



UNIVERSITAT OBERTA DE CATALUNYA (UOC)  
MÁSTER UNIVERSITARIO EN CIENCIA DE DATOS

## TRABAJO FINAL DE MÁSTER

ÁREA: 2

### **Optimización del sistema de bicicletas compartidas en la ciudad de Valencia.**

**Análisis predictivo, rutas de reparto para el balanceo y gestión eficiente de las estaciones.**

---

Autor: Jose Luis Santos Durango

Tutor: Raúl Parada Medina

Profesor: Esther Ibáñez Marcelo

---

Valencia, 10 de junio de 2024



# Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento - NoComercial - SinObraDerivada 3.0 España de CreativeCommons.



Los datos con los que se ha trabajado, han sido recopilados mediante la API que proporciona la empresa [JCDecaux](#). Tal como se detalla en la página que proporciona la API, los datos son de código abierto y se comparten bajo una licencia *Open source* en un contexto internacional, de modo que se puede hacer uso libre de los mismos. Los datos históricos han sido cedidos por un tercero, por lo que para cumplir con el Reglamento General de Protección de Datos 2016/679 será necesario:

- Asegurarse de que la información de identificación personal (PII) de los usuarios esté anónimizada y protegida para evitar posibles usos indebidos o acceso no autorizado a datos sensibles.
- Equidad: asegurarse de que el análisis de datos y los procesos de toma de decisiones sean justos y no sesgados, evitando la discriminación o el trato injusto a ciertos grupos en función de sus características demográficas.

Al no tratarse de datos personales, si no, datos meramente informativos distribuidos por la empresa que brinda el servicio, podemos asegurar la protección de los mismos.



# FICHA DEL TRABAJO FINAL

Título del trabajo:	Optimización del sistema de bicicletas compartidas en la ciudad de Valencia.
Subtítulo del trabajo:	Análisis predictivo, rutas de reparto para el balanceo y gestión eficiente de las estaciones.
Nombre del autor:	Jose Luis Santos Durango
Nombre del colaborador/a docente:	Raúl Parada Medina
Nombre del PRA:	Esther Ibáñez Marcelo
Fecha de entrega (mm/aaaa):	06/2024
Titulación o programa:	Máster Universitario en Ciencia de Datos
Área del Trabajo Final:	M2.879 - Trabajo Final de Máster - Área 2
Repositorio GitHub	<a href="#">TFM - GitHub repository</a>
Idioma del trabajo:	Español
Palabras clave	<i>Machine Learning, Smart City, Bike Sharing</i>



# Agradecimientos

*A mis matemáticos de la UCM, en especial a mi amiga Patricia, que me recomendó este máster y que siempre me ha dado y me seguirá dando tan buenos consejos.*

*A Arantxa, que aunque nos dispersemos juntos a veces, ella me aclara y orienta como buena psicóloga y amiga que es.*

*También me gustaría expresar mis agradecimientos a Alex Barros ([alex@lexbar.es](mailto:alex@lexbar.es)) por la cesión de los datos históricos de Valenbisi, datos que han sido fundamentales para la realización de este proyecto de investigación.*



# Resumen

Este trabajo de investigación es fruto de la motivación personal que me surge al suscribirme al sistema de bicicletas compartidas que ofrece la empresa **Valenbisi** en la ciudad de Valencia, elegida como la mejor ciudad del mundo según la revista **Forbes** y seleccionada como la Capital Verde Europea en 2024 por la **Comisión Europea**. Además, Valencia es una ciudad comprometida con los objetivos de desarrollo sostenible y es la tercera ciudad de España en el ranking de longitud de carril bici. Todo esto, sumado a la posibilidad de aplicar los conocimientos que me han brindado las distintas asignaturas del máster de ciencia de datos, son el resultado de este proyecto de investigación.

El principal problema que se da en los sistemas de bicicletas compartidas (BSS), es el balanceo de estaciones. Ciudades como Oslo, ofrecen recompensas a los usuarios por aparcar las bicicletas en estaciones cercanas con mayor número de bornetas libres. Sin embargo, esta solución no puede evitar que el sistema esté desbalanceado y que por lo tanto, los usuarios puedan disfrutar de un servicio óptimo. Por este motivo, en este proyecto se realizará un estudio para poder estimar las bornetas libres de una estación según el momento del día y poder planificar así una ruta de reparto óptima para el camión que se dedica a balancear las estaciones. Además, con el fin de completar el *dataset*, se hará uso de otras fuentes secundarias de datos: datos meteorológicos, datos geográficos sobre la zona de las estaciones y datos económicos de los barrios de la ciudad.

Con los datos procesados se aplicarán técnicas de *Machine Learning* como los árboles de decisión y los bosques aleatorios de clasificación, que usaremos para poder contestar la pregunta que se plantea en el proyecto: ¿cuándo una estación necesitará ser atendida por el camión de reparto?. Entre los modelos seleccionados y las variables a predecir, elegiremos el que mejor resultados nos proporcione, para posteriormente, optimizar la ruta de reparto.

**Palabras clave:** *Machine Learning, Smart City, Bike Sharing*



# Abstract

This research work is the result of personal motivation that arose when I subscribed to the bike-sharing system offered by [Valenbisi](#) in the city of Valencia, which was chosen as the best city in the world according to [Forbes](#) and selected as the European Green Capital in 2024 by the [European Commission](#). Additionally, Valencia is a city committed to sustainable development goals and is the third city in Spain in terms of bike lane length. All of this, combined with the opportunity to apply the knowledge gained from various subjects in the master's program in data science, has resulted in this research project.

The main issue with bike-sharing systems (BSS) is station balancing. Cities like Oslo offer rewards to users for parking bikes at nearby stations with more available docks. However, this solution does not ensure that the system remains balanced, therefore failing to provide optimal service to users. Therefore, this project will conduct a study to estimate the number of available docks at a station based on the time of day, and to plan an optimal distribution route for the truck responsible for balancing the stations. Moreover, in order to complete the *dataset*, other secondary data sources will be used: meteorological data, geographical data about the station areas, and economic data of the city's neighborhoods.

With the processed data, we will apply *Machine Learning* techniques such as decision trees and random forests classifiers, which we will be used to answer the project's central question: when will a station need to be serviced by the delivery truck? Among the selected models and the variables to be predicted, we will choose the one that provides us with the best results to subsequently apply optimization techniques to the delivery route.

