3.3. Interpretación mediante un árbol simple	59
11.4. Otras consideraciones sobre el modelo óptimo	60
12. Conclusión	62
13. Limitaciones y posibles líneas de trabajo futuro	63
Bibliografía	66
A. Material adicional	67

Índice de Figuras

1.1. Distribución de la población en España	1
1.2. Densidad de población en España	2
4.1. Modelo CRISP-DM	8
5.1. Configuración básica de una red neuronal artificial	16
7.1. Estaciones totales de Metro Madrid	24
7.2. Estaciones próximos a 3km de radio de Chamartín	24
8.1. Evolución de las entradas de viajeros en la estación de Chamartín	31
8.2. Gráfica de dispersión entre variables predictoras y Chamartín	31
8.3. Detección de multicolinealidad mediante mapa de calor	32
8.4. Distribución de variables dummies (0/1)	33
9.1. Variables escogidos por cada método de selección de variables	37
9.2. Resultados de los método de selección de variables	38
9.3. Análisis de multicolinealidad entre variables predictoras	38
9.4. Análisis de multicolinealidad tras el ajuste	40
10.1. Fine tuning automático de los hiperparámetros de la Red Neuronal	43
10.2. Fine tuning de los hiperparámetros de la Red Neuronal	44
10.3. Fine tuning del parámetro <i>mtry</i> en el modelo Random Forest	46
10.4. Evolución del error OOB en función del <i>ntree</i> en Random Forest	47
10.5. Comparación del RMSE para distintos tamaños de nodo en Random Forest	47
10.6. Fine tuning de los hiperparámetros del modelo XGBoost	49
10.7. Fine tuning de los hiperparámetros del modelo LightGBM	50
10.8. Fine tuning del parámetro <i>C</i> en el modelo SVR Lineal	51
10.9. Fine tuning del parámetro <i>C</i> en el modelo SVR Lineal	52
10.10. Fine tuning de los hiperparámetros en el modelo SVR Radial (RBF)	53
10.11. Comparación del error (RMSE) entre modelos SVR	54
11.1. Comparación de modelos finales según RMSE	55
11.2. Comparación de modelos finales según <i>R</i> ²	56
11.3. Resumen de errores con SAS	57
11.4. Resumen de errores de validación en SAS	57
11.5. Importancia de las variables según incremento en el error MSE	58
11.6. Ejemplo de explicabilidad local mediante técnica LIME	59
11.7. Árbol representativo dentro del modelo <i>Random Forest</i>	60
11.8. Coeficientes del modelo de regresión lineal	61

A.1. Resumen estadístico descriptivos	67
A.2. Gráfica dispersión 1/4	68
A.3. Gráfica dispersión 2/4	68
A.4. Gráfica dispersión 3/4	69
A.5. Gráfica dispersión 4/4	69
A.6. Diagrama SAS	70

Índice de Cuadros

4.1. Fases del ciclo CRISP-DM y su aplicación en el trabajo	10
6.1. Lenguajes de programación y entornos de desarrollo empleados	20
7.1. Variables de calendario incorporadas al conjunto de datos	25
7.2. Variables climatológicas incorporadas al conjunto de datos	25
7.3. Proporción de valores atípicos por estación	26
7.4. Variables con valores nulos detectados en el conjunto de datos	27
9.1. Valores del Factor de Inflación de la Varianza (VIF)	39
9.2. Valores del Factor de Inflación de la Varianza (VIF) final	40
9.3. Comparación del rendimiento entre el modelo original y el ajustado	41
10.1. Resultado del mejor modelo de red neuronal (nodo 5)	44
10.2. Resultados del mejor modelo de regresión lineal	45
10.3. Resultados del mejor modelo Ridge	45
10.4. Resultados del mejor modelo Lasso	46
10.5. Resultados del mejor modelo Random Forest	48
10.6. Resultados del mejor modelo XGBoost	49
10.7. Resultados del mejor modelo LightGBM	50
10.8. Resultados del mejor modelo SVR Lineal	51
10.9. Resultados del mejor modelo SVR Polinomial	52
10.10. Resultados del mejor modelo SVR Radial	53
10.11. Resultados del mejor modelo SVR (Lineal) con $C = 0,1$	54

Resumen

El crecimiento de la densidad poblacional durante los últimos años en las grandes ciudades, ha intensificado la presión sobre las infraestructuras viales y la movilidad urbana, especialmente en núcleos urbanos consolidados desde hace décadas. Esta situación ha impulsado el interés en la investigación sobre los sistemas de transporte sostenible, sirviendo como base para el presente análisis sobre la afluencia en las estaciones de metro de Madrid.

Este trabajo se basa en los registros diarios de entradas de viajeros en cada estación del metro de Madrid durante los años 2022 y 2023. El objetivo es analizar una zona específica para estudiar la influencia de las estaciones cercanas sobre una estación objetivo, Chamartín, incorporando además variables temporales y climatológicas.

Para ello, se han implementado distintos modelos de *Machine Learning*, incluyendo modelos lineales, de ensamblado, redes neuronales y métodos basados en *kernels*. El entrenamiento se ha realizado mediante un ajuste exhaustivo de hiperparámetros (*fine tuning*) y validación cruzada repetida, con el fin de identificar el modelo con mejor desempeño.

Palabras clave

Machine Learning | Fine Tuning | EDA | Minería de Datos | CRISP-DM | Sistema de Transporte Metro | Validación Cruzada Repetida | Serie Temporal | Random Forest

Abstract

The growth in population density in recent years in large cities has intensified the pressure on road infrastructure and urban mobility, especially in urban centers that have been consolidated for decades. This situation has fostered interest in research on sustainable transportation systems, serving as the basis for the present analysis of ridership patterns in the Madrid metro stations.

This study is based on daily records of passenger entries at each Madrid Metro station during the years 2022 and 2023. The main objective is to analyze a specific area in order to study the influence of nearby stations on a target station, Chamartín, while also incorporating temporal and climatological variables.

To this end, various *Machine Learning* models have been implemented, including linear models, ensemble methods, neural networks, and kernel-based techniques. The models were trained using an exhaustive hyperparameter tuning process (*fine-tuning*) and repeated cross-validation to identify the best-performing model.

Keywords

Machine Learning | Fine Tuning | EDA | Data Mining | CRISP-DM | Metro Transportation System | Repeated Cross-Validation | Time Series | Random Forest

Capítulo 1

Introducción

En este capítulo se presenta el marco introductorio del trabajo, comenzando por el contexto socio-demográfico presente en las grandes ciudades y sus implicaciones en la movilidad urbana. A partir de esta base, se plantean los retos y problemas que conlleva a la justificación del estudio y los objetivos planteados para el presente trabajo.

1.1. Contexto

Durante las últimas décadas, la población mundial ha tendido a concentrarse en las grandes ciudades, transformando profundamente el paisaje urbano y las relaciones sociales entre ellas. Este fenómeno sociodemográfico también se ha reflejado en España, en la cual, según el Ministerio de Vivienda⁹, las grandes áreas urbanas concentran el 69 % de la población española y el 76 % del empleo.

Figura 1.1: Distribución de la población en España

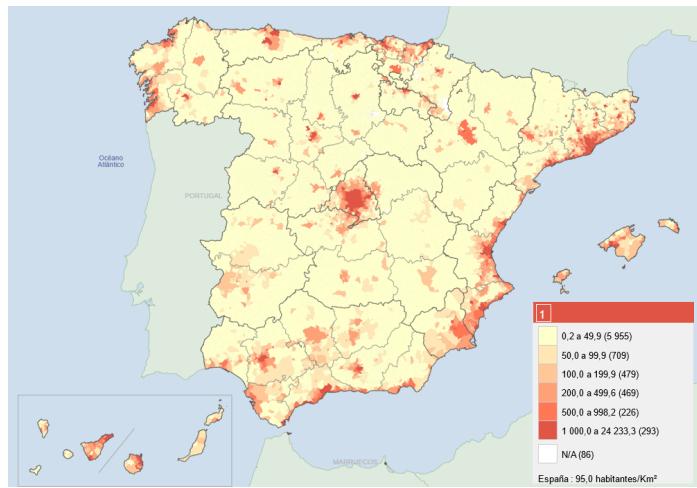


Fuente: Ministerio de Vivienda y Agenda Urbana, 2023

Este continuo crecimiento urbanístico ha convertido a estas grandes ciudades en los motores de desarrollo regional (figura 1.1), impulsando un notable crecimiento económico y la generación de nuevas oportunidades laborales para las regiones próximas.

Asimismo, la concentración poblacional favorece el avance en educación, investigación e innovación, atrayendo talento tanto nacional como internacional y enriqueciendo la oferta cultural y de servicios.

Figura 1.2: Densidad de población en España



Fuente: Ministerio de Vivienda y Agenda Urbana, 2023

Sin embargo, esta concentración poblacional también acarrea desafíos significativos, especialmente en términos de movilidad y tráfico (figura 1.2). Esto es particularmente relevante en las grandes ciudades de España, cuyo casco urbano se encuentra estructuralmente consolidado desde hace décadas, con la presencia de infraestructuras de patrimonio histórico que dificultan su modificación para adaptarse a la creciente demanda del tráfico.

El aumento de la densidad poblacional en estas áreas genera una mayor presión sobre las infraestructuras viales, especialmente en el tráfico terrestre. La congestión del tráfico, el incremento en los tiempos de desplazamiento y el impacto ambiental derivado del uso masivo de automóviles son algunos de los problemas más evidentes en estas grandes urbes.

Tomando como referencia la capital del país para el presente trabajo de investigación, la ciudad de Madrid, se enfrenta a elevados niveles de congestión vehicular, agravados por la limitación de su infraestructura vial en el centro de la ciudad, donde el trazado urbano heredado de siglos pasados dificulta su adaptación a la creciente demanda del tráfico.

Debido a su importancia como capital del país, desde 2022, Madrid fue seleccionada como una de las 100 ciudades climáticamente neutrales y *Smart* para 2030 por la

Comisión Europea¹¹. Lo que la convierte en uno de los centros de experimentación e innovación, sirviendo de ejemplo para el resto de las ciudades europeas para impulsar el desarrollo de los Planes de Movilidad Urbana Sostenible (SUMP o PMUS en español) y poder alcanzar la neutralidad de emisiones en Europa para 2050.

La puesta en marcha de este concepto en la Comunidad de Madrid se conoce como el Plan de Movilidad Sostenible de Madrid 360, el cual tiene como objetivo transformar el sistema de transporte hacia uno más sostenible, saludable, seguro y eficiente, mediante acciones como la descentralización poblacional, la implantación de Zonas de Bajas Emisiones (ZBE) y la transición a energías verdes en el transporte público⁴.

Ante este escenario, el transporte subterráneo, como el metro de Madrid, se ha convertido en una solución fundamental para mitigar estos problemas, gracias a su capacidad de transportar grandes volúmenes de pasajeros de manera rápida y eficiente. Al operar de forma independiente del tráfico rodado, también garantiza unos tiempos de viajes más predecibles y contribuye a disminuir la contaminación ambiental²⁴.

Con una de las redes más extensas y utilizadas de Europa, el metro de Madrid facilitó en 2024 el desplazamiento de más de 715,2 millones de viajeros, lo que representa aproximadamente 4 de cada 10 desplazamientos.

1.2. Planteamiento del problema

El crecimiento de la población y la elevada demanda de transporte han puesto de manifiesto diversos desafíos en el sistema de metro de Madrid, los cuales deberán ser abordados en el futuro. Entre estos retos se destacan:

- **Conexión con los centros urbanísticos periféricos:** A diferencia de la red de buses EMT, en la cual consigue acceder a todos los rincones de los municipios, la red de metro aún presenta zonas no cubierta, como los corredores radiales; o que no finalizan sus líneas en los nodos de transporte, imposibilitando el aprovechamiento completo de la capacidad que ofrece una línea de metro⁴.
- **Congestión actual y futura:** El incremento del tráfico de pasajeros en horas punta genera una gran presión sobre los sistemas de validación y dificulta la gestión de los intervalos entre trenes.
- **Planes de ampliación y mejoras en función de la demanda:** La aparición de núcleos de población concentrados en la periferia influye en la planificación de nuevas estaciones, destinadas a acercar el servicio a los usuarios con necesidades de desplazamiento.
- **Limitación en la disponibilidad de datos específicos:** Actualmente, Metro de Madrid dispone únicamente de registros diarios de entradas y carece de información sobre las salidas en cada estación, lo que impide conocer con precisión el flujo real de pasajeros en cada estación.

1.3. Justificación del trabajo

La presencia de estos desafíos, derivados del creciente número de pasajeros, resalta la importancia de desarrollar un sistema de predicción y monitorización de la demanda que facilite la toma de decisiones. Un modelo predictivo genérico y reproducible para cada estación permitiría anticipar las necesidades del sistema y optimizar la asignación de recursos.

Metro de Madrid cuenta con un amplio volumen de datos unitarios, en forma de registros diarios de entradas, lo que representa una oportunidad para aplicar técnicas de *Machine Learning*. El análisis de estos datos puede proporcionar información de gran relevancia, sirviendo como base para evaluar la viabilidad de crear modelos predictivos. Además, este enfoque metodológico ofrece un caso real aplicable a otras entidades que disponen de datos limitados y carecen de variables independientes externas para desarrollar modelos más sofisticados.

Capítulo 2

Estado del arte

La creciente demanda de los sistemas de transporte público a nivel global en las grandes urbes ha repercutido en el interés de la comunidad de investigación por encontrar soluciones que optimicen el servicio y permitan el desarrollo del Sistema de Transporte Inteligente (ITS). Con el objetivo de conocer los enfoques principales de estudios en los últimos años, se ha realizado una revisión de la bibliografía existente, publicada en libros, revistas científicas, artículos originales y artículos de revisión.

Uno de los problemas principales en la investigación sobre el sistema de transporte es la limitación de datos recopilados, los cuales se basan principalmente en encuestas periódicas y datos genéricos, lo que resulta en conclusiones generales al carecer de grandes bases de datos. Es por ello que una de las soluciones implementadas para mejorar la precisión y validación de los resultados es la aplicación de Inteligencia Artificial mediante modelos de Machine Learning³⁴.

Por otro lado, la investigación revisada por Behrooz, H. y Hayeri, Y. M.³, que cuenta con más de 100 referencias bibliográficas sobre la implementación de Machine Learning en sistemas de transporte, ha mostrado que el 74 % de los artículos se ha centrado en el estudio de la predicción a corto plazo. En esta revisión se ha categorizado el enfoque de la investigación en dos grupos principales: el primero, con los factores espaciales relacionados con la conectividad, el enrutamiento y la direccionalidad; mientras que el segundo, con los factores externos como la condición climática, la hora del día, la época del año, los accidentes y el nivel socioeconómico. Cabe destacar que muchos de los estudios publicados tienden a ignorar uno o ambos grupos de factores, cuando lo ideal sería considerarlos para la elaboración de modelos robustos.

Según Torre, R., Corlu, C.G., Faulin, J., Onggo, B.S. y Juan, A.A.⁸, mediante la revisión de más de 20 artículos indexados en la base de datos bibliográfica Scopus durante la última década, se ha demostrado que las metodologías de simulación y optimización son las más utilizadas para abordar el tema del “transporte sostenible”, siendo este uno de los objetivos principales de los estudios relacionados con

la movilidad. Una gran parte de las investigaciones pasadas se ha enfocado en el análisis de los sistema de transporte terrestre, en particular en la predicción del tráfico o del tiempo de viaje basado en datos históricos y actuales de los vehículos de predicción¹⁰.

En cuanto a la predicción del tráfico, Boukerche, A. y Wang, J.⁵ han detectado que los modelos de Machine Learning ofrecen un buen rendimiento gracias a sus menores restricciones en las tareas de predicción y a su mejor desempeño en modelos no lineales. Cabe señalar que algunos modelos de Machine Learning y Deep Learning presentan un desempeño destacado en la generación de modelos predictivos,

tales como la regresión lineal, k-vecinos más próximos (KNN), máquina de vectores de soporte (SVM), codificadores automáticos (AE) y diversas arquitecturas de redes neuronales, incluyendo redes recurrentes (RNN), redes convolucionales (CNN), redes profundas (DNN), aprendizaje por refuerzo profundo (DRL), entre otros¹⁴.

Por lo tanto, en la práctica, la investigación enfocada en los sistemas de transporte continúa enfrentándose al problema de la limitación de datos para generar modelos predictivos. Sin embargo, esta revisión evidencia la viabilidad del uso de modelos de Machine Learning y Deep Learning para la predicción sobre este campo.

A partir de artículos y bibliografía científica, se observa que la mayoría de los estudios se centran en el análisis del tráfico terrestre como el servicio de autobuses. En contraste, el análisis del sistema de transporte público subterráneo, como el metro, es menos común, y aún menos se ha estudiado la interacción entre factores espaciales y externos, tal como lo describen Behrooz, H. y Hayeri, Y. M.³, para analizar la relación de las estaciones y otros factores sobre la afluencia de una estación.

Capítulo 3

Objetivos

Tras una revisión exhaustiva del estado del arte sobre el uso de técnicas de Machine Learning en la predicción de los sistemas de transporte público, el objetivo de este trabajo se pretende encontrar el modelo óptimo de Machine Learning para estimar la afluencia de la entrada de una de las estaciones del Metro de Madrid.

Debido a la gran cantidad de estaciones que conforman la red de metro de Madrid, para llevar a cabo el análisis se procederá a realizar el estudio por áreas, definido en este caso por la estación objetivo: Chamartín.

Dado que los datos disponibles se limitan a los registros de entradas en cada estación, el estudio se enfocará en determinar si existe una relación directa entre la cantidad de pasajeros que acceden a una estación y la afluencia registrada en las estaciones próximas.

Además de las variables espaciales, seleccionadas mediante distancia lineal en un radio de 3 km respecto a la estación objetivo, se han incorporado nuevas variables temporales como el tipo de día y los días de la semana, así como variables climatológicas como la temperatura, precipitación, viento, etc. extraídas de fuentes públicas. Asimismo, dado que se trata de una base de datos de tipo serie temporal, se han incorporado variables rezagadas mediante “lags” correspondientes a distintos períodos temporales.

De esta forma, se analizará la interdependencia entre la estación objetivo y las estaciones adyacentes, así como la influencia de otros factores externos de carácter temporal o climático que puedan afectar al uso del metro de Madrid, repercutiendo en la variación del número de entradas en una estación determinada.

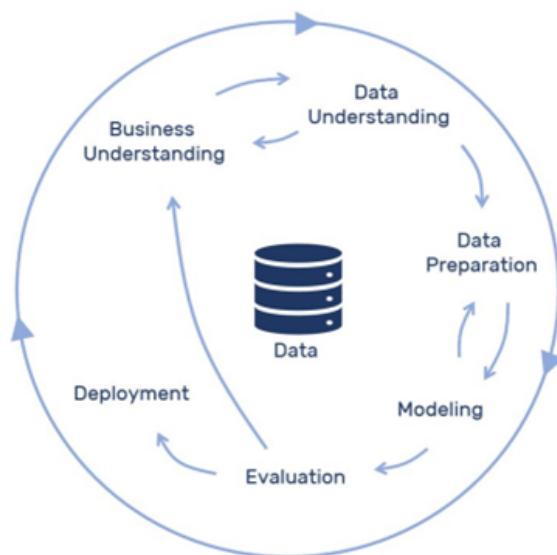
Capítulo 4

Metodología

Para seguir un criterio lógico y ordenado en el proceso de investigación, se ha adoptado la metodología CRISP-DM en el presente trabajo. La metodología Cross-Industry Standard Process for Data Mining, definida por sus siglas CRISP-DM, es considerado como uno de los métodos de referencia en los proyectos de minería de datos.