**Keywords:** *Machine Learning, Smart City, Bike Sharing*

**X**

---

# Índice general

<b>Resumen</b>	<b>VII</b>
<b>Abstract</b>	<b>IX</b>
<b>Índice</b>	<b>XI</b>
<b>Índice de Figuras</b>	<b>XV</b>
<b>Índice de Tablas</b>	<b>1</b>
<b>1. Introducción</b>	<b>2</b>
1.1. Descripción y justificación del trabajo . . . . .	2
1.2. Objetivos del trabajo . . . . .	3
1.3. Sostenibilidad, Ética y Diversidad . . . . .	3
1.4. Metodología del desarrollo . . . . .	4
1.5. Planificación del proyecto . . . . .	5
1.6. Productos obtenidos . . . . .	7
1.7. Resumen de los capítulos . . . . .	8
<b>2. Estado del arte</b>	<b>10</b>
2.1. Sistemas de bicicletas compartidas . . . . .	10
2.1.1. Problemática de un BSS . . . . .	10
2.1.2. Soluciones desde el enfoque de análisis de datos . . . . .	11
2.2. Literatura preliminar . . . . .	12
2.3. Comparativa del estado del arte . . . . .	14
2.4. Planteamiento del problema . . . . .	15
<b>3. Procesamiento de los datos</b>	<b>17</b>
3.1. Tecnologías empleadas . . . . .	17
3.1.1. Tecnologías de desarrollo . . . . .	18

3.1.2. Tecnologías de análisis . . . . .	19
3.2. Conjuntos de datos . . . . .	20
3.2.1. Tratamiento de datos de fuentes externas a Valenbisi . . . . .	21
3.2.2. Tratamiento de datos históricos de Valenbisi . . . . .	24
3.2.3. Tratamiento de datos históricos de Valenbisi con Spark . . . . .	24
<b>4. Analítica descriptiva y calidad de los datos</b>	<b>26</b>
4.1. Datos meteorológicos . . . . .	26
4.1.1. Análisis de los datos . . . . .	26
4.1.2. Imputación de valores nulos . . . . .	27
4.1.3. Imputación de valores outliers . . . . .	28
4.1.4. Correlación entre variables . . . . .	29
4.2. Datos de las estaciones: geográficos, económicos y zonas verdes . . . . .	30
4.2.1. Análisis de valores outliers . . . . .	30
4.2.2. Evaluación de outliers en zonas verdes de grandes dimensiones . . . . .	32
4.3. Reducción de la dimensionalidad de los datos . . . . .	32
4.3.1. Evaluación de los datos sobre movimientos: registros erróneos . . . . .	33
4.3.2. Estaciones de referencia . . . . .	34
<b>5. Análisis predictivo de los datos. Atributos, separación de conjuntos y modelos de predicción</b>	<b>36</b>
5.1. Tratamiento de los atributos del conjunto de datos final . . . . .	36
5.1.1. Elección de los atributos . . . . .	36
5.1.2. Atributos adicionales de los datos . . . . .	37
5.2. Preparación de los conjuntos de entrenamiento y testeo . . . . .	39
5.3. Modelos predictivos . . . . .	43
5.3.1. Regresión multipolinomial . . . . .	43
5.3.2. Regresión logística . . . . .	46
5.3.3. Árboles de decisión . . . . .	48
5.3.4. Bosques aleatorios de árboles de decisión . . . . .	51
5.4. Modelo de predicción seleccionado . . . . .	52
<b>6. Optimización de la ruta de reparto</b>	<b>53</b>
6.1. Conclusiones sobre la optimización de la ruta . . . . .	57
<b>7. Conclusiones</b>	<b>58</b>
7.1. Valoración personal . . . . .	59
7.2. Trabajos futuros . . . . .	60

**8. Glosario****61****Bibliografía****61**



# Índice de figuras

1.1. CRISP-DM Methodology . . . . .	4
1.2. Diagrama de Gantt con la planificación del proyecto . . . . .	6
3.1. Mapa de concentración de valores nulos sobre datos atmosféricos . . . . .	22
4.1. Valores outliers de los datos meteorológicos. . . . .	28
4.2. Distribución de los valores mayor de 400 de la variable radiación por mes, día y hora. . . . .	29
4.3. Matriz de correlación de las variables atmosféricas. . . . .	30
4.4. Estudios de outliers en datos geográficos. . . . .	31
4.5. Mapa de las zonas verdes consideradas outliers. . . . .	32
4.6. Mapa de las estaciones resultantes del estudio. . . . .	35
6.1. Conteo de estaciones que precisan balanceo. . . . .	54
6.2. Ruta de reparto optimizada para un Lunes a las 07:00 A.M. . . . .	56
6.3. Ruta de reparto optimizada para un Martes a las 00:00 A.M. . . . .	56
6.4. Ruta de reparto optimizada para un Jueves a las 07:00 A.M. . . . .	57



# Índice de cuadros

2.1. Tabla comparativa de estudios relacionados. . . . .	14
4.1. Valores nulos y outliers de los datos meteorológicos. . . . .	27
4.2. Valores outliers y porcentaje de valores outliers de datos de estaciones. . . . .	31
5.1. Tipos de datos y descripciones por categoría. . . . .	37
5.2. Datos codificados. . . . .	38
5.3. Valores mínimos y máximos de las variables . . . . .	38
5.4. Conjuntos de entrenamiento y testeo . . . . .	42
5.5. Resultados de regresión de bikes_avg con diferentes grados de polinomio . . . . .	45
5.6. Resultados de predicción con regresión logística . . . . .	48
5.7. Resultados de predicción con árboles de decisión con máxima profundidad de 10	49
5.8. Resultados de predicción con árboles de decisión con máxima profundidad de 20	50
5.9. Resultados de predicción con árboles de decisión con máxima profundidad de 30	50
5.10. Mejores resultados para un bosque aleatorio de árboles de decisión . . . . .	52
5.11. Resultados de predicción para DataSet3 con 100 estimadores y profundidad de 30	52

# **Capítulo 1**

## **Introducción**

### **1.1. Descripción y justificación del trabajo**

Este proyecto surge tras la motivación personal para contribuir a la mejora de un servicio de bicicletas compartidas en la ciudad en la que resido actualmente. Tras abonarme al servicio y comprobar que el problema de balanceo en algunas estaciones es una gran desventaja, me surge la idea de hacer un análisis del servicio para optimizarlo y plantear una solución útil con un importante impacto social en la ciudad. Siguiendo con la dinámica de la Universidad Oberta de Catalunya con el compromiso ético global y los objetivos de desarrollo sostenible, teniendo en cuenta, como ya he mencionado anteriormente, que Valencia es la capital verde europea en 2024 [1], me parece un tema bastante interesante a analizar. Con el objetivo de hacer las ciudades cada vez más sostenibles, son muchos los gobiernos que deciden implantar un servicio de bicicletas compartidas en sus ciudades, reduciendo así el uso del coche y por ende, las emisiones de CO<sub>2</sub>. Sin embargo, estos sistemas de bicicletas compartidas presentan la problemática de no poder aparcar la bicicleta en estaciones que están llenas, o no poder coger una bicicleta cuando una estación está vacía. Por este motivo, muchos usuarios prefieren optar por otros medios de transporte más fiables, pero menos comprometidos con el medio ambiente. Ante este problema, me planteo emplear la ciencia de datos para poder ofrecer una solución a la empresa que gestiona el servicio y que así aumente el número de usuarios.

Por otro lado, todo proyecto de análisis de datos necesita una fuente de datos consistente para que podamos aplicar algoritmos con lógica que nos proporcionen estimaciones realistas e insesgadas. Por este motivo, al iniciar la propuesta de este proyecto, lo primero que se ha valorado ha sido la extracción de datos y el formato de los mismos, algo que ha sido posible gracias a la API que proporciona la empresa del BSS y que nos brinda los datos de cada estación con una granularidad del minuto [2].

## 1.2. Objetivos del trabajo

En este proyecto se pretenden trabajar una serie de objetivos con el fin de garantizar una mejora del sistema de bicicletas compartidas y balancear las estaciones optimizando la ruta de la empresa de reparto y estudiando la viabilidad de instalar almacenes de bicicletas en determinados núcleos urbanos.

Por un lado, realizaremos un estudio del estado de las estaciones y el uso del servicio. Para ello recopilaremos los datos temporales y los transformaremos al formato de estudio adecuado, para así poder comprobar las tendencias temporales y aplicar técnicas de *Machine Learning* (ML), de modo que al cruzar los datos de las estaciones con otras fuentes de datos (meteorológicas, socioeconómicas, demográficas, etc.) podamos detectar patrones y entender mejor el contexto en el que se utiliza el sistema.

Por otro lado, otro objetivo, será detectar estaciones con alta demanda y baja disponibilidad, o viceversa. Se realizará un análisis predictivo para estimar las demandas a futuro, generando así un modelo de ML, que entrenaremos con datos históricos. Así podremos evaluar las técnicas de balanceo de las estaciones: reubicación de bicicletas, aumento de estaciones, optimización de la ruta de reparto, viabilidad de creación de almacenes en determinados núcleos urbanos.

## 1.3. Sostenibilidad, Ética y Diversidad

En el marco del desarrollo de este Trabajo de Fin de Máster, es importante resaltar la importancia de la **sostenibilidad** como uno de los pilares fundamentales del proyecto. El fomento del uso de la bicicleta tiene un impacto directo en la reducción de la emisión de gases de efecto invernadero y la mejora de la calidad del aire. Este enfoque está alineado con el objetivo de desarrollo sostenible número 11: ciudades y comunidades sostenibles, tal como se menciona en la sección 1.1. Además contribuye a reducir el impacto del cambio climático.

Por otro lado, siguiendo con los principios de **ética**, cada vez más importante en proyectos de ciencia de datos, es importante destacar que en el proyecto no se trabajan con datos personales ni sensibles, si no con datos de fuentes abiertas que no comprometen a ningún individuo y que por lo tanto, fortalecen la privacidad de los usuarios del sistema y la confianza en el mismo.

Por último, la **diversidad** juega un papel fundamental en la implementación de sistemas de *Smart City*. El desarrollo de este proyecto, permitirá la mejora del sistema en otras ciudades

adaptándolo a las necesidades urbanas de cada ciudad: densidad, rutas viales, demografía, etc. Esto, además de mejorar el impacto positivo del proyecto, también promueve el acceso a medios de transporte sostenibles en otros entornos urbanos.

## 1.4. Metodología del desarrollo

Al tratarse de un proyecto de minería de datos, para su desarrollo nos basaremos en la metodología CRISP-DM [3], por su siglas en inglés *Cross-industry standard process for data mining*. Se trata de la metodología estándar en proyectos de minería de datos que consiste en 6 fases distintas tal como podemos observar en la *Figura 1.1*.

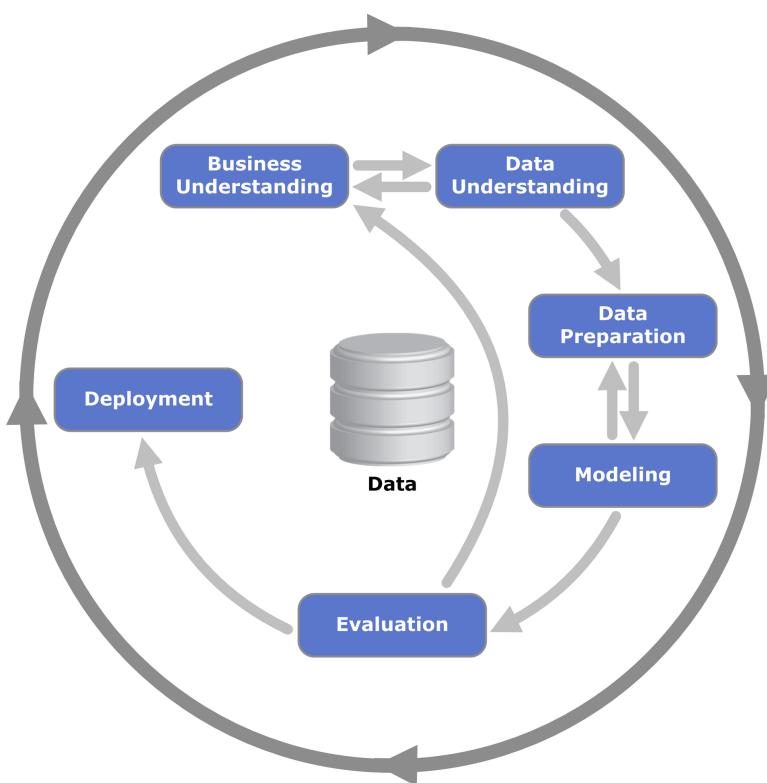


Figura 1.1: CRISP-DM Methodology

Fuente: Wikipedia [3]

- **Business Understanding:** en esta fase será necesaria la comprensión profunda de los objetivos y requisitos del proyecto desde una perspectiva empresarial. En este caso al tratarse de un sistema BSS, tendremos que entender el funcionamiento del mismo, en qué consiste el servicio, los datos relevantes que puede proporcionar, etc.

- **Data Understanding:** en esta fase se explorarán y evaluarán los datos disponibles para comprender su estructura, calidad y posibles problemas.
- **Data preparation:** en esta fase se realizará el proceso ETL (extracción, transformación y carga de los datos) para que los datos tengan la estructura deseada para el análisis.
- **Modeling:** esta fase consiste en evaluar los distintos modelos de ML para ver cuál es el que mejor se ajusta a la solución de nuestro problema.
- **Evaluation:** en esta fase evaluaremos el modelo seleccionado y en caso de que no sea el que mejor se ajusta a nuestros datos, podremos volver al inicio del proceso para elegir el modelo que más aporte al objetivo con las métricas seleccionadas.
- **Deployment:** la última fase consiste en el despliegue del proyecto de negocio. En nuestro caso sería desarrollar el *Business case*, una vez realizado el análisis, haciendo la propuesta de mejora de negocio a la empresa del BSS.