Basado en una estructura jerárquicos, donde se dividen las distintas fases del proyecto hasta su despliegue final. Esta estructura mantiene una secuencia principal de trabajo predefinido, pero permite el desarrollo de procedimientos de retroalimentación para realizar ajustes y mejoras en las diferentes fases críticas del proceso, conformando así un ciclo de vida completo para el desarrollo del proyecto²⁸.

Figura 4.1: Modelo CRISP-DM



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

El ciclo genérico de la metodología CRISP-DM (figura 4.1), está definido por seis fases principales, unidas por flechas que indican las dependencias y relaciones de retroalimentación entre ellas²⁹. Para el presente trabajo, estas fases se pueden interpretarse de la siguiente forma:

1. **Business Understanding:** Fase inicial de comprensión del negocio, que en muchos de los casos se desarrolla de manera paralela con la siguiente etapa de comprensión de los datos. En esta fase se define el objetivo del proyecto, así como las fuentes de información y herramientas necesarias para su desarrollo.
2. **Data Understanding:** Fase centrada en la comprensión de los datos disponibles. En este proceso se identifican los datos a utilizar y se analiza su relevancia, estado y consistencia. Una de las metodologías más empleadas es el Análisis Exploratorio de Datos (EDA), que agrupa un conjunto de técnicas destinadas a obtener una comprensión profunda sobre el estado de los datos.
3. **Data Preparation:** Fase previa al modelado, enfocada en el análisis exhaustivo y la preparación del conjunto de datos. En ella se llevan a cabo las tareas de limpieza, depuración, transformación de los datos. Además, en el presente trabajo, la metodología EDA se mantiene activa durante toda esta fase, en un proceso iterativo guiado por la retroalimentación de los resultados intermedios, con el fin de garantizar la idoneidad del conjunto de datos antes de la modelización.
4. **Modelling:** Fase principal del trabajo, dedicada a la aplicación de técnicas de Machine Learning sobre el conjunto de datos ya preparado. En esta etapa se establecen los criterios de entrenamiento y se realiza el ajuste de hiperparámetros, con el objetivo de identificar el modelo con mejor rendimiento predictivo.
5. **Evaluation:** Fase dedicada a la evaluación de los resultados obtenidos a partir de métricas predefinidas para comparar el desempeño de cada modelo. A partir de esta comparación, se selecciona el modelo final.
6. **Deployment:** Fase final de despliegue, abordada en este trabajo desde una perspectiva conceptual, centrada en la interpretación del modelo seleccionado y su viabilidad para una futura implementación en un entorno productivo.

La metodología CRISP-DM aplicada en este trabajo se resume en el cuadro 4.1:

Cuadro 4.1: Fases del ciclo CRISP-DM y su aplicación en el trabajo

Fase CRISP-DM	Aplicación en el trabajo
1. Comprensión de negocio	<ul style="list-style-type: none"> - Introducción - Estado del arte - Objetivo
2. Comprensión de datos	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - <i>Modelos individuales lineales:</i> <ul style="list-style-type: none"> · Regresión lineal · Lasso/Ridge - <i>Modelos de ensamblado:</i> <ul style="list-style-type: none"> · Random Forest · XGBoost · LightGBM - <i>Modelos no lineales:</i> <ul style="list-style-type: none"> · Redes Neuronales · SVM (SVR)
5. Evaluación	<ul style="list-style-type: none"> - Validaciones cruzadas - Métricas: R^2, RMSE
6. Despliegue	<ul style="list-style-type: none"> - Interpretación y muestreo del modelo final

Fuente: Elaboración propia.

Capítulo 5

Fundamentos teóricos

Una vez definido el esquema de trabajo y los modelos para utilizar en la fase de modelado, en este apartado se procede a describir brevemente en qué consiste cada método y cómo se aplica en el contexto del presente estudio.

Los modelos seleccionados se agrupan en tres categorías generales. En primer lugar, se encuentran los modelos individuales lineales, como la regresión lineal y sus variantes penalizadas: Lasso (regresión con penalización L1) y Ridge (regresión con penalización L2).

En segundo lugar, se consideran los modelos basados en técnicas de ensamblado, que incluyen Random Forest (bagging de árboles), XGBoost (boosting) y LightGBM (boosting optimizado mediante histogramas).

Por último, se han incorporado modelos no lineales, como las redes neuronales artificiales y los modelos con funciones kernel, representados en este trabajo por el algoritmo SVR.

La selección de estos modelos supervisados se ha basado en el objetivo de comparar el rendimiento de distintos enfoques de Machine Learning frente a modelos tradicionales como la regresión lineal, teniendo en cuenta las características técnicas del conjunto de datos: el tamaño de la muestra, el tipo de variables utilizadas, y el propósito analítico del estudio, centrado en identificar la relación y la influencia de variables explicativas sobre la variación diaria de la afluencia en una estación de metro.

5.1. Modelos individuales lineales

Los modelos lineales constituyen una categoría de modelos caracterizados por explicar el comportamiento de la variable dependiente (objetivo) a partir de una combinación lineal de variables independientes (explicativas).

5.1.1. Regresión lineal

El modelo de *regresión lineal múltiple* constituye una de las herramientas más empleadas en los últimos años como referencia para la predicción de una variable numérica \hat{y} , a partir de un conjunto de variables independientes x_i . Este modelo proporciona una ecuación que permite estimar el valor de la variable dependiente en función de las variables explicativas, reflejando la influencia de cada una de ellas sobre la variable objetivo¹⁶. La ecuación general del modelo se expresa de la siguiente manera:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \quad (5.1)$$

Donde:

- \hat{y} representa el valor estimado de la variable objetivo.
- β_0 es el intercepto (valor de \hat{y} cuando todas las x_i toman valor 0).
- β_i corresponde con el coeficiente de la regresión (pesos) de cada x_i sobre \hat{y} .
- x_i corresponde a cada una de las variables explicativas del modelo.

5.1.2. Lasso

El modelo de *Lasso* se puede entender como una extensión del modelo de regresión lineal previamente definido, con la particularidad de que incorpora una penalización tipo L_1 sobre los coeficientes de las variables explicativas. Esta penalización tiene como finalidad mejorar la capacidad de generalización del modelo y facilitar la selección automática de variables, reduciendo a cero aquellos coeficientes cuya contribución sea irrelevante¹⁶.

La estimación del modelo *Lasso* se puede mediante la siguiente expresión matemática:

$$\hat{\beta}^{lasso} = \arg \min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\} \quad (5.2)$$

Donde:

- λ es el parámetro de regularización que controla la magnitud de la penalización aplicada a los coeficientes.

- El término $\sum_{j=1}^p |\beta_j|$ corresponde a la norma L_1 de los coeficientes del modelo, que promueve la reducción de algunos de ellos a cero.

5.1.3. Ridge

El modelo *Ridge* constituye otra extensión derivada del modelo de regresión lineal, en la que se incorpora una penalización de tipo L_2 sobre los coeficientes de las variables explicativas. A diferencia del modelo *Lasso*, *Ridge* no reduce los coeficientes a cero, por lo que no realiza una selección automática de variables; sin embargo, sí atenúa su magnitud con el fin de evitar el sobreajuste y mejorar la capacidad de generalización.

Este tipo de regularización resulta especialmente útil en situaciones de alta multicolinealidad o cuando el número de predictores es elevado en relación con el número de observaciones, ya que permite estabilizar la estimación de los coeficientes sin necesidad de eliminar variables del modelo¹⁶.

La estimación del modelo *Ridge* se expresa mediante la siguiente formulación:

$$\hat{\beta}^{ridge} = \arg \min_{\beta} \left\{ \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\} \quad (5.3)$$

Donde:

- λ es el parámetro de regularización que controla la intensidad de la penalización aplicada sobre los coeficientes.
- El término $\sum_{j=1}^p \beta_j^2$ corresponde a la norma L_2 de los coeficientes del modelo, cuya función es restringir la magnitud de estos para mejorar la estabilidad y generalización del modelo.

5.2. Modelos basados en técnicas de ensamblado

Los modelos supervisados basados en técnicas de *ensamblado* (ensemble learning) constituyen una categoría de algoritmos que combinan múltiples predicciones de los estimadores base generados a partir de modelos individuales, con el objetivo de mejorar el rendimiento predictivo respecto a cualquier modelo aplicado de forma individual. Esta aproximación parte de la premisa de que la agregación de múltiples predictores puede reducir la varianza, el sesgo o ambos, dependiendo de la técnica empleada, incrementando así la robustez y capacidad de generalización del modelo resultante²⁷.

En el contexto del aprendizaje supervisado, que abarca tanto tareas de clasificación como de regresión, como es el caso de este trabajo, estos modelos se agrupan en dos enfoques principales:

- **Bagging (Bootstrap Aggregating)**: Método que consiste en generar múltiples subconjuntos de entrenamiento mediante muestreo con reemplazo. A partir de cada subconjunto se entrena un modelo base y, posteriormente, se combinan sus predicciones. Su objetivo principal es la reducción de la varianza del modelo¹⁵. Siendo el algoritmo *Random Forest*, el representante más destacado.
- **Boosting**: Técnica basada en el entrenamiento secuencial de modelos, donde cada nuevo modelo intenta corregir los errores cometidos por los anteriores. Este procedimiento permite que el modelo compuesto mejore progresivamente su rendimiento¹⁵. En este trabajo se analizarán los modelos *XGBoost* y *LightGBM* como referencia de este método.

5.2.1. Random Forest

El modelo *Random Forest*, propuesto por Breiman⁶, es una de las técnicas más consolidadas dentro del aprendizaje supervisado mediante *bagging*. Su funcionamiento se basa en construir múltiples árboles de decisión a partir de subconjuntos aleatorios de entrenamiento generados mediante muestreo con reemplazo (*bootstrap*). Además, en cada partición de los nodos, se selecciona aleatoriamente un subconjunto de variables, lo que reduce la correlación entre árboles y mejora la generalización del modelo.

Cada árbol base genera una predicción, y el resultado final se obtiene como promedio (regresión) o voto mayoritario (clasificación) de todas las predicciones. Esta agregación permite reducir la varianza y mitigar el sobreajuste, siendo especialmente eficaz en contextos con muchas variables y relaciones complejas.

Aunque *Random Forest* es robusto por diseño, en este trabajo se ajustan sus hiperparámetros clave para optimizar el rendimiento.

- **mtry**: número de variables predictoras seleccionadas aleatoriamente en cada división del nodo.
- **ntree**: número total de árboles generados.
- **nodesize**: número mínimo de observaciones en cada nodo terminal.

5.2.2. XGBoost

El modelo *XGBoost* (Extreme Gradient Boosting)⁷ es una técnica de aprendizaje supervisado basada en *boosting* que construye árboles de decisión de forma secuencial, donde cada nuevo árbol busca corregir los errores del anterior.

Diseñado para optimizar tanto la precisión como el rendimiento computacional, *XGBoost* ha demostrado ser altamente eficaz en tareas de regresión con relaciones complejas y gran cantidad de variables. Su éxito se debe, en parte, a características como la regularización L1 y L2, el tratamiento automático de valores perdidos y la optimización por bloques, que lo hacen especialmente robusto y escalable.

En este trabajo se realizará el fine tuning de los siguientes hiperparámetros¹⁹:

- **nrounds**: número de iteraciones o árboles.
- **eta**: tasa de aprendizaje (*learning rate*), controla la contribución de cada nuevo árbol.
- **max_depth**: profundidad máxima de cada árbol.
- **min_child_weight**: número mínimo de observaciones requeridas en un nodo hijo para dividirse.

5.2.3. LightGBM

El modelo *LightGBM* (Light Gradient Boosting Machine)¹² es una versión optimizada del algoritmo de *Gradient Boosting*, diseñada para acelerar el proceso de entrenamiento sin comprometer la precisión. A diferencia de otras variantes como *XGBoost*, emplea un crecimiento de árbol basado en hojas (*leaf-wise*) y utiliza histogramas discretizados, lo que mejora su eficiencia computacional y reduce el uso de memoria¹⁸.

Estas características hacen de *LightGBM* una herramienta especialmente adecuada para conjuntos de datos grandes o con alta dimensionalidad, mostrando un rendimiento competitivo incluso en problemas con relaciones no lineales complejas.

En este trabajo, se aplicará *LightGBM*, ajustando los siguientes hiperparámetros:

- **nrounds**: número total de iteraciones o árboles generados.
- **learning_rate**: tasa de aprendizaje, que controla el impacto de cada nuevo árbol.
- **num_leaves**: número máximo de hojas permitidas por árbol.
- **min_data_in_leaf**: número mínimo de observaciones requeridas en una hoja terminal.

5.3. Modelos no lineales complejas y tipo kernel

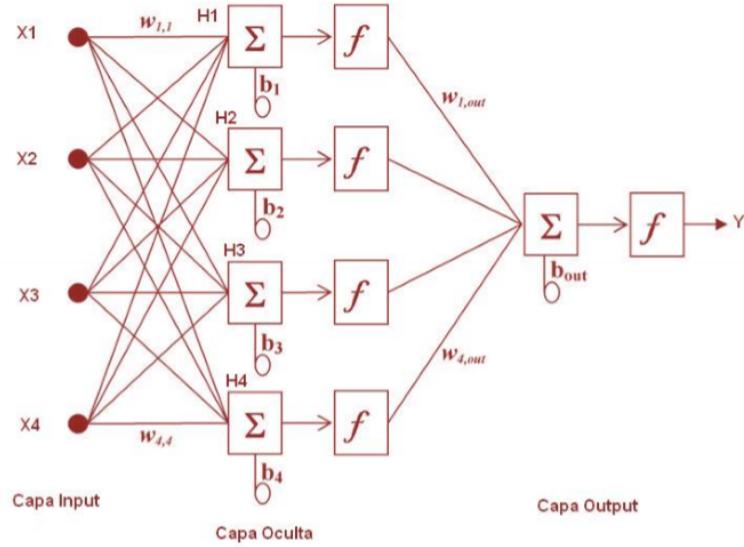
Además de los modelos lineales y los basados en ensamblado, existen algoritmos capaces de capturar relaciones no lineales complejas entre las variables explicativas y la variable objetivo. Entre ellos destacan las Redes Neuronales Artificiales y los modelos tipo kernel, como la Regresión por Vectores de Soporte (SVR).

5.3.1. Redes neuronales

El modelo de red neuronal representa un algoritmo de aprendizaje automático inspirado en el funcionamiento del cerebro humano, en el que la información se procesa mediante una red de nodos conectados entre sí³⁵. En el presente trabajo se emplea una Red Neuronal Artificial (ANN) como herramienta para modelar la relación entre un conjunto de variables predictoras (inputs) y la variable objetivo (output).

La figura 5.1 muestra la configuración básica de una red neuronal artificial del tipo perceptrón multicapa (MLP), compuesta por una capa de entrada con cuatro varia-

Figura 5.1: Configuración básica de una red neuronal artificial



Fuente: Elaboración propia

bles predictoras (X_1, X_2, X_3, X_4), una capa oculta con cuatro nodos (H_1, H_2, H_3, H_4) y una capa de salida.

Cada neurona de la capa oculta recibe una combinación lineal ponderada de todas las entradas, mediante pesos sinápticos $w_{i,j}$, a la que se añade un término de sesgo b_j . Esta suma se transforma posteriormente mediante una función de activación no lineal, como ReLU o sigmoide, permitiendo así al modelo capturar las relaciones complejas entre las variables³³.

Las salidas generadas por la capa oculta se combinan nuevamente con pesos y un sesgo adicional para obtener la salida final, que en problemas de regresión suele calcularse mediante una función lineal, devolviendo así la predicción estimada \hat{Y} .

Durante el proceso de entrenamiento, la red ajusta los pesos y sesgos mediante un procedimiento iterativo denominado retropropagación del error (*backpropagation*), cuyo objetivo es minimizar la diferencia entre las predicciones y los valores observados. Esta capacidad de aprendizaje confiere a las redes neuronales una notable eficacia para aproximar funciones no lineales en entornos complejos.

En el presente trabajo, se ha realizado el (*fine tuning*) de los siguientes hiperparámetros:

- **size**: número de nodos en la capa oculta.
- **decay**: parámetro de regularización que penaliza la magnitud de los pesos para evitar el sobreajuste.
- **maxit**: número máximo de iteraciones (epochs) permitidas.

5.3.2. SVR

El modelo de Soporte Vectorial para Regresión (SVR) es una extensión de las Máquinas de Soporte Vectorial (SVM) orientada a problemas de regresión. Su objetivo es encontrar una función que se desvíe como máximo en ε unidades del valor real de cada observación, manteniendo al mismo tiempo la menor complejidad del modelo³¹.

Una de las principales fortalezas del SVR es su capacidad para capturar relaciones no lineales complejas entre variables mediante funciones *kernel*, que transforman los datos a un espacio de mayor dimensión donde las relaciones son más fácilmente modelables. Los kernels más utilizados son el lineal, el polinomial y el radial (RBF).

Durante el entrenamiento, el algoritmo selecciona un subconjunto de observaciones, los *vectores soporte*, que definen el modelo final. Esta característica mejora la generalización y reduce el sobreajuste al centrarse en los puntos más relevantes. En este trabajo se han implementado las tres variantes de SVR, realizando un ajuste exhaustivo (*fine tuning*) de los hiperparámetros clave para cada kernel:

- **C**: parámetro de regularización presente en los tres métodos de SVR, que controla el equilibrio entre la penalización de errores y la amplitud del margen.
- **degree**: exclusivo del kernel polinomial, indica el grado del polinomio utilizado.
- **scale**: también asociado al kernel polinomial, ajusta la escala del término de entrada del polinomio, afectando la flexibilidad del modelo.
- **sigma**: utilizado en el kernel radial, determina la amplitud de la función de base radial, regulando la influencia de cada punto de entrenamiento sobre la predicción.

5.4. Métodos de evaluación

Una vez definidos los conceptos teóricos de cada uno de los modelos utilizados en este trabajo, así como los hiperparámetros que han sido objeto de un proceso exhaustivo de ajuste, en esta sección se describe el procedimiento empleado para su entrenamiento y las métricas utilizadas en la evaluación comparativa: R^2 y RMSE.

5.4.1. Validación cruzada repetida

La validación cruzada es una de las técnicas más utilizadas para comparar el rendimiento de modelos predictivos. Su principal objetivo es evaluar la estabilidad de los resultados mediante el uso de diferentes subconjuntos del conjunto de datos, lo que permite estimar de forma más robusta su capacidad predictiva sobre datos no observados.

Dado el tamaño del conjunto de datos utilizado y el enfoque del estudio, se ha optado por no aplicar la clásica división del conjunto en entrenamiento y testeo. En su lugar,

se ha empleado una estrategia de validación cruzada repetida con K pliegues (K -folds), configurada con 10 pliegues y 10 repeticiones.

Este procedimiento implica dividir el conjunto de datos en 10 subconjuntos diferentes, en los cuales, para cada iteración, se entrena con $K - 1$ folds y se evalúa sobre el fold restante. Este proceso se repite 10 veces, cada una con una partición distinta, dando lugar a 100 combinaciones entrenamiento-validación²³. Esta estrategia permite estimar con mayor precisión la capacidad de generalización del modelo y reducir la varianza asociada a una única partición, ofreciendo resultados más estables y fiables sin requerir un conjunto exclusivo de test.

5.4.2. Coeficiente de determinación (R^2)

Dado que la variable objetivo es numérica, una de las métricas más relevantes para evaluar el rendimiento de un modelo de regresión es el coeficiente de determinación (R^2), que cuantifica la proporción de la varianza total de la variable dependiente explicada por las variables independientes del modelo.

La expresión matemática del R^2 es la siguiente:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (5.4)$$

Donde:

- y_i : valor real de la observación i
- \hat{y}_i : valor predicho por el modelo para la observación i
- \bar{y} : media de los valores reales y_i
- n : número total de observaciones

Su valor oscila entre 0 y 1, donde un valor cercano a 1 indica un ajuste casi perfecto, es decir, una alta capacidad explicativa del modelo sobre los datos observados³².