Estas etapas se pueden iterar con los distintos modelos, de forma que iremos adquiriendo un mayor entendimiento del negocio y un mayor conocimiento de los datos que estamos tratando, surgiendo de este modo nuevas perspectivas de negocio [3].

## 1.5. Planificación del proyecto

Para planificar las tareas que se llevarán a cabo en el desarrollo de este proyecto, seguiremos el plan establecido por la UOC mediante las pruebas de evaluación continua (PEC).

Se pueden visualizar las distintas tareas en el diagrama de Gantt de la [Figura 1.2](#).

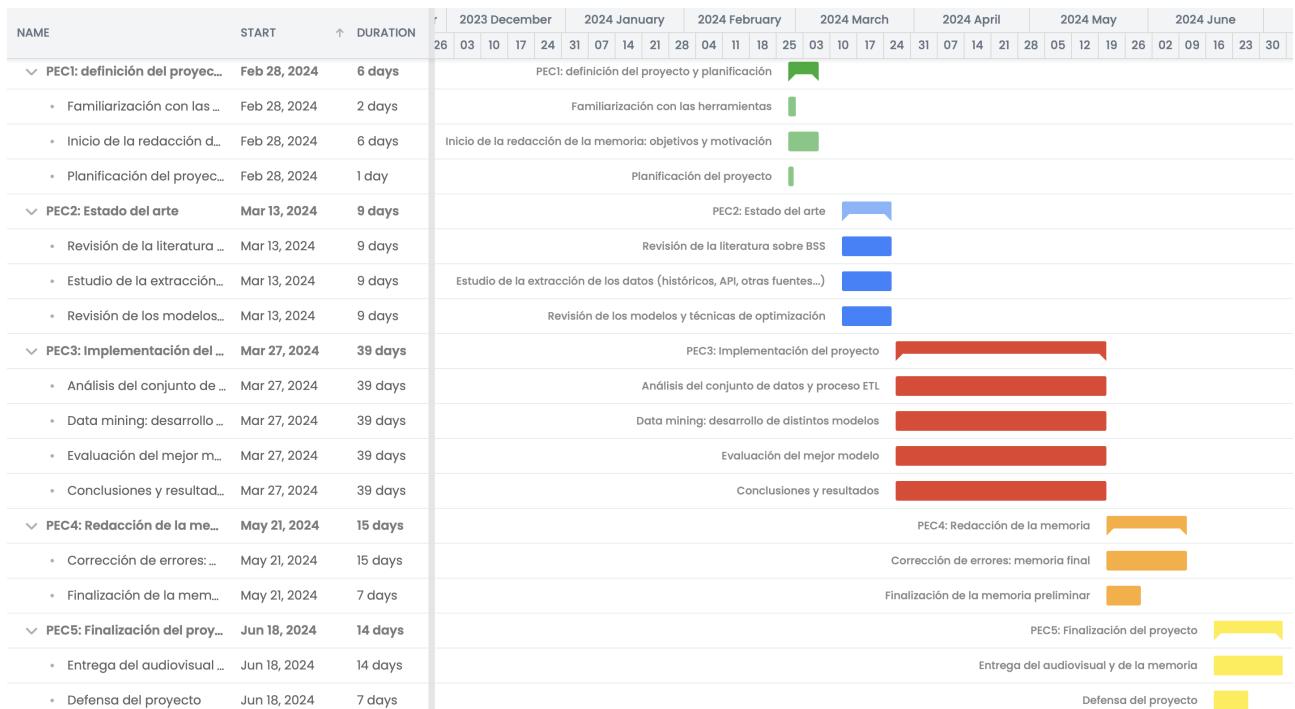


Figura 1.2: Diagrama de Gantt con la planificación del proyecto

Elaborado con: <https://bryntum.com>

Podemos distinguir 5 entregables generales donde desplegaremos distintas tareas para lograr la finalidad del proyecto:

- **Definición del proyecto y planificación:** esta entrega consiste en la instalación del entorno de trabajo para la memoria, el acuerdo con el tutor de las tecnologías que se usarán, el inicio de la redacción de la memoria: título del proyecto, breve resumen del proyecto, introducción (motivación personal, impacto social, objetivos, metodología y planificación temporal del proyecto)
- **Estado del arte:** con este entregable se pretende realizar una investigación de los artículos/proyectos que aborden la temática de optimización de los BSS. Además, haremos un estudio sobre la obtención de los datos y su formato, así como de la propuesta de modelos que podremos usar en la siguiente etapa del proyecto.
- **Implementación del proyecto:** esta es la fase más larga del proyecto. Consiste en la elaboración del proyecto tal como se ha definido, desde la extracción de los datos y su transformación, así como la combinación de distintas fuentes de datos, hasta la evaluación de los modelos de estimación sobre los datos y la optimización de la ruta de reparto. Distinguimos las siguientes tareas a realizar en esta fase:

- **Extracción y preparación de los datos:** en esta tarea elaboraremos el código que permitirá extraer los datos y combinarlos en el formato deseado para su explotación. Se desarrollará en dos fases dado que se va a trabajar con datos masivos y se va a necesitar un servidor externo con recursos suficientes para procesar los datos de alguna de las fuentes.
  - **Análisis de los datos procesados:** esta tarea, tal como su nombre indica, consiste en la elaboración de los gráficos que permitan entender los datos y detectar los posibles fallos que puedan tener: valores anómalos, datos nulos, etc.
  - **Modelos estadísticos:** esta tarea consiste en desarrollar distintos modelos estadísticos para poder estimar los datos deseados sobre los movimientos de bicicletas. Se buscarán distintos modelos y comapararán los resultados, con el fin de seleccionar el mejor modelo para el objetivo que hemos fijado.
  - **Diseño de la ruta óptima:** en esta parte del código, se va a desarrollar el algoritmo de optimización de la ruta de reparto, una vez se han estimado los datos necesarios para poder saber si una estación precisará ser balanceada o no.
- **Redacción de la memoria:** una vez finalizado el proyecto, tendremos que elaborar una memoria donde mostremos los resultados y las conclusiones. En esta fase primero elaboraremos un borrador y después la versión final con las correcciones del tutor.
  - **Finalización del proyecto:** para finalizar el proyecto elaboraremos una presentación en vídeo, exponiendo los puntos clave. Además, se elaborará un repositorio en GitHub con todos los archivos involucrados en este proyecto. Finalmente se hará entrega de la memoria y del repositorio GitHub y se realizará una presentación ante tribunal.