5.4.3. Raíz del error cuadrático medio (RMSE)

Según el criterio de evaluación adoptado en este trabajo, y dado que las variables han sido estandarizadas (media cero y varianza unitaria), la métrica más adecuada para evaluar el rendimiento es la raíz del error cuadrático medio (RMSE), ya que permite interpretar la magnitud del error en las unidades que los datos estandarizados²⁶.

La expresión matemática del RMSE es la siguiente:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5.5)$$

Donde:

- y_i : valor real de la observación i
- \hat{y}_i : valor predicho por el modelo para la observación i
- n : número total de observaciones

El valor mínimo posible del RMSE es 0, lo que indica una predicción perfecta. Por tanto, cuanto menor sea el RMSE, más preciso será el modelo.

En el proceso de comparación de los modelos analizará de manera complementaria el (R^2), que evalúa la proporción de variabilidad explicada, y el RMSE, con el error medio cometido

Capítulo 6

Entorno de desarrollo

Para llevar a cabo la tarea de investigación planteada en el presente trabajo, el desarrollo y la ejecución de todas las fases se han realizado desde un equipo con sistema operativo Windows 11, empleando tres entornos de desarrollo integrados (IDE) diferentes, correspondientes a sus respectivos lenguajes de programación. De este modo, se han puesto en práctica los conocimientos adquiridos durante el máster en un contexto aplicado a un caso real.

Cuadro 6.1: Lenguajes de programación y entornos de desarrollo empleados

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

Fuente: *Elaboración propia.*

La división del trabajo en la que se ha empleado cada uno de los IDE se puede resumir de la siguiente manera (cuadro 6.1). Para la fase de análisis exploratorio y preparación de los datos, se ha utilizado principalmente Python desde PyCharm, debido a que ofrece una clara ventaja en la visualización en tiempo real de los cambios efectuados sobre el conjunto de datos, de forma paralela al código, por lo que no se requiere abrir nuevas ventanas para visualizar los cambios. Además, la función de resaltar el último conjunto de datos modificado facilita el seguimiento del estado de transformación, especialmente cuando se asignan nombres distintos a los objetos en función de cada etapa del proceso, como se ha hecho en este trabajo.

Para la parte de modelado y aplicación de técnicas de *Machine Learning*, se ha utilizado el lenguaje R en el entorno RStudio. Una de las principales ventajas de esta herramienta es su facilidad de uso en tareas estadísticas, así como la agilidad que ofrece para realizar comprobaciones puntuales dentro del flujo de trabajo. Además,

dispone de la librería caret, que integra una amplia variedad de modelos de *Machine Learning*, y sobre la cual se basa este trabajo. Esta librería permite ajustar los hiperparámetros y obtener resultados bajo un formato unificado, lo que facilita la comparación directa entre modelos sin necesidad de procesamientos adicionales.

Por último, la herramienta SAS, al no disponer de todos los modelos de *Machine Learning* que se pretenden aplicar en este trabajo, se ha empleado únicamente de forma complementaria al final del estudio, con el fin de contrastar el comportamiento y la estabilidad de algunos de los modelos utilizados, usando el IDE visual de SAS Enterprise Miner Workstation.

Capítulo 7

Exploración inicial del conjunto de datos

Siguiendo con el procedimiento de comprensión de los datos, en este apartado se desarrollará una descripción detallada sobre los orígenes y fuentes de los conjuntos de datos empleados en el presente trabajo. A partir de las diferentes tablas de datos, se realiza una fusión mediante la variable identificadora, y se procederá con algunos análisis descriptivos iniciales para conocer el estado del conjunto de datos, el cual se continuará desarrollando de forma paralela con el siguiente apartado de preparación hasta obtener un *dataset* suficientemente estable para su posterior modelización.

7.1. Orígenes y fuentes de datos

Para el desarrollo del presente trabajo se han obtenido registros de entrada de viajeros en cada una de las estaciones del Metro de Madrid, con periodicidad diaria, abarcando el periodo comprendido entre el 1 de enero de 2022 y el 31 de diciembre de 2023. Dada su granularidad —registros diarios de cada estación según las entradas disponibles— esta información no se encuentra disponible en los repositorios públicos ni en los portales de datos abiertos, por lo que se considera un conjunto de datos de carácter interno, cedido para fines del desarrollo de proyectos de investigación en los que participan los tutores académicos de la Facultad de Informática de la Universidad Complutense de Madrid.

Tal como se ha explicado en la sección 1.2 sobre el planteamiento del problema, el sistema de Metro de Madrid dispone únicamente de tornos de registro de entrada en cada estación (según las puertas habilitadas para ello), pero no cuenta con un sistema de peaje cerrado que permita registrar las salidas de los viajeros, lo que implica una limitación estructural en el seguimiento del recorrido completo del viajero.

Además, con el objetivo de estudiar la relación espacial entre las estaciones de metro, se han incorporado los datos de localización geográfica (longitud y latitud) de

cada estación. Esta información permitirá analizar la proximidad entre estaciones y modelar la posible dependencia espacial en relación con la estación objetivo seleccionada.

Para enriquecer el conjunto de variables predictoras, se han incorporado datos climatológicos mediante el acceso a la API pública (API Key) proporcionada por la plataforma AEMET Open Data¹.

Dado que la variable identificadora del conjunto de datos es la fecha con registros diarios, se ha considerado pertinente incluir información de tipo temporal para estudiar la influencia del tipo de día en la afluencia de viajeros al metro. Para ello, se han empleado datos del calendario laboral de Madrid, obtenidos desde el Portal de Datos Abiertos del Ayuntamiento de Madrid, con un rango temporal comprendido entre 2013 y 2025².

7.2. Integración del conjunto de datos

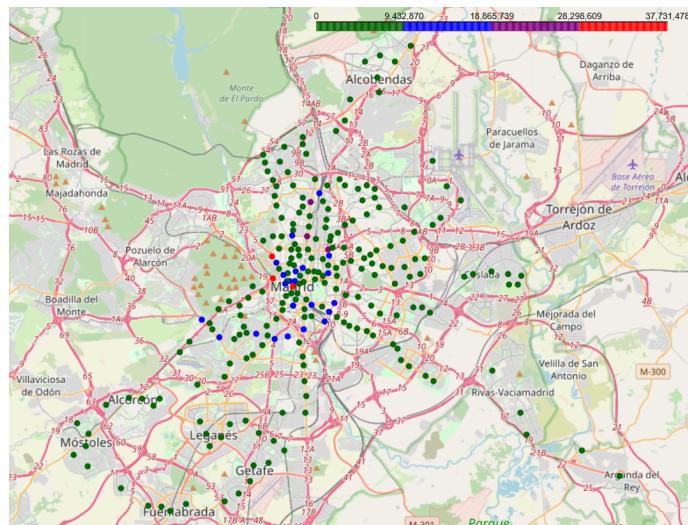
La obtención de los datos a partir de diversas fuentes conlleva la necesidad de integrar información proveniente de archivos en diferentes formatos, estructuras de variables y referencias temporales. Por este motivo, se accede a cada uno de los conjuntos de datos mediante consultas específicas en el entorno PyCharm, estableciendo como variable identificadora común una fecha estandarizada en formato YYYY-MM-DD, comprendido entre el 1 de enero de 2022 y el 31 de diciembre de 2023.

En relación con el conjunto de datos correspondiente a la entrada diaria de viajeros en cada estación de metro, originalmente las entradas de cada acceso de una misma estación se encontraban registradas en filas separadas. Para facilitar el análisis, se agrupan y consolidan todas las entradas por estación y fecha, pivotando la información para convertir cada estación en una variable independiente (columna), cuyo valor representa el número total de viajeros registrados ese día.

7.3. Selección espacial mediante distancias geográficas

La transformación del conjunto de datos anterior en formato ancho genera un total de 249 estaciones (figura 7.1) representadas como variables explicativas, lo que introduce un riesgo elevado de multicolinealidad y complica la interpretación del modelo, especialmente dada la dimensión relativa de los datos.

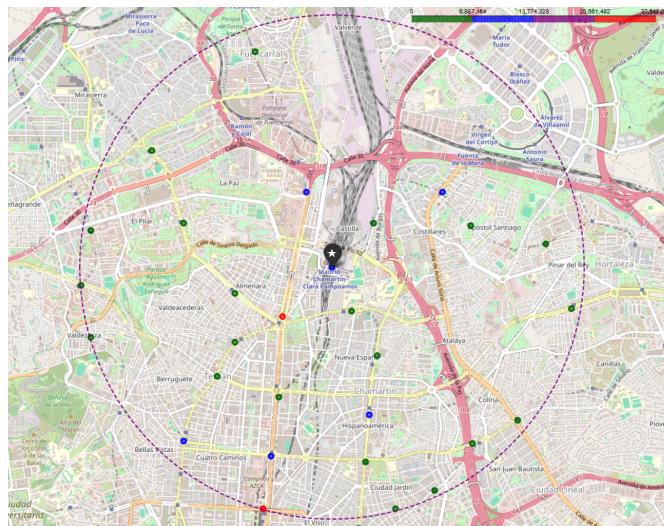
Figura 7.1: Estaciones totales de Metro Madrid



Fuente: Elaboración propia

Con el objetivo de mitigar este problema, se adopta un criterio de selección espacial centrado en la estación objetivo “Chamartín”, focalizando el análisis únicamente en las estaciones más próximas a esta. Para ello, se incorporan las coordenadas geográficas (latitud y longitud) de cada estación, y se calcula la distancia lineal entre cada una de ellas y la estación de referencia mediante fórmulas de distancia geodésica.

Figura 7.2: Estaciones próximos a 3km de radio de Chamartín



Fuente: Elaboración propia

Con ello, se estableció un criterio de selección basado en un radio de 3 kilómetros (figura 7.2), lo que permite identificar y conservar únicamente aquellas estaciones ubicadas dentro de este rango para su inclusión en el conjunto final de modelado.

7.4. Conjunto de datos inicial

El conjunto de datos inicial está compuesto por un total de 730 observaciones (correspondientes a días entre 2022 y 2023) y 37 columnas. A continuación, se describen las variables seleccionadas provenientes de los distintos *datasets* utilizados: calendario laboral, datos climatológicos y estaciones de metro.

Del conjunto de datos de calendario se han incorporado 2 variables de tipo temporal (cuadro 7.1):

Cuadro 7.1: Variables de calendario incorporadas al conjunto de datos

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

Fuente: *Elaboración propia.*

Del dataset climatológico se han seleccionado 5 variables (cuadro 7.2):

Cuadro 7.2: Variables climatológicas incorporadas al conjunto de datos

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

Fuente: *Elaboración propia.*

Finalmente, para el conjunto de datos principal, considerando únicamente las estaciones situadas en un radio de 3 km desde la estación objetivo Chamartín, se han seleccionado un total de 29 estaciones, incluida la propia Chamartín.

7.5. Análisis exploratorio de datos (EDA) inicial

Realizando una exploración inicial mediante estadísticas descriptivas básicas del conjunto de datos (véase figura A.1 en el apéndice de esta memoria), se observa la presencia de posibles valores atípicos, así como ciertos valores nulos en algunas variables.

7.5.1. Recuento de valores atípicos (*outliers*)

Un valor atípico (*outliers*) es una observación que se desvía de forma significativa del comportamiento general del conjunto de datos. Estas anomalías pueden deberse a errores de medición, a una variabilidad natural del fenómeno observado o a eventos poco frecuentes. En el contexto de la minería de datos, la detección y el tratamiento de *outliers* es una tarea fundamental, ya que pueden distorsionar el análisis y deteriorar el rendimiento de los modelos predictivos.

Para estimar la proporción de valores atípicos en las variables numéricas, se ha empleado una función personalizada basada en dos criterios: el primero depende de la simetría de la distribución (utilizando Z-score o MAD según corresponda), y el segundo se basa en el rango intercuartílico (IQR). Solo se consideran *outliers* aquellos valores que cumplen simultáneamente ambos criterios. Estos valores son reemplazados por valores nulos (NA's / NaN) para ser tratados en etapas posteriores de la preparación de datos.

Cuadro 7.3: Proporción de valores atípicos por estación

Variable	Proporción de valores atípicos
Santiago Bernabéu	0.00137

Fuente: Elaboración propia.

Como resultado, bajo este criterio de detección empleado, únicamente se ha identificado una pequeña proporción de observaciones atípicas (0.00137) en la estación de Santiago Bernabéu (cuadro 7.3). Este bajo porcentaje sugiere que se trata de una anomalía puntual y poco frecuente, por lo que se ha procedido a sustituir dichos valores por NA's para su posterior tratamiento durante la fase de depuración.

7.5.2. Recuento de Valores Nulos (NA's)

Tras realizar un recuento de las variables con valores nulos en el conjunto de datos, se han identificado un total de 7 variables que presentan este tipo de registros (cuadro 7.4). Estas variables se detallan a continuación:

Cuadro 7.4: Variables con valores nulos detectados en el conjunto de datos

Variable	Nº 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

Fuente: *Elaboración propia.*

Capítulo 8

Análisis y preparación de datos

Una vez realizado el Análisis Exploratorio de Datos inicial, en este apartado se abordarán los procedimientos necesarios para solucionar los problemas presentes en el *dataset*, previo a la aplicación de los modelos de aprendizaje automático.

8.1. Imputación de valores NA's

Dado que el conjunto de datos presenta variables con distintas proporciones de valores nulos (NA), se han aplicado tres estrategias de imputación, en función del patrón de ausencia observado:

- Para variables con valores nulos puntuales, como “velmedia”, “sol” y “Santiago Bernabéu”, se ha optado por imputar los valores NA con la media aritmética entre el valor del día anterior y el posterior.
- En el caso de las estaciones “Concha Espina” y “Cruz del Rayo”, que presentan un tramo de 33 días consecutivos sin datos, se ha aplicado una estrategia de imputación basada en la similitud temporal. Concretamente, para cada valor ausente, se han considerado todos los días de la semana equivalentes; por ejemplo, si se está imputando un miércoles, se escogen todos los miércoles en el mes anterior y el mes posterior. A partir de esos días, se calcula la media de los valores registrados y se utiliza como estimación del dato faltante. Esta metodología permite conservar la estacionalidad semanal y mejorar la coherencia temporal en la imputación.
- Finalmente, la estación “Pinar del Rey” presenta un periodo con 104 días consecutivos de datos ausentes, correspondiente a los primeros meses del registro. Al no disponer de suficiente información previa para una imputación fiable, se ha decidido eliminar esta variable del conjunto de datos empleado en la fase de modelado.

En el caso de la variable “prec”, que representa la precipitación diaria, se ha optado por imputar los valores faltantes con cero. Esta decisión se fundamenta en la naturaleza de la variable: dado que en la mayoría de los días del periodo analizado no

se registran precipitaciones en Madrid, el valor más frecuente y representativo es precisamente 0. Por tanto, imputar con la media de otras observaciones podría distorsionar la interpretación, mientras que asignar un valor de 0 resulta más coherente con el comportamiento general de la variable.

8.2. Generación de dummies

Dada la presencia de distribuciones desbalanceadas en algunas variables numéricas como “prec”, así como la existencia de variables categóricas como “Dia_semana” y “tipo_dia”, se procede a la transformación de estas variables mediante la creación de variables *dummies*, lo cual facilita su incorporación en modelos predictivos.

En concreto, la variable “prec” se ha recodificado en una nueva variable categórica con tres niveles según el volumen de precipitación diaria:

- prec_baja para valores inferiores a 10 u.m.
- prec_media para el rango [10, 20) u.m.
- prec_alta para valores iguales o superiores a 20 u.m.

Dado que la categoría “prec_baja” representa la mayoría de los casos, se utiliza como categoría de referencia y no se incluye entre las variables *dummies* generadas.

En cuanto a las variables “Dia_semana” y “tipo_dia”, ambas se transforman en variables *dummies*. Se establece como categoría de referencia el día “lunes” para “Dia_semana”, y “festivos” para “tipo_dia”.

8.3. Creación de variables rezagadas (lags)

Dado que los datos utilizados en este trabajo presentan una estructura temporal, tras redondear los valores imputados anteriormente, resulta relevante incorporar variables de rezago (*lags*) con el fin de capturar posibles dependencias temporales en la variable objetivo. Este tipo de transformación permite que los modelos aprendan patrones de comportamiento secuenciales presentes en la serie temporal.

En este caso, se han generado tres variables rezagadas que representan el número de viajeros registrados en la estación objetivo en los días:

- $t-1$: corresponde a un día anterior.
- $t-2$: corresponde a dos días anteriores.
- $t-7$: corresponde a una semana anterior.

De esta manera, se consigue cubrir tanto los efectos inmediatos como los patrones semanales.

8.4. Estandarización de variables

Para evitar problemas de escalado entre las variables numéricas, se procede a su estandarización, transformando sus valores a una distribución con media cero y desviación típica igual a uno. Este proceso es fundamental en modelos de aprendizaje automático sensibles a la escala de las variables, como las redes neuronales o los modelos basados en distancias (SVR).

Paralelamente, se ha normalizado el formato de los nombres de las variables con el fin de facilitar su manipulación posterior. Esto incluye la conversión de caracteres en mayúsculas a minúsculas, la sustitución de espacios por guiones bajos (_), y la eliminación de tildes y caracteres especiales, etc. Con ello se garantiza una mayor homogeneidad y se previenen errores durante el desarrollo del modelado y análisis.

8.5. Análisis exploratorio de datos (EDA) post-preparación

Una vez finalizados los procesos de limpieza, depuración y preparación del conjunto de datos, se procede a realizar una nueva fase de análisis exploratorio de datos con retroalimentación, utilizando el *dataset* ya depurado. Esta etapa se enfoca principalmente en la detección de la presencia de correlación con la variable objetivo y la multicolinealidad entre las variables explicativas, así como la distribución en las variables *dummies*.

8.5.1. Evolución temporal de la variable objetivo

La variable objetivo “Chamartín” presenta un comportamiento típico de serie temporal (figura 8.1), con una periodicidad marcada durante los meses de verano, en los cuales el número de viajeros que acceden a la estación disminuye, probablemente debido al periodo vacacional.

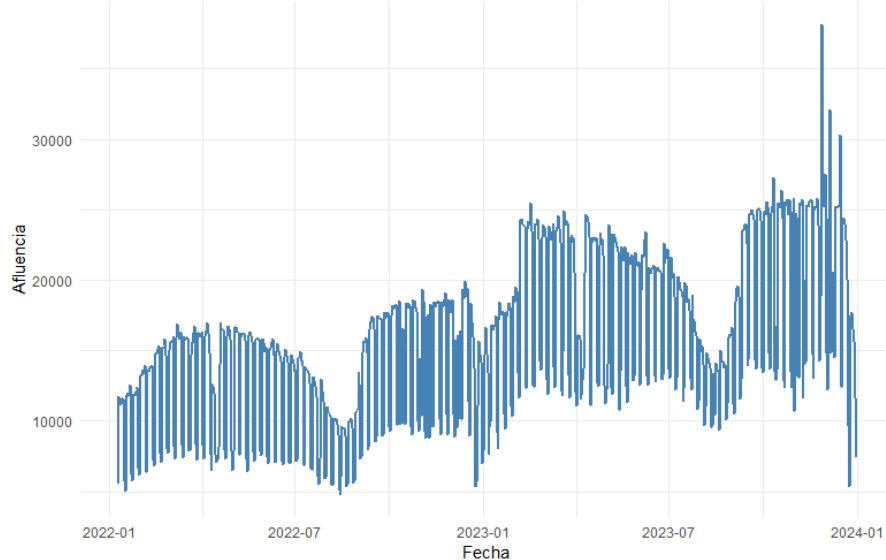
Aunque, de forma general, durante el período de análisis comprendido entre las fechas diarias de los años 2022 y 2023, se aprecia una tendencia creciente en el número de usuarios que acceden a esta estación.

8.5.2. Análisis de la correlación (gráfica de dispersión)

A partir de las variables predictoras seleccionadas, se ha realizado un análisis de la correlación con la variable objetivo mediante gráficas de dispersión individuales (figura 8.2).

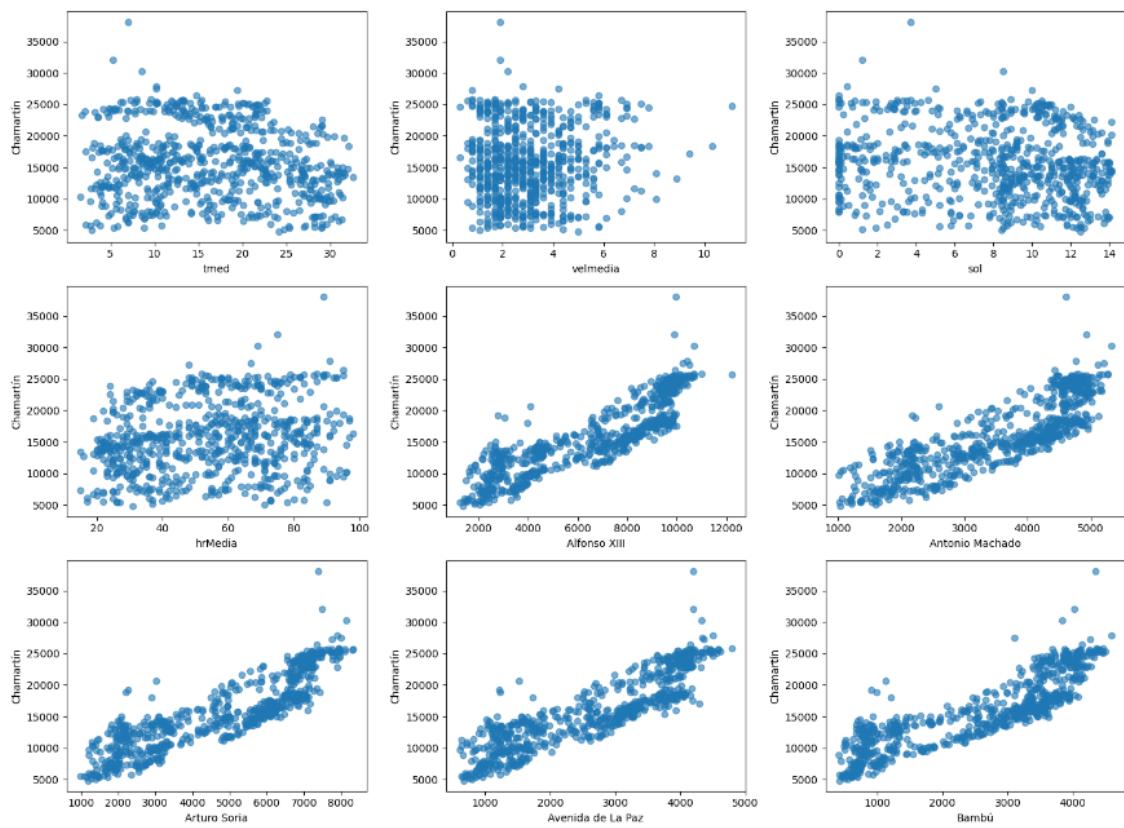
Seleccionando algunas de las variables explicativas, se aprecia una alta correlación lineal positiva en muchas variables numéricas correspondientes al número de entradas en estaciones próximas. La tendencia creciente casi lineal, con nubes de puntos

Figura 8.1: Evolución de las entradas de viajeros en la estación de Chamartín



Fuente: Elaboración propia

Figura 8.2: Gráfica de dispersión entre variables predictoras y Chamartín



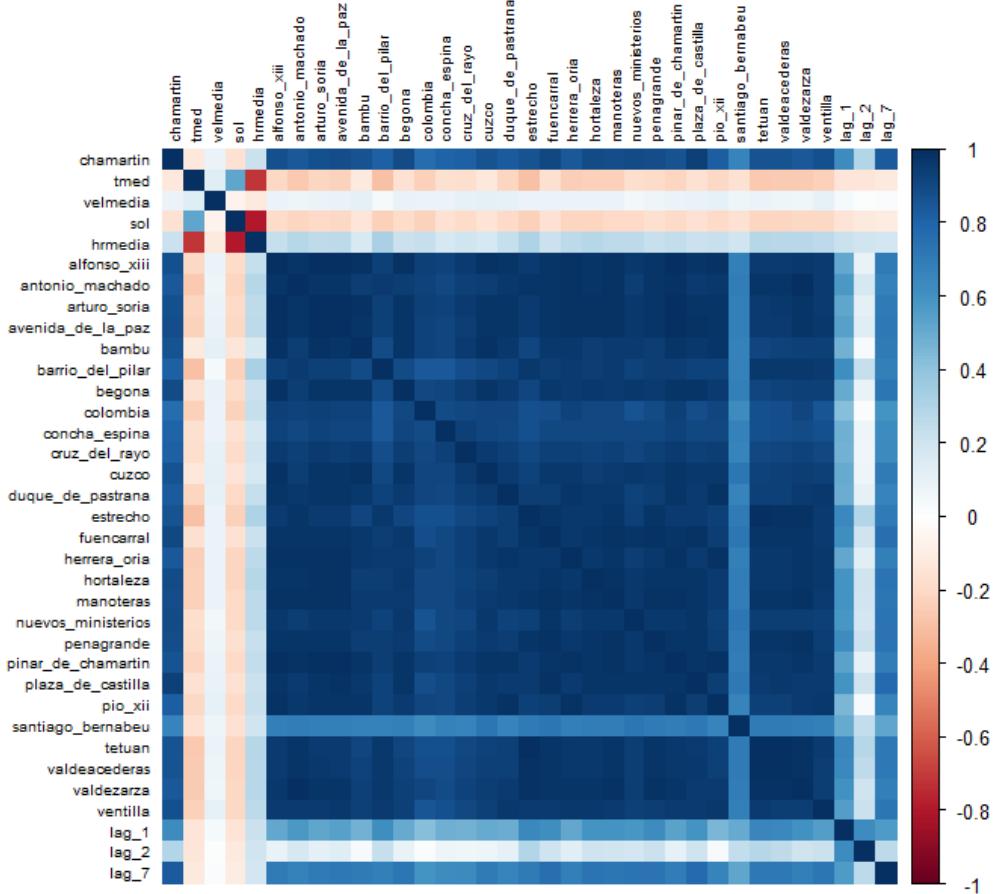
Fuente: Elaboración propia

concentradas y ajustadas a la recta, sugiere una fuerte capacidad predictiva sobre la variable dependiente. No obstante, esta fuerte relación también puede indicar la presencia de multicolinealidad entre algunas de las variables explicativas (véanse las figuras A.2, A.3, A.4 y A.5 en el apéndice de esta memoria).

8.5.3. Análisis de la multicolinealidad (mapa de calor)

El mapa de calor de correlaciones permite detectar visualmente posibles problemas de multicolinealidad entre variables independientes de forma rápida y efectiva.

Figura 8.3: Detección de multicolinealidad mediante mapa de calor



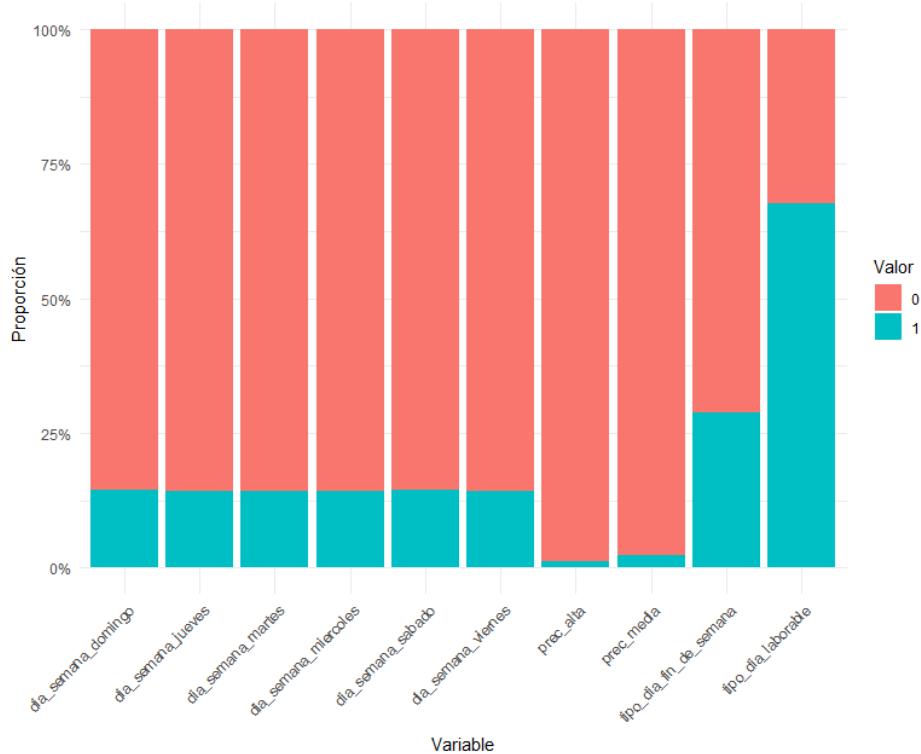
Fuente: Elaboración propia

En el mapa de calor generado (figura 8.3), se observa una zona central de color azul oscuro, que evidencia una alta correlación positiva entre las variables numéricas relacionadas con estaciones cercanas a Chamartín. Esto confirma la existencia de multicolinealidad, lo cual puede afectar negativamente a modelos lineales al reducir la estabilidad de los coeficientes estimados.

8.5.4. Distribución de variables dummies

La generación de variables *dummies* fue implementada para mejorar la representación de variables categóricas y algunas variables numéricas desbalanceadas, buscando así aportar una mayor capacidad explicativa al modelo final. Sin embargo, para que estas variables *dummies* sean útiles, se requiere una distribución equilibrada de las categorías, evitando un fuerte desbalance que podría sesgar el modelo.

Figura 8.4: Distribución de variables dummies (0/1)



Fuente: Elaboración propia

Tras analizar las distribuciones que se presentan en la figura 8.4, se detecta un desbalance considerable en las variables *dummies* generadas a partir de “Día_semana” y “prec”. En el primer caso, ningún día de la semana supera el 20 % del total de observaciones, y en el segundo, los días con precipitaciones medias o altas representan menos del 10 %. Esto limita la capacidad explicativa de estas variables dentro del modelo predictivo.

Capítulo 9

Selección de variables

El resultado obtenido del EDA post-preparación permite identificar la presencia de problemas relacionados con la multicolinealidad entre las variables independientes, así como un desbalance en algunas variables *dummies*. Para abordar estas limitaciones y trabajar con un conjunto óptimo de variables en el proceso de modelado, se procede a realizar un proceso de selección de variables²⁰.

9.1. Métodos de selección

Para abordar el proceso de selección de variables, se han empleado una serie de métodos atendiendo a diferentes prioridades y enfoques.

9.1.1. Métodos basados en criterios de información

Se ha utilizado el método de selección paso a paso (*stepwise*), que consiste en añadir o eliminar variables del modelo de forma iterativa. En cada paso, se reevalúa el conjunto de variables en función de un criterio de penalización que busca equilibrar el ajuste del modelo y su complejidad²⁰. En este trabajo se ha empleado dos criterios principales:

- **AIC (Akaike Information Criterion)**: favorece modelos con mejor capacidad predictiva, permitiendo estructuras algo más complejas al penalizar suavemente el número de variables.
- **BIC (Bayesian Information Criterion)**: incorpora una penalización más estricta, favoreciendo modelos más simples, especialmente en muestras grandes.

Ambos criterios permiten encontrar un modelo parsimonioso sin perder capacidad explicativa ni de predicción, y han sido aplicados tanto en su forma estándar como en versiones repetidas para mejorar la estabilidad del modelo seleccionado.

Por otra parte, dentro de esta misma categoría se incluye también el método *LEAPS* (Best Subset Selection), el cual realiza una búsqueda exhaustiva de todos los sub-

conjuntos posibles de variables explicativas. Esta técnica evalúa cada combinación de variables en función de criterios como el AIC, BIC o el R^2 ajustado, con el objetivo de identificar el modelo óptimo dentro del conjunto de posibles combinaciones¹⁷. En el presente trabajo, se ha empleado como generador de modelos candidatos, cuya calidad ha sido posteriormente evaluada mediante validación cruzada, priorizando un equilibrio entre rendimiento predictivo (RMSE, R^2) y menor complejidad del modelo.

9.1.2. Métodos de selección por penalización (regularización)

En cuanto a los métodos de selección mediante penalización en la función de pérdida, se han utilizado los modelos *Lasso* (Least Absolute Shrinkage and Selection Operator) y *Ridge Regression*, explicados previamente en el apartado 5.1 sobre modelos de regresión lineal.

En resumen, Lasso realiza selección automática de variables al reducir algunos coeficientes a cero mediante penalización L1, mientras que Ridge aplica una penalización L2 que reduce la magnitud de los coeficientes sin eliminarlos, siendo útil para manejar la multicolinealidad.

9.1.3. Métodos basados en importancia de variables (embedded)

Esta categoría agrupa aquellos métodos que seleccionan las variables durante el proceso de entrenamiento del modelo, en función de su contribución relativa al rendimiento predictivo. En este trabajo, se ha empleado el algoritmo de *Random Forest* para estimar la importancia de cada variable, aprovechando su capacidad para generar múltiples árboles de decisión y evaluar la influencia de cada predictor.

9.1.4. Métodos de selección tipo wrapper

Este tipo de métodos evalúa diferentes subconjuntos de variables de forma iterativa, entrenando un modelo predictor en cada caso y seleccionando aquellas combinaciones que optimizan una métrica de rendimiento (por ejemplo, RMSE o R^2 en regresión).

En este trabajo se han utilizado los siguientes métodos de tipo *wrapper*:

- **RFE (Recursive Feature Elimination)**: elimina iterativamente las variables menos relevantes según los coeficientes del modelo, evaluando en cada paso el rendimiento obtenido. Se detiene al alcanzar el número óptimo de variables que maximiza el rendimiento¹³.
- **SBF (Selection By Filtering)**: combinación de filtro previo (basado en importancia de variables) seguido de validación cruzada con el modelo para seleccionar los subconjuntos más prometedores¹³.
- **Boruta**: aunque puede clasificarse como *embedded*, en este trabajo se considera un método *wrapper* por su enfoque iterativo basado en Random Forest.

Compara la importancia de las variables reales con la de variables aleatorizadas (*shadow features*), reteniendo solo aquellas claramente relevantes²¹.

9.1.5. Métodos basados en independencia condicional

Este grupo de métodos parte de principios estadísticos más rigurosos para la selección de variables, centrándose en la detección de relaciones condicionales entre las variables predictoras y la variable objetivo. El objetivo es identificar conjuntos de variables que aporten información única y no redundante para la predicción, descartando aquellas que resultan irrelevantes una vez controlado por otras.

En este trabajo se han empleado estos dos algoritmos²²:

- **MMPC (Max-Min Parents and Children)**: este algoritmo busca identificar el conjunto mínimo de variables que son padres o hijos directos de la variable objetivo en una estructura de red bayesiana. Funciona a través de tests de independencia condicional (como tests de correlación parcial o chi-cuadrado).
- **SES (Statistically Equivalent Signatures)**: extiende MMPC al permitir encontrar múltiples subconjuntos de variables equivalentes, es decir, combinaciones distintas que poseen el mismo poder predictivo estadístico.

9.2. Conjunto de variables seleccionadas

Una vez aplicados los distintos métodos de selección sobre el conjunto de 44 variables previamente depurado, se obtienen las combinaciones óptimas específicas de cada técnica. En el caso de los métodos *Stepwise* repetidos, se ha seleccionado la combinación que, contando con al menos una variable, presenta el menor número de predictores y la mayor frecuencia de aparición.

Figura 9.1: Variables escogidos por cada método de selección de variables

	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	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	
10	barrio_del_pilar							X	X		X	X	X
11	begona	X			X			X			X	X	
12	colombia	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	
16	duque_de_pastrana							X	X		X	X	X
17	estrecho	X	X					X	X		X	X	X
18	fuencarral	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
23	penagrande	X	X					X	X	X	X	X	X
24	pinar_de_chamartin	X						X	X	X	X	X	
25	plaza_de_castilla	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	tetuán							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
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_miércoles	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	X	X	
40	tipo_dia_laborable	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
44	lag_7	X	X		X			X	X	X	X	X	X

Fuente: Elaboración propia

La figura 9.1 muestra una gran variabilidad en las combinaciones seleccionadas según el método empleado. El método más parsimonioso ha resultado ser SES, que selecciona únicamente cinco variables, mientras que Ridge, debido a su naturaleza de regularización sin eliminación explícita de predictores, incluye la totalidad de las variables disponibles.

9.3. Resultado obtenido

Una vez identificadas las combinaciones óptimas de variables por método, se ha procedido a evaluar su rendimiento mediante validación cruzada repetida (figura 9.2), obteniendo así las métricas de error asociadas a cada modelo.

Figura 9.2: Resultados de los método 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	0.0472	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

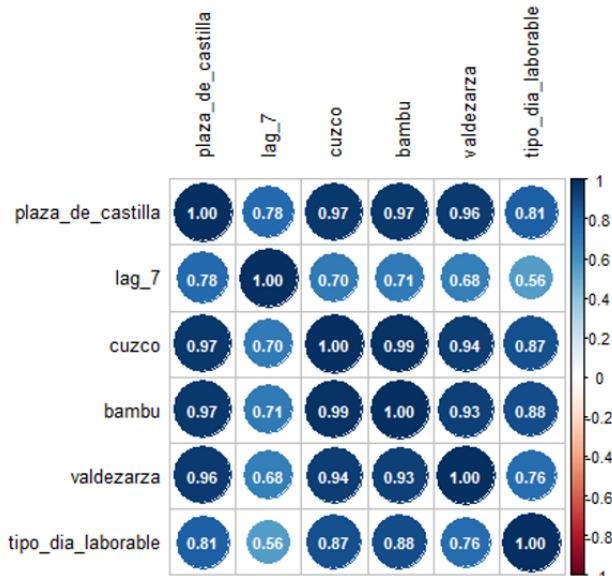
Fuente: Elaboración propia

Se observa que, aunque SES presenta la estructura más sencilla, también es el método que mayor error comete, con un RMSE = 0,425. Por tanto, bajo el criterio de seleccionar el modelo que ofrezca el mejor equilibrio entre rendimiento y simplicidad, el modelo más recomendable sería el obtenido mediante Stepwise repetido con criterio BIC (STEP_rep_BIC). Este modelo alcanza un RMSE = 0,227, muy cercano al mejor modelo en rendimiento (Stepwise AIC), pero utilizando únicamente seis variables.

9.4. Revisión de la multicolinealidad

Tras seleccionar el modelo óptimo, se procede a revisar de nuevo la multicolinealidad:

Figura 9.3: Análisis de multicolinealidad entre variables predictoras



Fuente: Elaboración propia

Al observar el mapa de calor (figura 9.3), se aprecia que aún persisten niveles elevados de correlación entre algunas variables, especialmente en estaciones como *Cuzco*, *Bambú* y *Valdezarza*, lo que sugiere una posible redundancia de información.

Para confirmar esta situación, se recurre al *Factor de Inflación de la Varianza (VIF)*, una métrica que cuantifica el aumento de la varianza estimada de los coeficientes debido a la colinealidad entre predictores, permitiendo así detectar redundancias en el modelo.