## 1.6. Productos obtenidos

Los productos obtenidos en este proyecto de investigación, son los siguientes *notebooks*:

- **ETL\_other\_data.ipynb:** en este *notebook* vamos a realizar el procesamiento de los datos externos al sistema de Valenbisi con los que queremos cruzar nuestros datos de las estaciones de bicicletas. Para realizar este análisis se han tenido en cuenta distintas cuestiones comentadas con el tutor, considerando las siguientes como las más relevantes en el impacto del uso del sistema de Valenbisi: cuestiones meteorológicas y atmosféricas; datos geolocalizados de las estaciones; zonas verdes; cuestiones socioeconómicas (por barrios de la ciudad).

- **ETL\_datos\_históricos.ipynb:** en este *notebook* vamos a realizar el tratamiento de los datos históricos de Valenbisi. Los datos que tenemos, son de 4 años (2020-2023) con una granularidad de minuto (580 millones de registros aprox.) Vamos a descomprimir los datos de las carpetas que tenemos y después les daremos el formato deseado, creando un data-frame para poder guardarlo en un fichero con todos los datos formateados y compactados en la misma estructura.
- **ETL\_spark.ipynb:** contiene el procesamiento de los datos masivos mediante Spark en un servidor externo de Cloudera.
- **data\_analysis.ipynb:** en este *notebook* se realizarán las tareas correspondientes de análisis de datos y procesado de valores anómalos o nulos. Se estudiará también la distribución de los datos y se definirá el conjunto final con el que se trabajarán las predicciones.
- **data\_prediction.ipynb:** finalmente, se desarrollan los modelos predictivos con los datos resultantes y se analizan los resultados obtenidos para poder aplicar un modelo de predicción, cuyos resultados permitirán desarrollar la optimización de la ruta de reparto.

A parte de estos *notebooks*, se obtiene la presente memoria, donde se recoge la introducción del proyecto con el problema planteado y la literatura previa al respecto, así como el desarrollo del proyecto y los resultados. Además se incluyen posibles trabajos a futuro no contemplados en este proyecto.

## 1.7. Resumen de los capítulos

En el capítulo 1 se presenta una descripción del trabajo, justificándolo. Se establecen los objetivos del proyecto, se discute la importancia de la sostenibilidad, ética y diversidad en el contexto del trabajo. Se describe la metodología de desarrollo seguida y se presenta la planificación del proyecto. Finalmente, se mencionan los productos obtenidos como resultado del trabajo.

El capítulo 2 comienza con una exploración de los sistemas de bicicletas compartidas, abordando sus problemáticas y soluciones desde el enfoque de análisis de datos. Se revisa la literatura preliminar y se realiza una comparativa del estado del arte en el área de estudio. En este apartado plantearemos el problema que se abordará en el análisis.

En el capítulo 3, se detallan las tecnologías utilizadas en el proyecto, tanto en el desarrollo como en el análisis de datos. Se describe el tratamiento de diferentes conjuntos de datos, inclu-

yendo aquellos de fuentes externas a Valenbisi.

En el capítulo 4 se analizan los datos meteorológicos y de las estaciones, evaluando su calidad y realizando técnicas de imputación de valores nulos y outliers. Se aborda también la reducción de la dimensionalidad de los datos.

En el capítulo 5 se presentan los atributos del conjunto de datos final, se muestra la preparación de conjuntos de entrenamiento y testeo, y se exploran diferentes modelos predictivos como la regresión multipolinomial, regresión logística, árboles de decisión y bosques aleatorios. Finalmente, se selecciona un modelo de predicción.

Finalmente, el capítulo 6 se enfoca en la optimización de la ruta de reparto, utilizando los resultados obtenidos del análisis predictivo de los datos.

En el capítulo, 7 se añaden las conclusiones generales obtenidas con la elaboración de este proyecto y un listado con mejoras para futuros proyectos, que no han sido contempladas en el desarrollo.

# **Capítulo 2**

## **Estado del arte**

En esta sección se presentará brevemente el estado del arte sobre la temática de este proyecto de investigación. En una primera parte se va a exponer en qué consiste un sistema de bicicletas compartidas y cuáles son los inconvenientes/problemas que se pretenden mejorar con nuestro proyecto, aplicando técnicas de análisis de datos. La segunda parte consistirá en un listado de proyectos/artículos donde se haya abordado esta misma temática, incluyendo un breve resumen de cada uno de ellos y destacando qué podemos aportar con nuestro proyecto y que aún no haya sido contemplado por el resto. Finalmente, se incluirá una tabla comparativa con los distintos aspectos que cubriremos y se realizará el planteamiento del problema.

### **2.1. Sistemas de bicicletas compartidas**

El uso de los sistemas de bicicletas compartidas ha visto un incremento en los últimos 3 años por diversos motivos: la COVID-19 impacta directamente sobre el uso del transporte público y la preferencia por los espacios abiertos; existe un mayor interés por parte de los gobiernos en cumplir con los Objetivos de Desarrollo Sostenible (ODS), de modo que cada vez son más frecuentes las inversiones en sistemas sostenibles de transporte; son muchas las ciudades donde estos sistemas se integran con el resto de medios de transporte público, facilitando así el acceso a los usuarios [4].

#### **2.1.1. Problemática de un BSS**

Tal como se detalla en [5], estos sistemas, con el incremento de la demanda, presentan distintos problemas:

- **Localización de las estaciones:** a la hora de diseñar el sistema es importante considerar los puntos principales donde establecer las estaciones de parking de bicicletas.
- **Aumento de la demanda:** es importante poder estimar el incremento del servicio en función del tiempo, para poder aumentar el número de estaciones del sistema.
- **Balanceo de las estaciones:** esta es una de las principales causas de déficit de los BSS, ya que es necesario tener un sistema que permita proveer del número de bicicletas que son necesarias en cada momento a cada estación. Para solucionar este problema, se proponen dos opciones: o bien, **balanceo estático** - redistribución de bicicletas por la noche, cuando hay menor demanda, una única vez al día; o bien, **balanceo continuo** - redistribución de bicicletas durante el funcionamiento del servicio, teniendo en cuenta la demanda [6].
- **Coste del sistema de balanceo:** la solución de balanceo continuo puede ocasionar un sobre coste del sistema, que tendremos que considerar a la hora de la implementación.