Cuadro 9.1: Valores del 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

Fuente: *Elaboración propia*

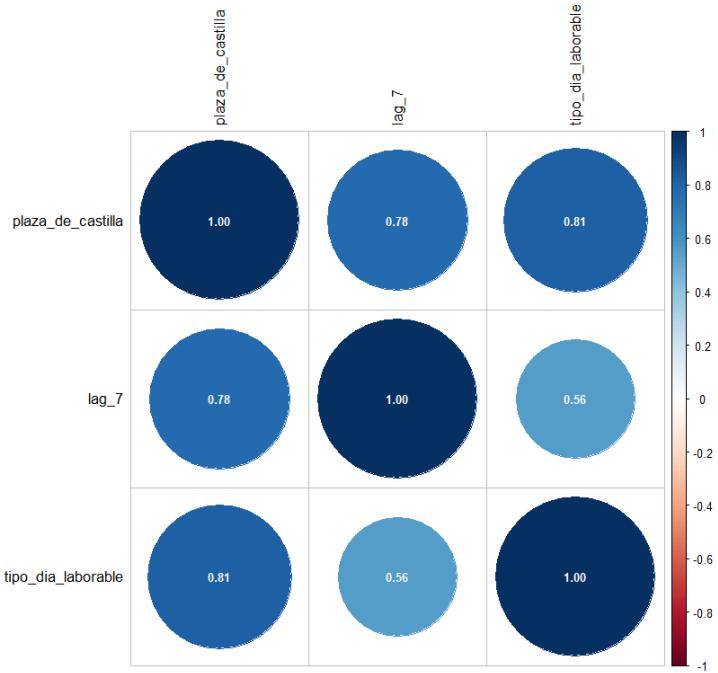
El rango aceptable de VIF se encuentra entre 1 y 10. Valores superiores a 10 indican una alta multicolinealidad, que puede afectar a la interpretación y estabilidad del modelo. En este caso, se detectan valores muy por encima del umbral, destacando “Bambú”, con un VIF de 52.13 (cuadro 9.1), reflejando una colinealidad inaceptable³⁰.

9.5. Reducción de variables por multicolinealidad

La presencia de variables predictoras altamente correlacionadas entre sí provoca severos problemas a la hora de modelar con ello, debido a la presencia de variables redundantes que no aportan valor extra al modelo.

Por ello, se procede a eliminar de forma gradual las variables con mayor valor VIF. Como resultado se ha descartado “bambu”, “cuzco” y “valdezarza”. La estación “plaza_de_castilla”, aunque presenta un valor VIF superior al de “valdezarza” a la hora de eliminarlo, se ha repercutido considerablemente en el resultado de R^2 , por lo que se demostró ser una variable muy importante para predecir la estación de Chamartín.

Figura 9.4: Análisis de multicolinealidad tras el ajuste



Fuente: Elaboración propia

Revisando de nuevo el mapa de calor (figura 9.4), se observa que ahora presenta una correlación alta entre si, pero sin alcanzar a valores muy extremos.

Cuadro 9.2: Valores del Factor de Inflación de la Varianza (VIF) final

Variable	VIF
plaza_de_castilla	5.295454
lag_7	2.627384
tipo_dia_laborable	3.036607

Fuente: Elaboración propia

Con la estadística de valor de VIF (cuadro 9.2), tras el ajuste de las variables predictoras, ahora se encuentra en un rango inferior a 10, por lo que se considera como un modelo aceptable para el modelado. El nuevo modelo para el entrenamiento estará formado por “plaza_de_castilla”, “lag_7” y “tipo_dia_laborable”.

Tras realizar un reentrenamiento del modelo de regresión lineal utilizando los parámetros por defecto y dos conjuntos de variables (el original y una versión reducida sin multicolinealidad), se obtienen los siguientes resultados (cuadro 9.3):

Cuadro 9.3: Comparación del rendimiento entre el modelo original y el ajustado

Modelo	Nº variables	RMSE	R ²
STEP_rep_BIC	6	0.222	0.950
STEP_rep_BIC (ajustado)	3	0.280	0.922

Fuente: Elaboración propia

Los resultados de RMSE y R^2 no se vieron afectados de forma significativa tras el ajuste del modelo. Esto indica que el nuevo modelo, con solo tres variables explicativas, mantiene prácticamente la misma capacidad predictiva que el modelo original con seis variables, lo que lo convierte en una opción más parsimoniosa y eficiente para su implementación.

Capítulo 10

Construcción del modelo y evaluación de resultados

Tras obtener la combinación óptima de variables, se inicia la fase de *Modelado*. Esta se desarrolla en tres etapas. Primero, se entrena cada modelo explorando un rango predefinido de hiperparámetros para identificar la configuración con mejor rendimiento y control de complejidad.

A continuación, se reentrena el modelo con la mejor combinación encontrada, aplicando validación cruzada repetida de 10 pliegues (*folds*) y 10 repeticiones, generando así 100 combinaciones entrenamiento-validación que aseguran resultados estables y fiables.

Finalmente, se selecciona el modelo ganador de cada técnica para su posterior comparación.

10.1. Red neuronal

Comenzando con el modelo de red neuronal, en primer lugar se procede a establecer el rango de nodos que deben ser analizados. Para ello, teniendo en cuenta que el conjunto de datos dispone de 723 observaciones y tres variables predictoras finales para el modelado, una opción segura es establecer 20 observaciones por parámetro. En este caso, el límite empírico de parámetros debería ser:

$$\frac{N^{\circ} \text{ obs}}{N^{\circ} \text{ obs/parámetro}} = \text{límite máximo de parámetros recomendados}$$

Donde:

$$\frac{723}{20} = 36,15 \quad \text{parámetros máximos recomendados}$$

Por lo tanto, según esta fórmula empírica, el número máximo de parámetros reco-

mendados es de 36.15. Si se aplica esta restricción a la siguiente ecuación empírica:

$$\text{Nº parámetros} = h(k + 1) + h + 1$$

Donde:

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

Entonces:

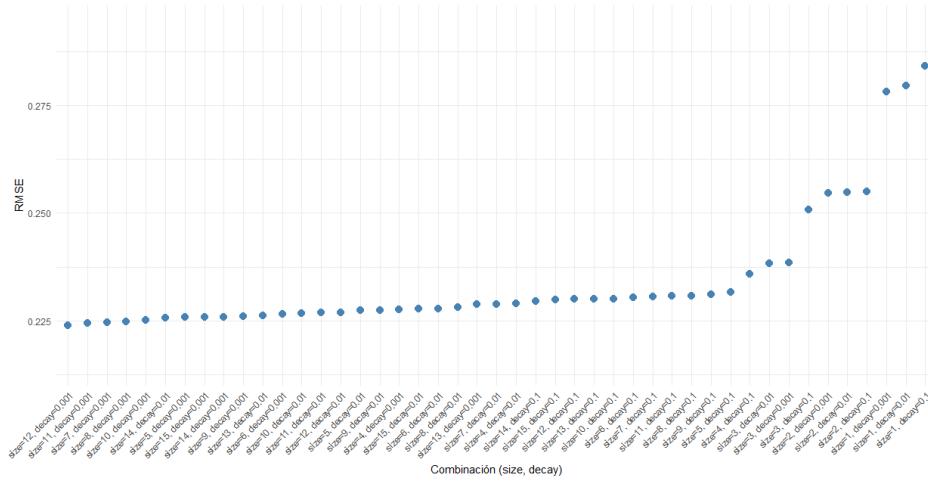
$$36,15 = h(3 + 1) + h + 1$$

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

$h = 7,03$ nodos

En consecuencia, el número de nodos ocultos recomendado no debería superar el valor de 7, de acuerdo con esta restricción empírica. Para observar cómo evoluciona el RMSE, se realiza un entrenamiento exploratorio (figura 10.1) con un rango de nodos desde 1 hasta 15 y valores de decay iguales a 0.1, 0.01 y 0.001.

Figura 10.1: Fine tuning automático de los hiperparámetros de la Red Neuronal



Fuente: Elaboración propia

Se puede observar que, a partir del nodo 3, los errores ya no tienden a mejorar de forma significativa respecto a los nodos menores. A partir del nodo 5, el RMSE tiende a estabilizarse.

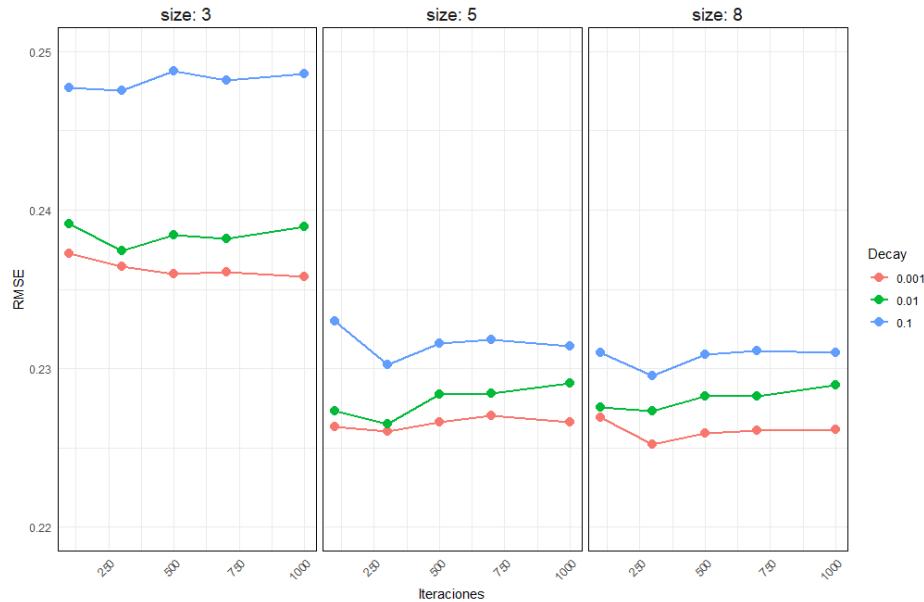
Por lo tanto, teniendo en cuenta la evolución del RMSE con el rango de nodos de 1 a 15 y el resultado obtenido mediante la fórmula empírica, se considera razonable analizar el rendimiento de la red neuronal estableciendo puntos de corte en los nodos 3, 5 y 8 para el entrenamiento.

Se fijan los siguientes hiperparámetros para cada número de nodos, incluyendo el

parámetro de regularización (*decay*) y el número máximo de iteraciones (*maxit*), utilizando los siguientes valores:

- *decay_values*: 0.1, 0.01, 0.001
- *listaiter*: 100, 300, 500, 700, 1000

Figura 10.2: Fine tuning de los hiperparámetros de la Red Neuronal



Fuente: Elaboración propia

Como resultado, se obtiene la siguiente gráfica (figura 10.2). En ella se puede observar que, a partir de 5 nodos ocultos, la red prácticamente no mejora el error. Además, dentro del nodo 5, tampoco se aprecia una mejora significativa del *decay* a medida que aumenta el número de iteraciones. Por tanto, según el criterio rendimiento-complejidad, se opta por la combinación más simple: *decay* = 0.1 y *maxit* = 300, donde se alcanza el punto mínimo del error.

Asimismo, al comparar esta configuración con los resultados obtenidos con los otros dos nodos y sus respectivas combinaciones de hiperparámetros, se concluye que la red con 5 nodos ha demostrado ofrecer el mejor rendimiento en términos de equilibrio entre precisión y complejidad del modelo.

El resumen del mejor modelo de red neuronal queda definido de la siguiente manera (cuadro 10.1):

Cuadro 10.1: Resultado del mejor modelo de red neuronal (nodo 5)

size	decay	bag	RMSE	R²	MAE	RMSESD	R²SD	MAESD
5	0.1	FALSE	0.2308	0.9465	0.1602	0.0488	0.0226	0.0176

Fuente: Elaboración propia

10.2. Regresión lineal

En el caso del modelo de regresión lineal, no fue necesario realizar ninguna configuración de hiperparámetros. Únicamente se seleccionaron las tres variables explicativas finales y se aplicó una validación cruzada repetida con el fin de obtener el resultado del mejor modelo de regresión lineal.

El resumen del mejor modelo de regresión lineal es el siguiente (cuadro 10.2):

Cuadro 10.2: Resultados del mejor modelo de regresión lineal

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

Fuente: Elaboración propia.

10.3. Ridge

Para el modelo de Ridge, se trabaja con el parámetro de regularización (λ). Para ello, se determina el valor de λ que minimiza el error de validación.

$$\lambda_{\text{óptimo}} = 0,09388932$$

En este caso, el valor de λ es aproximadamente 0,0939. Incorporando este valor al proceso de entrenamiento mediante validación cruzada repetida, se obtiene el siguiente mejor modelo de Ridge (cuadro 10.3):

Cuadro 10.3: Resultados del mejor modelo Ridge

RMSE	R ²	MAE
0.3015469	0.9123395	0.2279584

Fuente: Elaboración propia

10.4. Lasso

El criterio de ajuste de hiperparámetros (fine tuning) para el modelo de Lasso es el mismo que en el caso de Ridge. Se calcula el valor de λ que minimiza el error de validación para su posterior incorporación al modelo final.

En este caso, el valor resultante de λ es aproximadamente 0,001843085. Estableciendo dicho valor en el modelo y aplicando validación cruzada repetida, se obtiene el siguiente mejor modelo de Lasso (cuadro 10.4):

Cuadro 10.4: Resultados del mejor modelo Lasso

RMSE	R ²	MAE
0.2803189	0.9221134	0.2123291

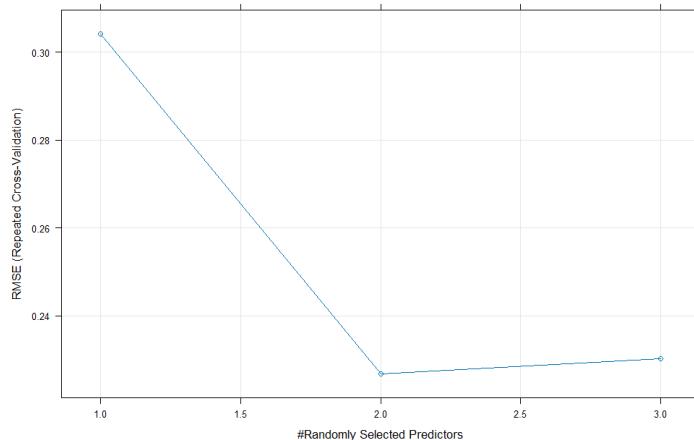
Fuente: Elaboración propia

10.5. Random forest

Para el modelo de Random Forest, se procede a trabajar con los hiperparámetros como el número de variables en cada nodo (*mtry*), el número de árboles generados (*ntree*) y el número mínimo de observaciones en cada nodo terminal.

Comenzando con el parámetro *mtry*, dado que se ha seleccionado un total de tres variables predictoras, se limita su rango de búsqueda entre 1 y 3.

Figura 10.3: Fine tuning del parámetro *mtry* en el modelo Random Forest

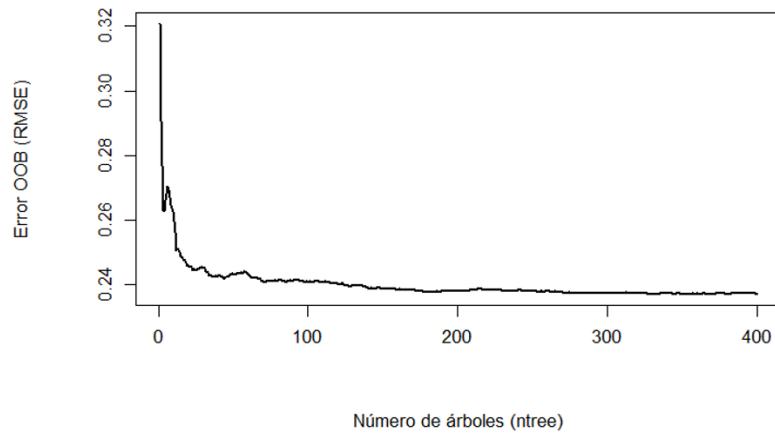


Fuente: Elaboración propia

A partir del análisis gráfico realizado (figura 10.3), se observa que con *mtry* = 2 se alcanza el valor mínimo de RMSE. Por lo tanto, éste será el punto óptimo del hiperparámetro.

En relación con el número de árboles generados (*ntree*), se procede a determinar el punto de corte a partir del cual el incremento en el número de árboles no aporta mejoras significativas en el rendimiento del modelo. Para ello, se emplea el cálculo del error *Out-of-Bag* (OOB), el cual se obtiene evaluando las observaciones que no han sido utilizadas en la construcción de cada árbol específico durante el proceso de *bagging* (muestreo con reemplazo).

Figura 10.4: Evolución del error OOB en función del *ntree* en Random Forest

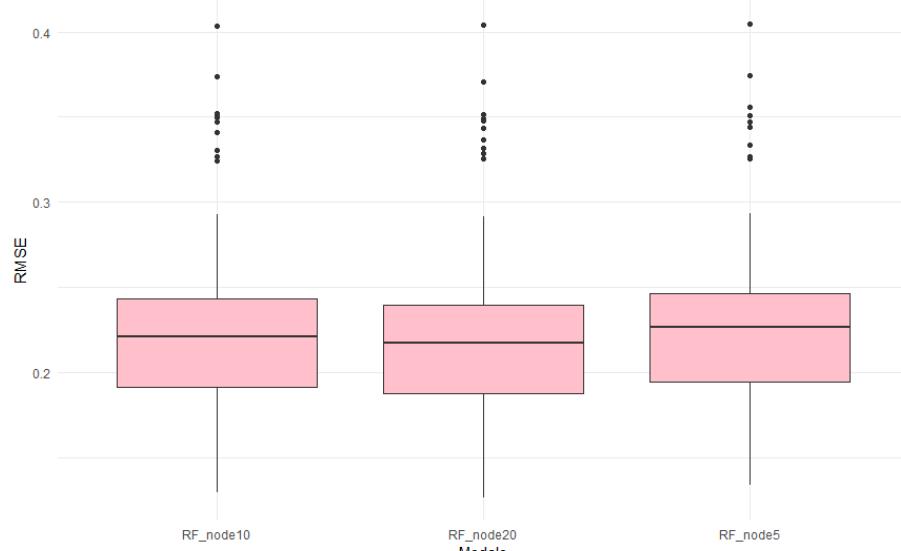


Fuente: Elaboración propia

Observando la gráfica del error OOB (figura 10.4), se aprecia que a partir de $ntree = 200$, el valor del error se estabiliza. Por consiguiente, se considera 200 como un valor adecuado para el hiperparámetro *ntree*.

Respecto al número mínimo de observaciones en cada nodo terminal (*nodesize*), se prueban los valores 5, 10 y 20.

Figura 10.5: Comparación del RMSE para distintos tamaños de nodo en Random Forest



Fuente: Elaboración propia

Mediante el entrenamiento del modelo con validación cruzada repetida, se obtiene una gráfica de caja (figura 10.5) donde muestra que, para *nodesize* = 20, el RMSE

alcanza su valor mínimo entre los tres valores evaluados.

Siguiendo el criterio de equilibrio entre rendimiento y complejidad del modelo, se selecciona *nodesize* = 20 para el modelo final. Esta elección contribuye a simplificar la profundidad de los árboles generados, lo que reduce la complejidad del modelo y disminuye el riesgo de sobreajuste.

El resumen del mejor modelo final de Random Forest es el siguiente:

Cuadro 10.5: Resultados del mejor modelo Random Forest

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

Fuente: Elaboración propia

10.6. XGBoost

En el caso del modelo XGBoost, se ha realizado un ajuste de hiperparámetros (*fine tuning*) sobre los siguientes elementos: el número de árboles (*nrounds*), la tasa de aprendizaje (*eta*), la profundidad máxima de los árboles (*max_depth*) y el número mínimo de observaciones en cada nodo hijo (*min_child_weight*).

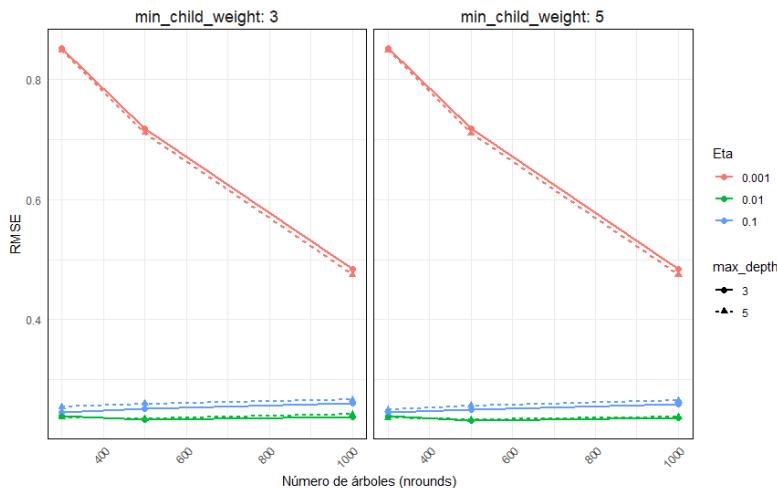
Los rangos establecidos para cada uno de estos hiperparámetros han sido los siguientes:

A partir del análisis de la gráfica del RMSE (figura 10.6) en función del número de árboles (*nrounds*), se observa, en primer lugar, que el valor del parámetro *min_child_weight* no influye de manera significativa en la mejora del error. Por tanto, se opta por *min_child_weight* = 5, ya que permite generar árboles más simples y menos propensos al sobreajuste.

- *nrounds*: 300, 500, 1000
- *eta*: 0.1, 0.01, 0.001
- *max_depth*: 3, 5
- *min_child_weight*: 3, 5

Del mismo modo, la profundidad máxima de los árboles (*max_depth*) presenta resultados similares entre los valores evaluados, por lo que se selecciona *max_depth* = 3, favoreciendo así modelos menos complejos.

Figura 10.6: Fine tuning de los hiperparámetros del modelo XGBoost



Fuente: Elaboración propia

En cuanto a la tasa de aprendizaje (*eta*), se observa un mejor rendimiento para el valor *eta* = 0,01, estableciendo el punto de corte óptimo en *nrounds* = 500.

Con esta configuración de hiperparámetros seleccionados, se procede a entrenar mediante validación cruzada repetida (cuadro 10.6). El mejor modelo de XGBoost es:

Cuadro 10.6: Resultados del mejor modelo XGBoost

nrounds	eta	max_depth	gamma	colsample_bytree	min_child_weight	
500	0.01	3	0	1	5	
subsample	RMSE	R²	MAE	RMSESD	R²SD	MAESD
1	0.2314668	0.946073	0.1589347	0.05127528	0.02360878	0.01975624

Fuente: Elaboración propia.

10.7. LightGBM

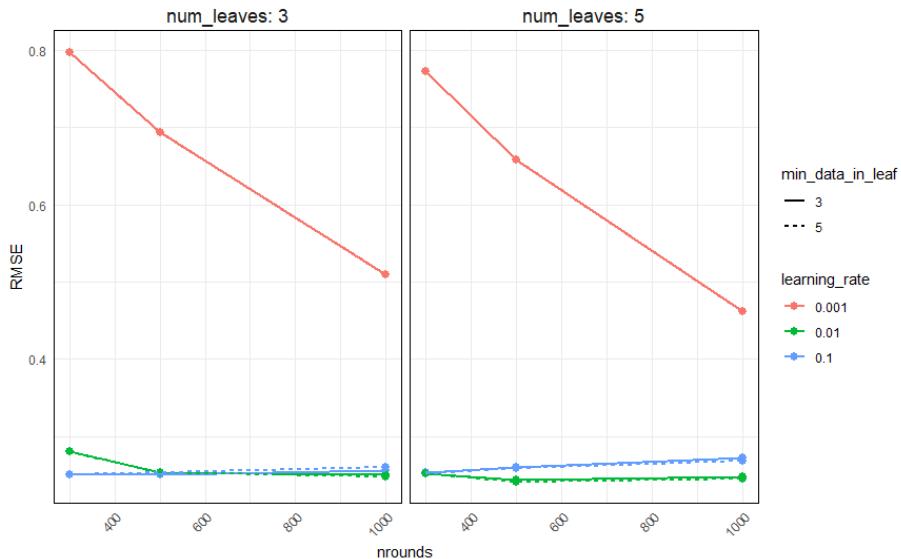
Otro de los modelos basados en árboles utilizados en este trabajo es LightGBM. Los hiperparámetros considerados para su ajuste son similares a los empleados en el modelo XGBoost, incluyendo el número de árboles (*nrounds*), la tasa de aprendizaje (*learning_rate*), el número máximo de hojas por árbol (*num_leaves*) y el número mínimo de observaciones en una hoja terminal (*min_data_in_leaf*).

Los rangos establecidos para cada hiperparámetro han sido los siguientes:

- *nrounds*: 300, 500, 1000

- *learning_rate*: 0.1, 0.01, 0.001
- *num_leaves*: 3, 5
- *min_data_in_leaf*: 3, 5

Figura 10.7: Fine tuning de los hiperparámetros del modelo LightGBM



Fuente: Elaboración propia

A partir del análisis de la gráfica del RMSE (figura 10.7), se observa que el número de hojas por árbol (*num_leaves*) apenas influye en el rendimiento del modelo. Por este motivo, se opta por el valor más sencillo: *num_leaves* = 3.

En cuanto a la tasa de aprendizaje, se compara el rendimiento entre los valores 0.1 y 0.01, concluyéndose que *learning_rate* = 0.1 constituye la mejor opción, ya que permite alcanzar el mínimo del RMSE con una configuración más simple, situando el punto óptimo en *nrounds* = 500.

Respecto al número mínimo de observaciones por hoja terminal (*min_data_in_leaf*), no se aprecian diferencias significativas entre las opciones. Por tanto, se elige *min_data_in_leaf* = 5, ya que genera árboles más simples y menos propensos al sobreajuste.

La mejor combinación de hiperparámetros para el modelo LightGBM tras realizar el entrenamiento con validación cruzada repetida es la siguiente (cuadro 10.7):

Cuadro 10.7: Resultados del mejor modelo LightGBM

Modelo	RMSE	R ²	MAE
LGBM	0.250832	0.9371698	0.1678948

Fuente: Elaboración propia.

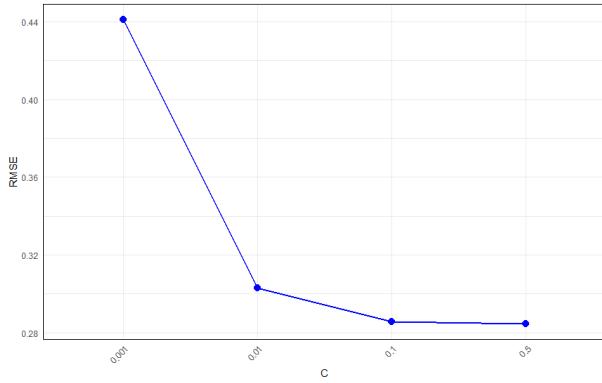
10.8. SVR

En cuanto a los modelos de Soporte Vectorial para Regresión (SVR), en este trabajo se han considerado tres enfoques distintos según el tipo de *kernel* empleado: SVR Lineal, SVR Polinomial y SVR Radial (RBF).

10.8.1. SVR Lineal

Comenzando con el modelo SVR Lineal, se ha realizado el ajuste del parámetro de regularización (C) evaluando los siguientes valores: 0.5, 0.1, 0.01 y 0.001 (figura 10.8).

Figura 10.8: Fine tuning del parámetro C en el modelo SVR Lineal



Fuente: Elaboración propia

Al observar la gráfica, se aprecia que el valor de C que ofrece el menor RMSE corresponde a 0.1. A partir de dicho punto, una reducción adicional de C no mejora el rendimiento del modelo, mientras que un incremento de este parámetro tiende a generar modelos más complejos y con mayor probabilidad de sobreajuste. Por tanto, se selecciona $C = 0.1$ como valor óptimo.

Aplicando esta configuración en el entrenamiento mediante validación cruzada repetida, se obtienen los siguientes resultados (cuadro 10.8):

Cuadro 10.8: Resultados del mejor modelo SVR Lineal

RMSE	R ²	MAE
0.2858566	0.9204151	0.207427

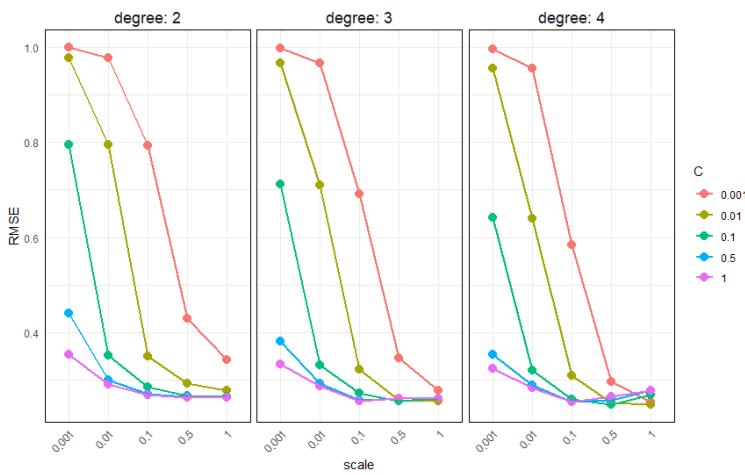
Fuente: Elaboración propia.

10.8.2. SVR polinomial

Para el modelo SVR Polinomial se ha llevado a cabo el ajuste de los hiperparámetros C , el grado del polinomio (*degree*) y el parámetro de escala del término de entrada (*scale*). Los valores evaluados fueron los siguientes:

- *degree*: 2, 3, 4
- *scale*: 1, 0.5, 0.1, 0.01, 0.001
- $C =$: 1, 0.5, 0.1, 0.01, 0.001

Figura 10.9: Fine tuning del parámetro C en el modelo SVR Lineal



Fuente: Elaboración propia

Del análisis gráfico (figura 10.9) se desprende que el grado del polinomio no influye de manera significativa en el rendimiento del modelo para este conjunto de datos, por lo que se selecciona el valor más simple: *degree* = 2.

En cuanto al parámetro C , se observa un comportamiento similar al del modelo SVR Lineal: valores mayores a 0.1 no mejoran de forma sustancial el RMSE y aumentan la complejidad del modelo, mientras que valores más bajos tampoco ofrecen mejores resultados. Por tanto, se escoge $C = 0.1$.

Del mismo modo, el valor óptimo de *scale* se establece en 0.1, punto en el cual el error se minimiza sin comprometer la simplicidad del modelo. Con esta configuración, los resultados del entrenamiento mediante validación cruzada repetida son (cuadro 10.9):

Cuadro 10.9: Resultados del mejor modelo SVR Polinomial

RMSE	R ²	MAE
0.2864518	0.9202174	0.2039351

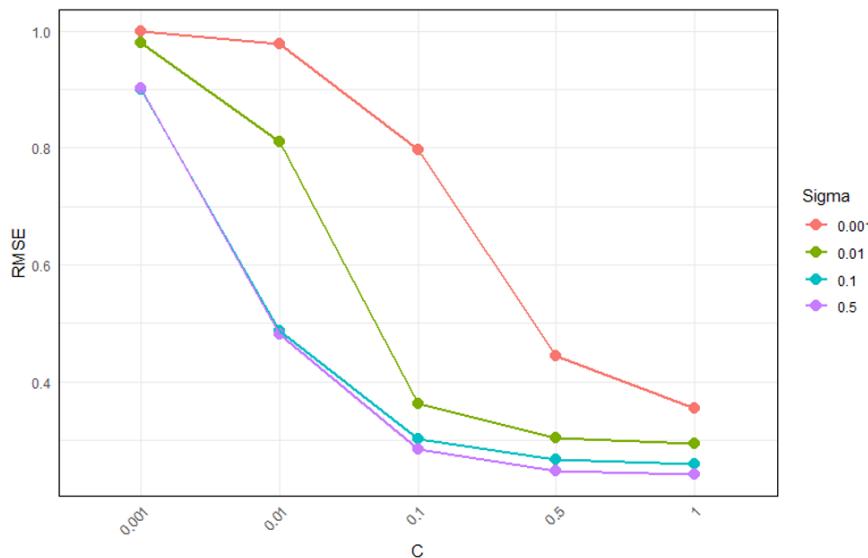
Fuente: Elaboración propia.

10.8.3. SVR radial (RBF)

Para el modelo SVR Radial (RBF), se ha realizado el ajuste de los hiperparámetros C y σ , este último correspondiente a la amplitud de la función de base radial. Los valores evaluados fueron los siguientes:

- $\sigma = \{0.001, 0.01, 0.1, 0.5\}$
- $C = \{1, 0.5, 0.1, 0.01, 0.001\}$

Figura 10.10: Fine tuning de los hiperparámetros en el modelo SVR Radial (RBF)



Fuente: Elaboración propia

Del análisis de la gráfica (figura 10.10) se concluye que el valor óptimo para el parámetro σ es 0,1, ya que proporciona un buen equilibrio entre rendimiento y complejidad del modelo. En cuanto al parámetro de regularización C , también se selecciona el valor 0,1 siguiendo el mismo criterio adoptado en los modelos anteriores.

Aplicando esta combinación en el proceso de entrenamiento con validación cruzada repetida, se obtienen los siguientes resultados (cuadro 10.10):

Cuadro 10.10: Resultados del mejor modelo SVR Radial

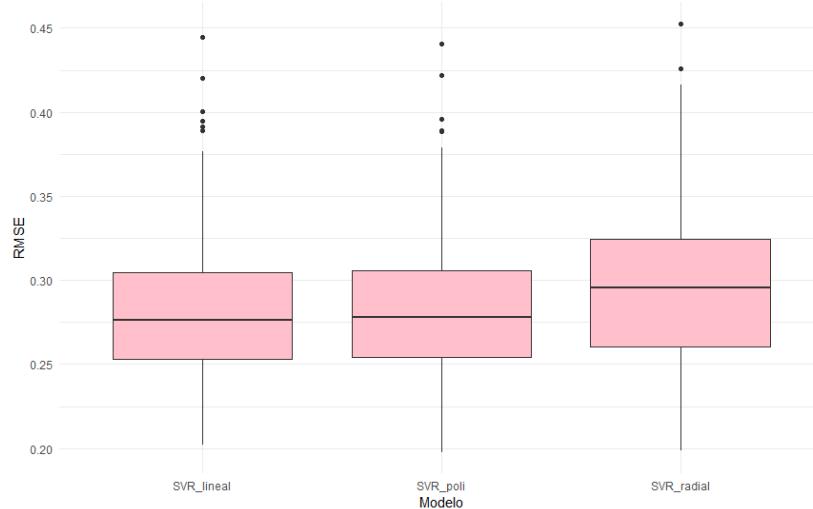
RMSE	R ²	MAE
0.3009869	0.9132075	0.2115539

Fuente: Elaboración propia

10.8.4. Mejor modelo SVR

Una vez analizados los tres modelos del SVR, se procede a evaluar su desempeño mediante una comparación directa (figura 10.11).

Figura 10.11: Comparación del error (RMSE) entre modelos SVR



Fuente: Elaboración propia

Analizando el RMSE de los tres modelos, se observa que el modelo SVR Lineal presenta el mejor rendimiento, al obtener el menor error entre las tres variantes evaluadas. Además, este resultado se complementa con el hecho de que dicho modelo cuenta con la estructura más sencilla, lo cual reduce su complejidad y el riesgo de sobreajuste.

Por lo tanto, el modelo óptimo de SVR seleccionado es el SVR Lineal, cuyos resultados se detallan a continuación (cuadro 10.11):

Cuadro 10.11: Resultados del mejor modelo SVR (Lineal) con $C = 0,1$

C	RMSE	R ²	MAE	RMSESD	R ² SD	MAESD
0.1	0.2858566	0.9204151	0.207427	0.04875873	0.02497062	0.02140923

Fuente: Elaboración propia.

Capítulo 11

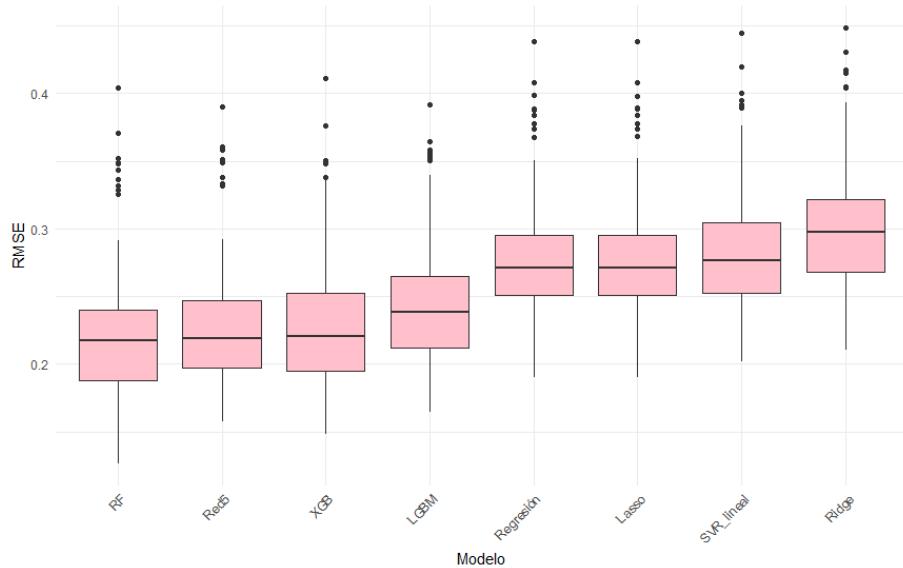
Selección del modelo óptimo

Tras entrenar todos los modelos predictivos con las variables seleccionadas, se procede a evaluar los resultados con el objetivo de identificar el modelo óptimo e interpretarlo adecuadamente. Adicionalmente, se realiza un análisis complementario en SAS para contrastar el comportamiento de algunos modelos bajo configuraciones similares de los hiperparámetros a las utilizadas en el desarrollo principal.

11.1. Evaluación de los modelos finales

Para la evaluación del modelo, se han considerado los resultados de RMSE y R^2 como métricas de comparación.

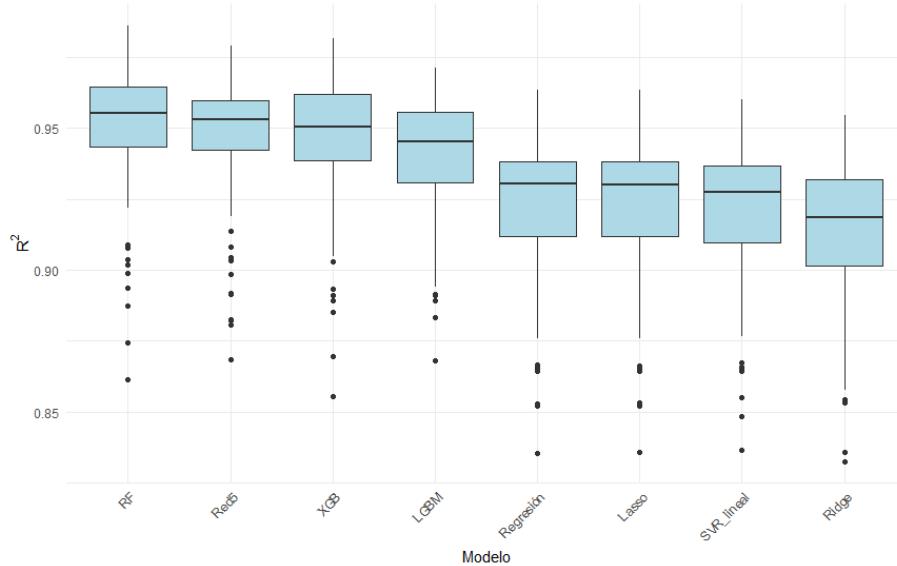
Figura 11.1: Comparación de modelos finales según RMSE



Fuente: Elaboración propia

Tras agrupar los ocho modelos finales y ordenarlos de menor a mayor RMSE (figura 11.1), se observa que el modelo de *Random Forest* presenta el mejor resultado en términos de precisión, al obtener el valor de error más bajo.