### 2.1.2. Soluciones desde el enfoque de análisis de datos

Para poder mejorar los problemas que se plantean en la sección 2.1.1, la mayoría de los artículos de la literatura se basan en la predicción de los trayectos que realizarán los usuarios del sistema. Existen distintas técnicas de análisis de datos empleadas en la literatura, que se proponen como solución. A continuación se listan algunas de ellas:

- **Auto-Regressive and Moving-Average (ARMA):** este método suele aplicarse a datos históricos sobre series temporales para poder predecir los datos a futuro. En el contexto de un BSS permite definir el número de bicicletas de una estación en función del tiempo a partir de los datos históricos del sistema [7].
- **Random Forest (RF):** al igual que la técnica anterior, esta técnica de *Machine Learning* nos permitirá estimar las bicicletas que habrá en una estación en un momento determinado a partir de una serie de variables. En este caso, es importante definir las variables que seleccionaremos para desarrollar el algoritmo, ya que en función de los distintos árboles que se vayan generando, el algoritmo tomará una decisión u otra. Ocurre lo mismo con los hiperparámetros: profundidad, número mínimo, etc. [6]
- **Multiple Additive Regression Tree (MART):** en MART, se construyen múltiples árboles de regresión de forma aditiva. Cada árbol se ajusta secuencialmente al residuo del modelo anterior, de modo que cada árbol se enfoca en modelar la parte no explicada de la respuesta [8].

- **Redes Neuronales Recurrentes - GRU, LSTM:** métodos eficaces cuando se trata de modelar secuencias de datos temporales. Las redes *Gated Recurrent Unit* (GRU) son redes neuronales recurrentes que ayudan a modelar y procesar secuencias de datos permitiendo retener la información relevante en el tiempo. Estas redes, son una variante de las *Long Short-Term Memory* (LSTM), las cuales tiene una estructura más compleja, con tres tipos de puertas, permitiendo la retención de la información a largo plazo y sobre conjuntos de datos más grandes. [6]
- **Regresión:** puede aplicarse para la predicción de la demanda, estudiar la influencia de otros factores sobre el uso del sistema (meteorológicos, socioeconómicos, etc.), optimizar la ubicación de las estaciones, etc.
- **Clusterización:** en el caso de la clusterización, en función de distintas variables a analizar se pueden establecer los grupos de estaciones, para poder mejorar el sistema tomando las medidas más adecuadas para cada uno de los grupos, por ejemplo.

## 2.2. Literatura preliminar

A continuación se listan artículos y proyectos que abordan una temática similar, con enfoques más o menos parecidos. Se incluye un pequeño resumen para cada uno de ellos.

- **Análisis y mejora de un sistema de bicicletas compartidas para el balanceo de estaciones** [6]: en este proyecto de final de máster se trabaja con datos de un sistema de bicicletas compartidas, en este caso, de Madrid, con el objetivo de desarrollar un modelo que permita estimar el aumento del número de bicicletas en las estaciones para suprir la demanda. No se contempla ni la clusterización ni la optimización de la ruta de reparto.
- **Mobility Modeling and Data-Driven Closed-Loop Prediction in Bike-Sharing Systems** [7]: en este artículo se realiza un estudio sobre distintos sistemas de bicicletas compartidas para intentar predecir los viajes de los usuarios. Para abordarlo, se realiza un modelo de probabilidad espacio-temporal mediante *Random-forest*. Un inconveniente de este proyecto, es, que debido a la escasez de datos, cambios ambientales o una aproximación incorrecta, el modelo debe actualizarse periódicamente. Además, desde la perspectiva de los profesionales, múltiples ejecuciones de simulación y la recopilación de datos en línea suelen conllevar sobrecarga de cálculo y comunicación.
- **Bycicle-Sharing System Analysis and Trip Prediction** [9]: en este artículo, de forma similar al anterior, se realiza un análisis de los datos de las estaciones de Chicago en

Junio de 2013. En este caso el volumen de los datos es mucho menor que el anterior, al tratarse de datos de 75 estaciones. Para la predicción de los viajes, se emplea la técnica MART.

- **Estudio y predicción del estado de las estaciones de un sistema de bicicletas compartido [10]:** se trata de un proyecto final del máster de ciencia de datos de la UOC. En el caso de este proyecto, a diferencia de los dos artículos anteriores, no se pretende estimar la ruta que seguirá un usuario del sistema, si no que se pretende desarrollar un modelo que estime el estado de las estaciones y las entradas/salidas de bicicletas que habrá en cada estación. Esta orientación se asemeja más al objetivo de este proyecto, pero sin embargo, no se plantea el desarrollo de una ruta optimizada de reparto entre las estaciones.
- **Estudio de la aplicación de algoritmos de enruteado al balanceo de vehículos en sistemas de compartición de bicicletas [11]:** en este Trabajo Final de Grado (TFG) se analizan también los datos del BSS de Valencia, en los años 2014-2015. El alumno propone una optimización de la ruta de reparto de bicicletas para 12 estaciones concretas del sistema, proporcionando una visión similar a la que se plantea en este proyecto. Sin embargo, no realiza pruebas con distintos modelos para obtener el mejor, ya que el estudio lo realiza mediante el paquete *OR Tools*,<sup>1</sup> más orientada a la perspectiva de la Ingeniería Industrial. Tampoco realiza combinación con otras fuentes de datos.
- **A modeling framework for the dynamic management of free-floating bike-sharing systems [13]:** a diferencia del resto de artículos, en este no se trabaja directamente sobre el conjunto de datos de un BSS. En este artículo, basándose en un algoritmo de clusterización, se pretender solventar el problema de la falta de bicicletas o de estacionamiento, mediante un sistema sin parking fijo, es decir, con bicicletas que pueden ser aparcadas en cualquier lugar. Se trata de una problemática distinta a la contemplada en nuestro proyecto.
- **Bike sharing systems: Solving the static rebalancing problem [14]:** por último, profundizando más en la literatura, en este artículo se explica como motivación de un sistema BSS, el problema SVOCPDP: *Single Vehicle One-commodity Capacitated Pickup and Delivery Problem*, empleado como una solución a los problemas de enruteamiento.

---

<sup>1</sup>**OR Tools:** herramienta diseñada para abordar problemas de enruteamiento de vehículos, flujos, programación de números enteros y lineales y programación de restricciones [12].

## 2.3. Comparativa del estado del arte

En la tabla 2.1, pueden observarse las distintas características de los proyectos contemplados en el estado de la cuestión. La última propuesta, sombreada en gris, representa el proyecto en cuestión.