Figura 11.2: Comparación de modelos finales según R^2



Fuente: Elaboración propia

Comparando los resultados del R^2 , se confirma que el modelo de *Random Forest* alcanza el valor más elevado, lo que evidencia su mayor capacidad explicativa en comparación con el resto de modelos. Por lo que es elegido como el modelo ganador en base al análisis de los resultados obtenidos tras el tuneo exhaustivo de los hiperparámetros y el entrenamiento mediante validación cruzada repetida.

11.2. Evaluación complementaria con SAS

Con el objetivo de poner en práctica el entorno de trabajo de SAS, se ha llevado a cabo una comprobación adicional de algunos de los modelos empleados, a fin de contrastar el comportamiento y la estabilidad de los resultados obtenidos en R. Para ello, se ha utilizado la herramienta *SAS Enterprise Miner Workstation*.

Dado que *SAS Enterprise Miner* no contempla todos los modelos utilizados en el presente trabajo, se han seleccionado tres modelos de referencia que guardan relación con las metodologías implementadas: regresión lineal, red neuronal y *gradient boosting*. Estos modelos representan, de forma genérica, las tres categorías principales tratadas: regresión lineal tradicional, técnicas de ensamblado por *boosting* y modelos no lineales a través de redes neuronales. En todos los casos, se ha intentado replicar la configuración de los hiperparámetros del modelo ganador correspondiente, en la medida en que el entorno lo ha permitido.

Cabe destacar que, en SAS, no siempre se dispone del resultado de R^2 ni del RMSE de forma explícita para todos los modelos. Por ello, se ha optado por emplear la métrica del error cuadrático medio en SAS (ASE, *Average Squared Error*) como criterio común para evaluar el desempeño de los modelos en este entorno (véase figura A.6 en el apéndice de esta memoria).

Figura 11.3: Resumen de errores 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

Fuente: Elaboración propia

Como resultado, se observa que los modelos de *Machine Learning* presentan un mejor control del error en comparación con el modelo tradicional de regresión lineal. En particular, el modelo de *Gradient Boosting* (ASE = 0,0357) obtiene un error ligeramente inferior al modelo de *Red Neuronal* (ASE = 0,0483), lo que podría deberse a pequeñas diferencias en la configuración de los hiperparámetros y a divergencias en los criterios de selección y cálculo entre SAS y R.

Además, aunque en el desarrollo principal del trabajo no se realizó una partición del conjunto de datos en entrenamiento y validación, se ha incorporado este procedimiento de manera complementaria para verificar el comportamiento de los modelos en este escenario. Se ha aplicado una partición del conjunto de datos en una proporción 70-30 (entrenamiento-validación).

Figura 11.4: Resumen de errores de validación en 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
	Req2	Req2	Regresión	chamartin	0.090725	0.301205	0.301205

Fuente: Elaboración propia

En este caso, el modelo de *Red Neuronal* muestra un mejor desempeño (ASE = 0,0582) en el conjunto de validación, en comparación con el modelo de *Gradient Boosting* (ASE = 0,0675). No obstante, ambos modelos siguen superando al modelo de regresión lineal en términos de error.

En conclusión, los modelos basados en *Redes Neuronales* y *Gradient Boosting* han demostrado ser capaces de mejorar el rendimiento con respecto al modelo tradicional de regresión lineal para el conjunto de datos empleado. Asimismo, los resultados obtenidos en R han mostrado ser consistentes y sólidos al ser validados en un entorno distinto como SAS, lo que refuerza su fiabilidad para aplicaciones posteriores.

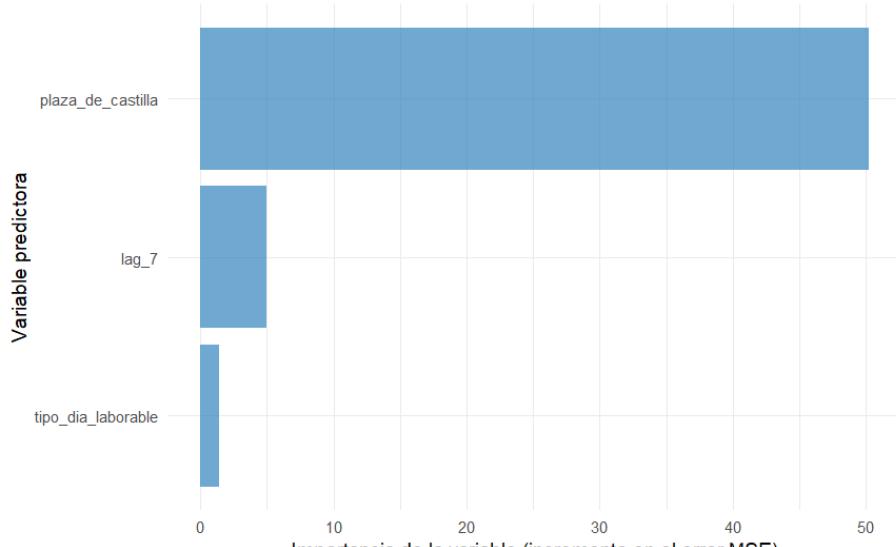
11.3. Interpretabilidad del modelo ganador: *Random Forest*

El modelo *Random Forest*, debido a su carácter de conjunto de árboles generados de manera aleatoria y agregada, no permite una interpretación directa de las relaciones entre variables como en los modelos paramétricos. No obstante, es posible aplicar técnicas de interpretabilidad para analizar la importancia relativa de las variables predictoras.

11.3.1. Importancia de las variables

Aplicando la estimación de importancia global por permutación, mediante la aproximación *SHAP*. Esta técnica cuantifica el incremento en el error cuadrático medio (MSE) al permutar aleatoriamente los valores de cada variable, manteniendo constante el resto de predictores²⁵.

Figura 11.5: Importancia de las variables según incremento en el error MSE



Fuente: Elaboración propia

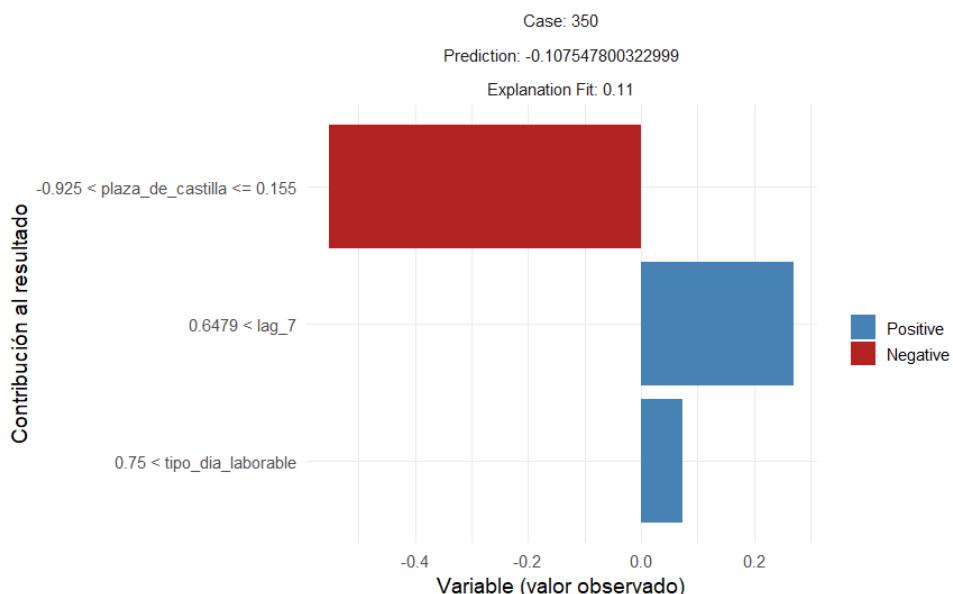
La Figura 11.5 muestra que la variable *plaza_de_castilla* es, con diferencia, la más influyente, con un incremento del MSE de 50 unidades. Le sigue *lag_7*, con un im-

pacto menor (alrededor de 5 unidades), mientras que *tipo_dia_laborable* tiene una influencia prácticamente despreciable en el desempeño del modelo.

11.3.2. Explicación local de la predicción (LIME)

Aplicando la técnica de *LIME*, en la cual consiste en genera un modelo lineal interpretable en torno a una observación específica, aproximando localmente el comportamiento del modelo complejo.

Figura 11.6: Ejemplo de explicabilidad local mediante técnica LIME



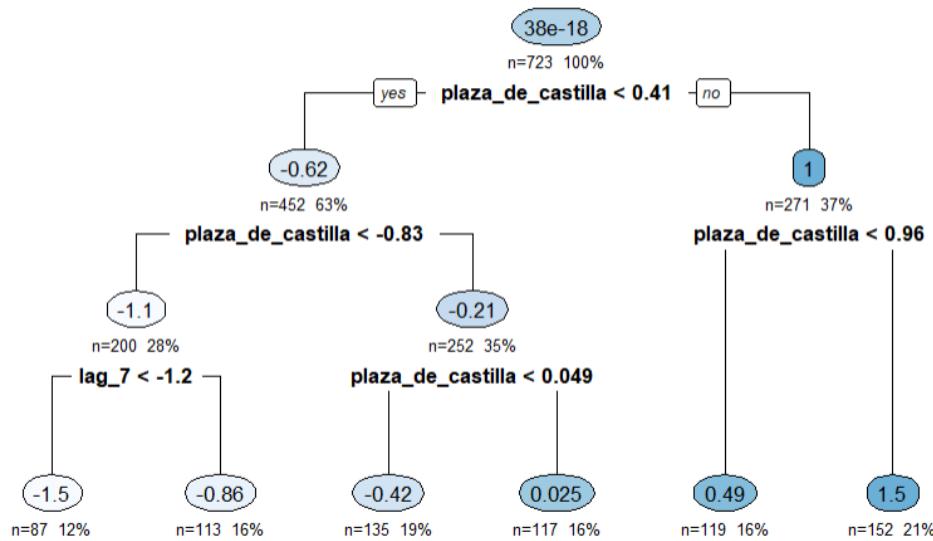
Fuente: Elaboración propia

En la Figura 11.6 se muestra la contribución de cada variable para la predicción de la observación número 350, cuyo valor estimado por el modelo es -0.1075. Se puede observar que la variable *plaza_de_castilla* se encuentra en un rango entre -0.925 y 0.155, siendo el principal responsable de la disminución en la predicción. Mientras que las variables *lag_7* y *tipo_dia_laborable* contribuyen positivamente, en menor magnitud.

11.3.3. Interpretación mediante un árbol simple

Empleando un árbol simple es posible aproximar el comportamiento del *Random Forest* de una forma ilustrativa sobre los criterios de segmentación empleada.

Figura 11.7: Árbol representativo dentro del modelo *Random Forest*



Fuente: Elaboración propia

Para mejorar la interpretabilidad, se ha establecido el parámetro de complejidad $cp = 0.01$, evitando que el árbol crezca innecesariamente si la mejora del error no supera el 1 %. El árbol resultante indica que la variable principal utilizada en la división es *plaza_de_castilla*, en línea con los resultados obtenidos mediante SHAP y LIME.

Desde este nodo raíz, el árbol se ramifica según distintos umbrales de dicha variable. En la rama izquierda, se generan nuevas divisiones si el valor de *plaza_de_castilla* es inferior a 0,41 y luego menor que -0,83, momento en el cual entra en juego la variable *lag_7*, identificada como la segunda más relevante. Con un valor menor de cp , es probable que también hubiera aparecido una ramificación asociada a *tipo_dia_laborable*.

11.4. Otras consideraciones sobre el modelo óptimo

Mediante la comparación crítica según las métricas de RMSE y R^2 en la Sección 11.1, se ha seleccionado al modelo *Random Forest* como el mejor del análisis. No obstante, dicho modelo presenta un RMSE de 0.2251605 y un R^2 de 0.9487059, mientras que el modelo de regresión lineal, utilizado como comparativa, obtiene un RMSE de 0.280276 y un R^2 de 0.9221544.

Dado que la mejora en el rendimiento es reducida, el uso de un modelo más complejo como *Random Forest* podría no justificarse en un entorno de producción, donde la interpretabilidad y la simplicidad son aspectos clave. Esta mínima diferencia en resultados implica trabajar con un modelo más difícil de explicar y de entender en cuanto a la relación entre la variable objetivo “Chamartín” y las variables predictoras seleccionadas.

Por tanto, en un escenario real de despliegue, priorizando la menor complejidad y una mayor capacidad explicativa, debería mantenerse el modelo de regresión lineal como la opción más adecuada.

Figura 11.8: 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 .

Fuente: Elaboración propia

Teniendo en cuenta los coeficientes estimados, el modelo de regresión lineal permite interpretar fácilmente las relaciones entre la variable objetivo y los predictores, mediante la ecuación generada a partir de dichos coeficientes:

$$\begin{aligned} \text{Chamartín} = & 0,23769 + 0,88303 \times \text{plaza_de_castilla} \\ & + 0,24348 \times \text{lag_7} - 0,35142 \times \text{tipo_dia_laborable} \end{aligned}$$

Capítulo 12

Conclusión

El objetivo de este trabajo fue construir y analizar modelos predictivos aplicados a datos temporales de afluencia en estaciones del Metro de Madrid, con el fin de interpretar los patrones de demanda de viajeros. A partir de la revisión del estado del arte en el ámbito del transporte, se identificaron retos relevantes para el desarrollo del objetivo del trabajo que guiaron a la incorporación de variables espaciales, temporales y climatológicas al conjunto de datos original para mejorar la variedad de variables.

Siguiendo las fases de la metodología CRISP-DM, se realizó un proceso riguroso de limpieza, imputación, estandarización y mejora de la calidad de los datos, aplicando técnicas para reducir la multicolinealidad y seleccionar las variables más relevantes.

En la fase de modelado, se implementaron modelos lineales, de ensamblado y no lineales (como redes neuronales o métodos kernel), ajustando los hiperparámetros mediante *fine tuning* y validación cruzada repetida. De los ocho modelos comparados, *Random Forest* obtuvo el mejor rendimiento en términos de RMSE y R^2 .

No obstante, considerando la fase de despliegue y la necesidad de interpretabilidad, el modelo de regresión lineal se presenta como la opción más adecuada. Aunque su rendimiento es ligeramente inferior, permite explicar fácilmente la relación entre variables mediante una ecuación explícita, cumpliendo con el objetivo principal del trabajo: comprender los factores que influyen en la afluencia en la estación de Chamartín.

Capítulo 13

Limitaciones y posibles líneas de trabajo futuro

Una de las principales limitaciones encontradas en este trabajo ha sido la escasa variedad de variables predictoras disponibles, ya que únicamente se contaba con los datos diarios del número de entradas por estación. Además, el periodo temporal registrado no es lo suficientemente amplio como para permitir el desarrollo de modelos más sofisticados, orientados a grandes volúmenes de datos. Este conjunto de datos relativamente reducido ha dificultado la fase de *fine tuning*, limitando la capacidad de aprendizaje de los modelos más complejos.

Como consecuencia, para garantizar el correcto funcionamiento y desempeño de los modelos, ha sido necesario dedicar un esfuerzo considerable a las tareas de preparación, limpieza y ajuste del conjunto de datos, con el fin de adecuarlo a un marco de trabajo suficientemente robusto.

No obstante, este trabajo representa un primer acercamiento al uso de diferentes modelos de *Machine Learning* aplicados a variables de naturaleza espacial, temporal y climatológica en su conjunto. De cara a futuras investigaciones, sería recomendable trabajar con un conjunto de datos más extenso y con mayor diversidad de variables explicativas, lo que permitiría implementar modelos más avanzados, incluyendo técnicas de ensamblado para combinar diferentes tipos de modelos. Teniendo en cuenta el buen rendimiento obtenido por el modelo *Random Forest*, este podría integrarse como componente base dentro de enfoques más complejos de modelado.

Bibliografía

- [1] Agencia Estatal de Meteorología (AEMET). Productos y servicios de opendata aemet [conjunto de datos: Climatologías diarias]. AEMET, 2025. Recuperado de <https://opendata.aemet.es/centrodedescargas/productosAEMET>.
- [2] Ayuntamiento de Madrid. Calendario laboral de madrid [conjunto de datos: Calendario laboral (2013–2025)]. Portal de Datos Abiertos del Ayuntamiento de Madrid, 2025. Recuperado de <https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?page=2&vgnextoid=9f710c96da3f9510VgnVCM2000001f4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default>.
- [3] H. Behrooz and Y. M. Hayeri. Machine learning applications in surface transportation systems: A literature review. *Applied Sciences*, 12(18):9156, 2022. Recuperado de <https://doi.org/10.3390/app12189156>.
- [4] BOAM. Plan de movilidad sostenible madrid 360. Documento institucional del Ayuntamiento de Madrid, 2022. Recuperado de https://sede.madrid.es/csvfiles/UnidadesDescentralizadas/UDCBOAM/Contenidos/Boletin/2022/Anexos%202022/3.1.%20Anexo%20Madrid%20360%20BOAM_.pdf.
- [5] A. Boukerche and J. Wang. Machine learning-based traffic prediction models for intelligent transportation systems. *Computer Networks*, 181:107530, 2020.
- [6] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. Recuperado de <https://doi.org/10.1023/A:1010933404324>.
- [7] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. Association for Computing Machinery, 2016. Recuperado de <https://doi.org/10.1145/2939672.2939785>.
- [8] R. de la Torre, C. G. Corlu, J. Faulin, B. S. Onggo, and A. A. Juan. Simulación, optimización y aprendizaje automático en sistemas de transporte sostenibles: modelos y aplicaciones. *Sustainability*, 13(3):1551, 2021. Recuperado de <https://doi.org/10.3390/su13031551>.
- [9] Ministerio de Vivienda y Agenda Urbana. Áreas urbanas en España 2024. Secretaría General de Agenda Urbana, Vivienda y Arquitectura, 2024. Recuperado de <https://publicaciones.transportes.gob.es/areas-urbanas-en-espa%C3%B1a-2024>.
- [10] F. R. Di Torrepadula, E. V. Napolitano, S. Di Martino, and N. Mazzocca. Machine learning for public transportation demand prediction: A systematic literature review. *Engineering Applications of Artificial Intelligence*, 137:109166, 2024.