Proyecto/Aspectos	Big-Data	Tiempo real	Optimización de la ruta de reparto	Granularidad de los datos	Dataset	Estaciones	Fecha de los datos	Combinación de datos	Técnicas
<b>Análisis y mejora de un sistema de bicicletas compartidas para el balanceo de estaciones</b>	NO	NO	NO	Horaria	BiciMad	215	Enero 2018 - Diciembre 2019	NO	Redes Neuronales Recurrentes (GRU, LSTM), RF, Regresión
<b>Mobility Modeling and Data-Driven Closed-Loop Prediction in Bike-Sharing Systems</b>	SI	SI	NO	Horaria	Hangzhou Bike-Sharing-System	3390	junio 2015	SI. Meteorológicos	ARMA, RF, PFM, PD
<b>Bicycle-Sharing System Analysis and Trip Prediction</b>	NO	SI	NO	Viaje	Divvy BSS (Chicago)	75	junio 2013	NO	MART
<b>Estudio y predicción del estado de las estaciones de un sistema de bicicletas compartido</b>	SI	NO	NO	Horaria	Oslo Bysykkel (Urban Infrastructure Partner)	301	Julio 2020 - Junio 2022	SI. Meteorológicos	Redes Neuronales Recurrentes (GRU, LSTM), RF, Silverkite, Prophet
<b>Estudio de la aplicación de algoritmos de enrutado al balanceo de vehículos en sistemas de compartición de bicicletas</b>	NO	NO	SI	Horaria	Valenbisi	12	2014, 2015	NO	OR Tools
<b>A modeling framework for the dynamic management of free-floating bike-sharing systems</b>	NO	NO	SI	A petición	-	-	-	NO	Clusterización
<b>Bike sharing systems: Solving the static rebalancing problem</b>	SI	SI	SI	-	-	-	-	-	SVOCPDP
<b>Optimización del sistema de bicicletas compartidas en la ciudad de Valencia.</b>	SI	SI	SI	Minuto - Horaria	Valenbisi	276	2019 - 2022	SI. Meteorológicos. Socioeconómicos. Geográficos	Random Forest Decision Trees. Regresión. Optimización: Dijkstra.

Cuadro 2.1: Tabla comparativa de estudios relacionados.

Fuente: elaboración propia

En la literatura, podemos encontrar distintos estudios que se enfrentan a la problemática de balanceo de los BSS desde puntos de vista táctico o estratégico. Algunos estudios plantean el aumento de estaciones en función de la demanda, otros plantean una red de transporte en base a las trayectorias de los usuarios, bien sobre datos reales, bien sobre datos ficticios. En el caso de este proyecto se trabajará con datos reales, con el objetivo de generar una ruta de reparto óptima entre las diversas estaciones que presentan problemas de ocupación/demanda, para facilitar el trabajo del camión que regula la ocupación de las estaciones. Además, la mayoría de los estudios no usan otras fuentes de datos y las que lo hacen, únicamente se trata de datos meteorológicos. Se intentará llevar a cabo un análisis de datos socioeconómicos, para valorar la opción de instalar almacenes de bicicletas en zonas estratégicas y con menor coste por metro cuadrado.

## 2.4. Planteamiento del problema

Tal como se ha comentado en la sección 1.2, este proyecto se plantea con el objetivo de mejorar y optimizar la ruta de reparto del camión que abastece las estaciones del sistema de Valenbisi. Para hacer esto posible, no solo hay que plantear un algoritmo de optimización de ruta, si no que tenemos que estimar cuándo habrá necesidad de balanceo en una estación determinada, bien sea para aumentar el número de bicicletas o bien sea para reducir el número de bicicletas para que haya bornetas libres. Por ese motivo, no solo nos vamos a centrar en la optimización, si no que también tendremos que desarrollar un modelo de estimación que nos diga cuando una estación necesitará ser balanceada.

Para lograr este objetivo las distintas tareas que se desarrollarán serán las siguientes:

- **Preparación de los datos:** en esta parte vamos a estructurar los datos que necesitamos para conseguir el objetivo. En los apartados 3.2.2 y 3.2.3 se explicará el procesamiento de los datos históricos del año 2021 del sistema de Valenbisi. Además en el apartado 3.2.1 se añadirán datos de otras fuentes externas para completar los datos y añadir mayor variabilidad al conjunto de datos. Una vez tenemos todos los datos procesados, será necesario realizar las tareas de *feature engineering*: evaluar qué datos nos aportan valor, si es necesario hacer una selección de características, si queremos reducir la dimensionalidad, añadir nuevas variables, etc.
- **Análisis descriptivo y calidad de los datos:** en esta parte del proyecto, vamos a analizar la calidad de los datos que disponemos, para ello vamos a hacer un estudio de la distribución de los datos y en caso de tener valores *outliers* o nulos, valoraremos la forma de imputarlos. Además vamos a realizar un análisis de las estaciones que tienen alta tasa de ocupación o baja tasa de disponibilidad. Así, podremos evaluar cuáles son las estaciones que mayor valor aportan al análisis. Se tendrá en cuenta además la ubicación de las estaciones, intentando hacer un filtrado de los distintos barrios de la ciudad y tomando como referencia estaciones en barrios de distinta categoría: residencial, negocios, ocio, estudiantes, etc.
- **Reducción de la dimensionalidad del problema:** en base al apartado anterior vamos a realizar una reducción sobre la dimensionalidad de los datos en cuanto al número de registros: reduciremos el número de estaciones y por ende, el número de registros. Además en este apartado se combinarán los datos que tenemos en una única fuente de datos.
- **Desarrollo de modelos de predicción:** una vez hemos procesado todos los datos con las estaciones seleccionadas para el estudio, y hemos juntado todos los datos en una

única fuente de datos, desarrollaremos los distintos modelos de predicción para poder predecir qué estaciones y cuándo necesitarán un balanceo. Añadiremos un subapartado con las conclusiones de los modelos que mejor se ajustan a nuestros datos, evaluando los parámetros resultantes de cada modelo.

- **Optimización de la ruta de reparto:** una vez hemos obtenido las estaciones que necesitan el balanceo, y los horarios de mayor necesidad, haremos un ranking e intentaremos desarrollar un modelo de optimización para la ruta de reparto entre las estaciones del estudio.
- **Conclusiones y trabajos futuros:** con este apartado finalizaremos el proyecto. Añadiremos las conclusiones sobre el algoritmo final sobre la optimización de la ruta de las estaciones y posibles mejoras a futuro.

# Capítulo 3

## Procesamiento de los datos

En esta sección vamos a listar las distintas tareas que se van a llevar a cabo para la resolución del problema planteado como objetivo del proyecto: optimización de una ruta de reparto para el balanceo entre estaciones de un sistema de bicicletas compartidas. Para abordar esta sección vamos a dividirla en las siguientes partes:

- **Tecnologías empleadas:** se realizará un listado con las tecnologías empleadas, tanto para el desarrollo del proyecto, como para el desarrollo analítico.
- **Procesamiento de los datos:** en esta parte se hará una explicación del proceso ETL (*Extract, Transform and Load*) de las distintas fuentes de datos. Se explicará cómo se han extraído los datos de las distintas fuentes, los posibles problemas que han surgido y las soluciones planteadas, y se listarán los distintos atributos de los datos con los que trabajaremos.
- **Planteamiento del problema:** por último se planteará el problema desde el punto de vista analítico a partir de los datos resultantes.

### 3.1. Tecnologías empleadas

A continuación se van a listar las tecnologías empleadas para el desarrollo del proyecto. Primero se hablará de las tecnologías generales. Posteriormente, de forma más detallada, se explicarán las distintas librerías/paquetes que se han empleado en los programas generales para la parte correspondiente al análisis.