- [11] European Commission – Cities Mission. Cities mission implementation plan. Documento en línea, 2021. Recuperado de https://research-and-innovation.ec.europa.eu/system/files/2021-09/cities_mission_implementation_plan.pdf.
- [12] M. Gan, S. Pan, Y. Chen, C. Cheng, H. Pan, and X. Zhu. Application of the machine learning lightgbm model to the prediction of the water levels of the lower columbia river. *Journal of Marine Science and Engineering*, 9(5):496, 2021. Recuperado de <https://doi.org/10.3390/jmse9050496>.
- [13] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [14] A. K. Haghishat, V. Ravichandra-Mouli, P. Chakraborty, Y. Esfandiari, S. Arabi, and A. Sharma. Applications of deep learning in intelligent transportation systems. *Journal of Big Data Analytics in Transportation*, 2:115–145, 2020.
- [15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
- [16] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2nd edition, 2021.
- [17] G. James, D. Witten, T. Hastie, and R. Tibshirani. Linear model selection and regularization. In *An Introduction to Statistical Learning: With Applications in R*, pages 225–288. Springer, 2021.
- [18] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [19] M. Kuhn. caret: Classification and regression training (versión 6.0-94) [manual de paquete r]. The Comprehensive R Archive Network (CRAN), 2024. Recuperado de <https://cran.r-project.org/web/packages/caret/caret.pdf>.
- [20] M. Kuhn and K. Johnson. *Applied Predictive Modeling*, volume 26. Springer, New York, 2013.
- [21] M. B. Kursa and W. R. Rudnicki. Feature selection with the boruta package. *Journal of Statistical Software*, 36:1–13, 2010.
- [22] V. Lagani, G. Athineou, A. Farcomeni, M. Tsagris, and I. Tsamardinos. Feature selection with the r package mxm: Discovering statistically equivalent feature subsets. *Journal of Statistical Software*, 80(7):1–25, 2016.
- [23] V. W. Lumumba, D. Kiprotich, N. G. Makena, M. D. Kavita, and M. L. Mpaine. Comparative analysis of cross-validation techniques: Loocv, k-folds cross-validation, and repeated k-folds cross-validation in machine learning models. *American Journal of Theoretical and Applied Statistics*, 13:127–137, 2024.

- [24] Metro de Madrid. Informe corporativo. Informe institucional, 2021. Recuperado de <https://www.metromadrid.es/sites/default/files/documentos/Informecorporativo2021ESP.pdf>.
- [25] C. Molnar. *Interpretable Machine Learning*. Lulu.com, 2020.
- [26] M. Z. Naser and A. Alavi. Insights into performance fitness and error metrics for machine learning. arXiv preprint arXiv:2006.00887, 2020. Recuperado de <https://arxiv.org/abs/2006.00887>.
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Ensemble methods — scikit-learn 1.4.2 documentation. Documentación web, 2025. Recuperado de <https://scikit-learn.org/stable/modules/ensemble.html>.
- [28] A. Rotondo and F. Quilligan. Evolution paths for knowledge discovery and data mining process models. *SN Computer Science*, 1:109, 2020. Recuperado de <https://doi.org/10.1007/s42979-020-0117-6>.
- [29] C. Schröer, F. Kruse, and J. M. Gómez. A systematic literature review on applying crisp-dm process model. *Procedia Computer Science*, 181:526–534, 2021. Recuperado de <https://doi.org/10.1016/j.procs.2021.01.199>.
- [30] N. Shrestha. Detecting multicollinearity in regression analysis. *American Journal of Applied Mathematics and Statistics*, 8(2):39–42, 2020.
- [31] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.
- [32] M. Steurer, R. J. Hill, and N. Pfeifer. Metrics for evaluating the performance of machine learning based automated valuation models. *Journal of Property Research*, 38(2):99–129, 2021.
- [33] A. Thakur and A. Konde. Fundamentals of neural networks. *International Journal for Research in Applied Science and Engineering Technology*, 9(VIII):407–426, 2021.
- [34] Y. Wang, Z. Cui, and R. Ke. *Machine learning for transportation research and applications*. Elsevier, 2023.
- [35] Y. C. Wu and J. W. Feng. Development and application of artificial neural network. *Wireless Personal Communications*, 102:1645–1656, 2018.

□ % +—————+—————+

Apéndice A

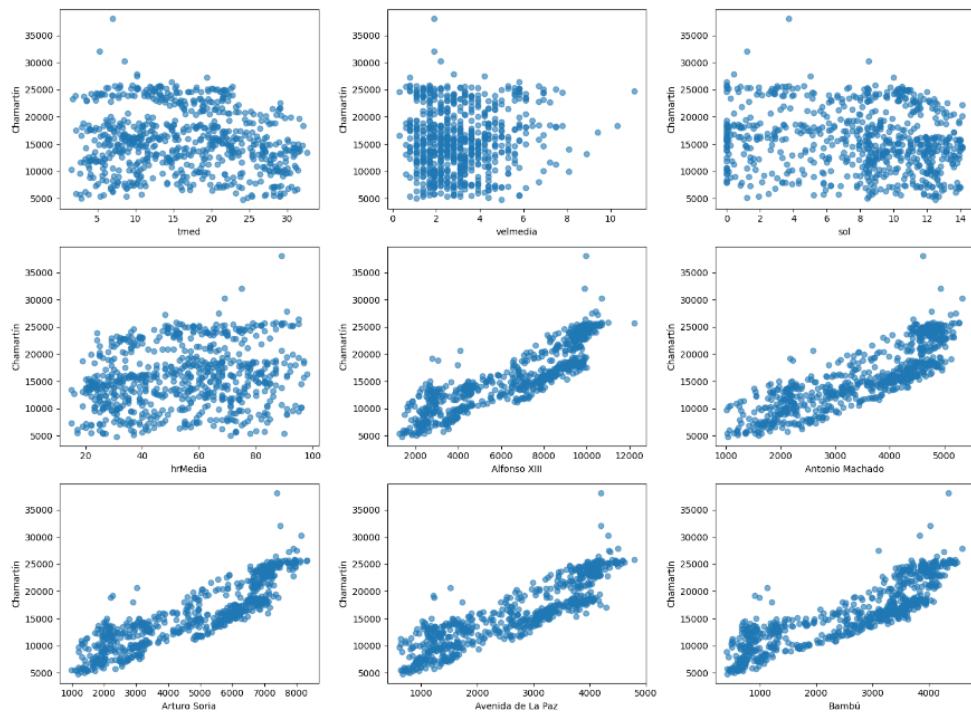
Material adicional

Figura A.1: Resumen estadístico descriptivos

	fecha	Dia_semana	tipo_dia	tmed	prec	velmedia	sol	hrMedia	Alfonso XIII	Antonio Machado	\
count	730	730	730	730.000000	712.0	729.000000	727.000000	730.000000	730.000000	730.000000	
unique	NaN	7	3	NaN	79.0	NaN	NaN	NaN	NaN	NaN	
top	NaN	domingo	laborable	NaN	0.0	NaN	NaN	NaN	NaN	NaN	
freq	NaN	105	493	NaN	547.0	NaN	NaN	NaN	NaN	NaN	
mean	2022-12-31 12:00:00	NaN	NaN	16.271781	NaN	3.042936	7.850894	54.530137	6514.709589	3480.468493	
min	2022-01-01 00:00:00	NaN	NaN	1.600000	NaN	0.300000	0.000000	15.000000	1176.000000	910.000000	
25%	2022-07-02 06:00:00	NaN	NaN	9.500000	NaN	1.900000	4.550000	36.000000	3813.750000	2345.000000	
50%	2022-12-31 12:00:00	NaN	NaN	15.650000	NaN	2.800000	8.700000	55.500000	7177.000000	3741.500000	
75%	2023-07-01 18:00:00	NaN	NaN	22.400000	NaN	3.900000	11.350000	71.000000	9197.250000	4543.500000	
max	2023-12-31 00:00:00	NaN	NaN	32.600000	NaN	11.100000	14.100000	98.000000	12231.000000	5330.000000	
std	NaN	NaN	NaN	8.036595	NaN	1.624320	4.216764	20.928575	2826.786522	1174.430295	
	Arturo Soria	Avenida de la Paz	Bambú	Barrio del Pilar	Begoña	Chamartín	Colombia	Concha Espina	Cruz del Rayo	\	
count	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	697.000000	697.000000		
unique	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	4795.691781	2631.406849	2516.553425	7445.609589	10049.460274	15602.276712	11065.684932	3538.192253	3057.527977		
min	912.000000	490.000000	332.000000	581.000000	1976.000000	3455.000000	1933.000000	682.000000	570.000000		
25%	2876.000000	1490.500000	1178.500000	5792.500000	5319.500000	11598.250000	6545.000000	2014.000000	1937.000000		
50%	5189.500000	2822.000000	2873.000000	7875.000000	10948.500000	15270.500000	11700.500000	4017.000000	3353.000000		
75%	6703.500000	3738.750000	3609.000000	9112.000000	14369.750000	19245.250000	15194.750000	4884.000000	4122.000000		
max	8321.000000	4797.000000	4593.000000	11556.000000	17653.000000	38099.000000	19432.000000	9423.000000	5051.000000		
std	2055.716618	1160.352079	1263.035262	2052.773314	4591.050263	5673.205828	4744.146866	1521.070341	1234.903293		
	Cuzco	Duque de Pastrana	Estrecho	Fuencarral	Herrera Oria	Hortaleza	Manoteras	Nuevos Ministerios	Peñagrande	\	
count	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000		
unique	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	8350.669863	1571.982192	12661.536986	4094.149315	3632.038356	3494.730137	2420.253425	37737.884932	5574.002740		
min	1451.000000	222.000000	3676.000000	1145.000000	521.000000	890.000000	754.000000	7167.000000	1566.000000		
25%	4627.500000	723.750000	9763.750000	3015.500000	2147.500000	2423.500000	1736.500000	24977.500000	4130.000000		
50%	9591.500000	1683.000000	13160.000000	4328.000000	4052.000000	3526.500000	2558.500000	40454.000000	5924.500000		
75%	11528.750000	2375.500000	15576.000000	5169.750000	5043.000000	4635.250000	3121.750000	50325.750000	7032.250000		
max	14776.000000	2751.000000	18931.000000	6316.000000	5885.000000	5673.000000	3864.000000	62681.000000	8810.000000		
std	3702.505504	818.755882	3352.213121	1187.780232	1493.975977	1150.357772	742.828944	14327.307857	1636.248523		
	Pinar de Chamartín	Pinar del Rey	Plaza de Castilla	Pio XII	Santiago Bernabéu	Tetuán	Valdeacederas	Valdezarza	Ventilla	\	
count	730.000000	626.000000	730.000000	730.000000	729.000000	730.000000	730.000000	730.000000	730.000000		
unique	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	10275.717808	3907.175719	36614.826027	3509.239726	10728.943759	9132.606849	6942.800000	3264.132877	3410.549315		
min	2224.000000	977.000000	7746.000000	479.000000	2370.000000	2962.000000	2018.000000	895.000000	241.000000		
25%	6323.500000	2706.250000	24507.000000	1867.250000	7646.000000	7195.250000	5356.250000	2379.250000	2508.250000		
50%	11565.000000	4004.000000	38727.000000	3934.500000	10992.000000	9589.000000	7329.500000	3503.000000	3551.500000		
75%	13991.000000	5249.000000	47863.000000	5010.000000	13252.000000	10998.750000	8433.750000	4069.000000	4469.000000		
max	17654.000000	6140.000000	60386.000000	5686.000000	29515.000000	13450.000000	11578.000000	4990.000000	5065.000000		
std	4145.121728	1355.128192	13150.744814	1604.265977	4311.589306	2227.431886	1829.043346	926.133361	1030.258290		

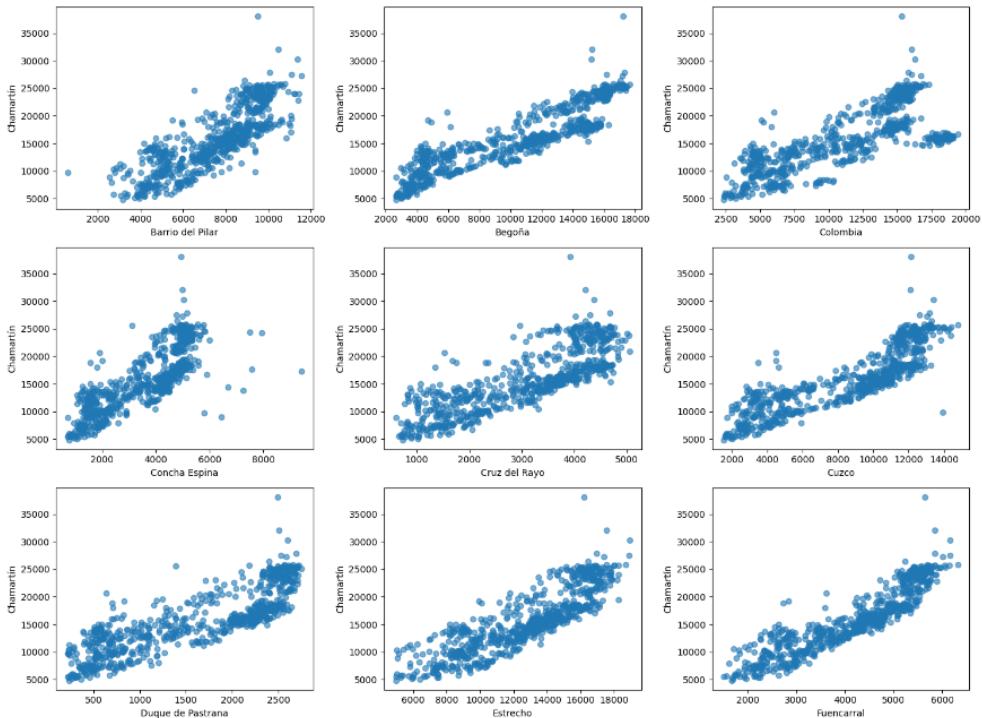
Fuente: Elaboración propia

Figura A.2: Gráfica dispersión 1/4



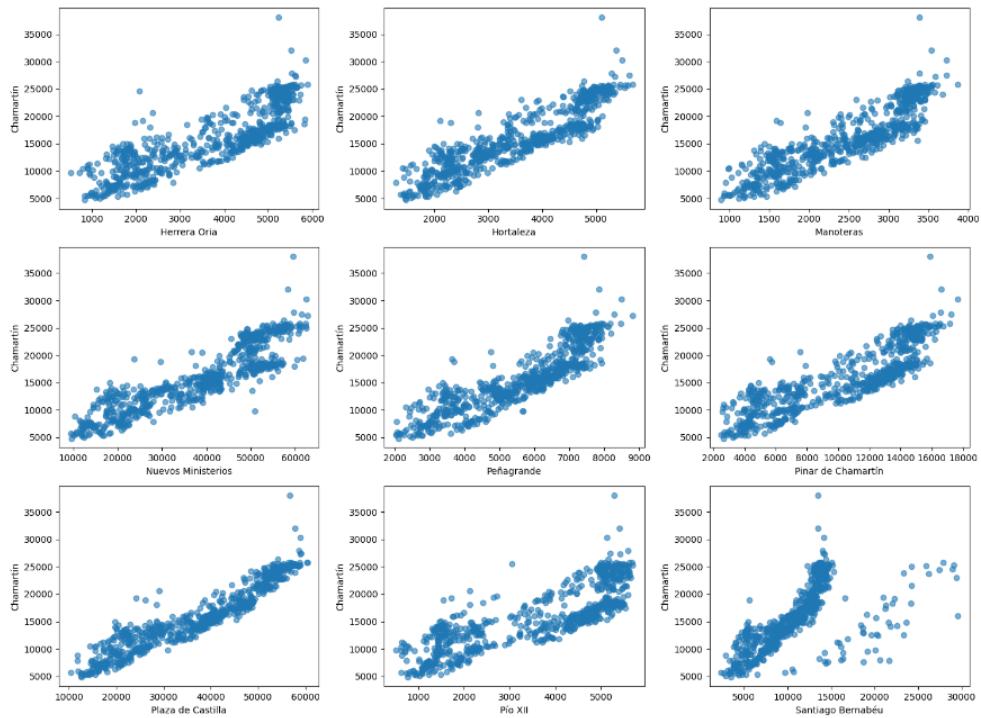
Fuente: Elaboración propia

Figura A.3: Gráfica dispersión 2/4



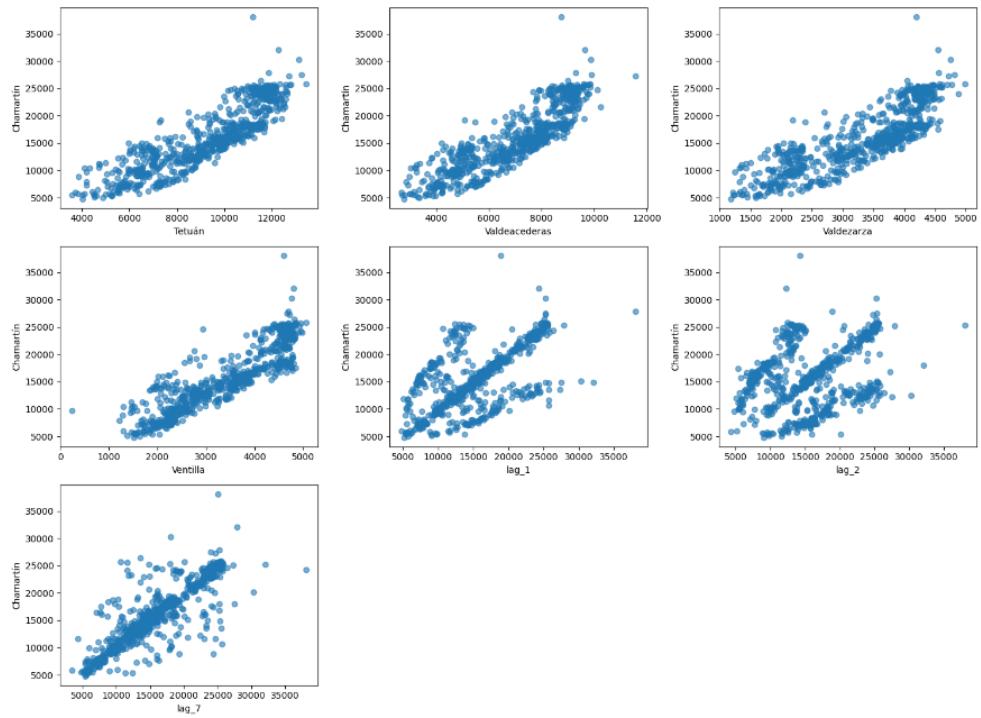
Fuente: Elaboración propia

Figura A.4: Gráfica dispersión 3/4



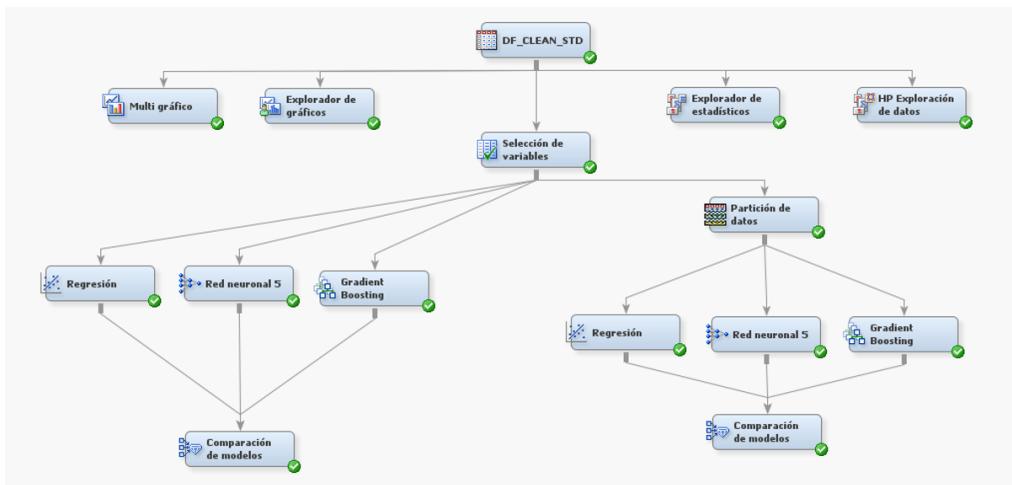
Fuente: Elaboración propia

Figura A.5: Gráfica dispersión 4/4



Fuente: Elaboración propia

Figura A.6: Diagrama SAS



Fuente: Elaboración propia