### 3.1.1. Tecnologías de desarrollo

#### LaTeX

Se trata de una herramienta que sirve para dar formato a artículos académicos con una estructura definida por el propio usuario, de forma que es sencillo definir distintas secciones o subsecciones y generar de forma automática partes del documento como los índices o las referencias. Es especialmente útil para fórmulas matemáticas ya que permite una edición más sencilla que otro tipo de editores, además de la multitud de formatos que ofrece y de tratarse de una herramienta multiplataforma que se puede emplear en distintos sistemas operativos [15].

#### TeXstudio

Se trata de un entorno de desarrollo que permite la creación de documentos en LaTex de forma sencilla y con una interfaz intuitiva: ofrece resaltado en la sintaxis, autocompletado y ayuda contextual, organización de documentos y gestión del proyecto, gestión bibliográfica, y además es personalizable en función de las necesidades de cada usuario [15].

#### Python

Es un lenguaje de programación de código abierto, altamente legible y compatible con multitud de librerías estadísticas y de análisis de datos, que hace que sea una de las opciones preferidas en proyectos de ciencia de datos, razón por la que ha sido elegido como el programa de desarrollo de este proyecto. Algunas de las características principales de Python son: sintaxis legible; multiplataforma, permitiendo su uso en distintos sistemas operativos; alta compatibilidad con diversas librerías que permiten realizar tareas sencillas como la entrada/salida de datos hasta tareas más complejas como el desarrollo web [8].

#### Anaconda Navigator

Es una interfaz gráfica de usuario que incluye Anaconda, plataforma popular entre científicos de datos para el manejo de paquetes en Python. Permite crear entornos virtuales de modo que hace sencillo el manejo de distintos proyectos y ofrece acceso a herramientas de desarrollo como *Jupyter Notebooks* [8].

#### Jupyter Notebooks

Se trata de una herramienta de código abierto donde se puede combinar texto con código y gráficos. Es altamente útil en tareas de ciencia de datos ya que permite explicar los resultados en el mismo documento sobre el que se ejecuta el código, permitiendo así una mejor interpretación de los datos y proporcionando una alta flexibilidad [8].

### Hadoop Distributed File System

Se trata de un sistema de archivos distribuidos diseñado para almacenar grandes conjuntos de datos repartidos en distintos clústeres de computadoras. Es un sistema escalable y tolerante a fallos debido a la replicación de los datos de forma automática en distintos nodos del sistema [8].

### Apache Spark

Apache Spark consiste en un lenguaje de programación para gran procesamiento de datos (*Big Data*) que es integrable con otros lenguajes de programación, como Python. Se caracteriza por el procesamiento en memoria, permitiendo mayor rapidez que otros sistemas Big Data. Proporciona APIs para trabajar con datos que permiten realizar diversas acciones como la agregación o el mapeo y es integrable con Hadoop [8]. En este proyecto se ha usado Spark para el procesamiento de los datos de todas las estaciones mediante la asignación de un servidor en Cloudera, que permite el procesado de los cerca de 200 millones de registros con los que se trabajará en una primera instancia.

## 3.1.2. Tecnologías de análisis

A continuación se listarán las librerías más relevantes que se han importado en el código y que se consideren imprescindibles para el desarrollo del proyecto.

### Pandas

Es una librería de código abierto de Python, que permite modular datos tabulares con los que trabajar de una forma más sencilla. La estructura de datos principal de esta librería es el *Dataframe*, similar a una tabla de dos dimensiones. Pandas es especialmente útil para tareas de tratamiento de datos, estadística descriptiva, agrupación y análisis de series temporales [8].

### Seaborn

Es una librería de visualización de datos, de forma que con poco código se pueden generar visualizaciones avanzadas que permiten estudiar el patrón de los datos. Tiene una buena integración con pandas, de modo que permite crear visualizaciones a partir de *dataframes* [8].

### Requests

Permite interactuar con servicios web y APIs para mandar solicitudes y obtener datos de una página web. Además, incorpora métodos que permiten acceder a los distintos datos de respuesta de una API [8].

### **Pyplot**

Esta librería nos servirá para crear gráficos en 2D, de modo que de forma sencilla podremos obtener gráficos de barras, histogramas, gráficos de dispersión, etc. Además podremos personalizar los gráficos con las distintas funciones de la librería, pudiendo integrarlos en otros programas o simplemente visualizarlos [8].

### **Pyspark**

Esta librería nos permite acceder al entorno de Apache Spark y usar las funcionalidades que este lenguaje de procesamiento masivo de datos nos ofrece [8].

### **Folium**

Es una biblioteca de Python que se utiliza para visualizar datos geoespaciales e interactuar con mapas web. Proporciona una interfaz fácil de usar para crear mapas interactivos utilizando la biblioteca Leaflet de JavaScript. Folium permite superponer diferentes capas en el mapa, incluidos marcadores, polígonos, líneas y datos rasterizados [8].

### **CodeCarbon**

CodeCarbon es una biblioteca de Python, de código abierto desarrollada para estimar las emisiones de dióxido de carbono (CO<sub>2</sub>) asociadas con la ejecución de programas en el hardware en el que se ejecutan (macOS en el caso de este proyecto). Utiliza un enfoque basado en el consumo de energía eléctrica del hardware, teniendo en cuenta la intensidad de carbono de la electricidad consumida, que se calcula como un promedio ponderado de las emisiones de las diferentes fuentes de energía utilizadas para generar electricidad. Esto permite a los usuarios estimar el impacto ambiental de sus actividades informáticas y fomentar prácticas más sostenibles en el desarrollo de software [16].

## **3.2. Conjuntos de datos**

Una vez hemos comentado las herramientas que hemos usado para el tratamiento de los datos, pasamos a detallar cuáles son los datos que tenemos y las distintas transformaciones a realizar sobre los mismos. Para ello vamos a hacer referencia a los distintos *Notebooks* que se han creado en *Jupyter*: uno de ellos para el tratamiento de los datos de las fuentes externas de Valenbisi, otro para el tratamiento de los datos históricos de Valenbisi y el último para el procesamiento con Spark de datos masivos.

### 3.2.1. Tratamiento de datos de fuentes externas a Valenbisi

Tal como se ha comentado en el capítulo 2, en este proyecto hemos decidido combinar los datos que tenemos del BSS junto con otro tipo de datos. El código referente a esta sección se encuentra en el notebook **ETL\_other\_data.ipynb** del repositorio [Git](#). En este notebook se realiza la extracción de datos que se consideran relevantes en el impacto del uso del sistema de Valenbisi. Hemos considerado los siguientes datos:

#### Datos meteorológicos y atmosféricos

Los datos meteorológicos han sido obtenidos del portal de [datos abiertos del Ayuntamiento de Valencia](#), que permite descargarlos en distintos formatos. En este caso los hemos descargado en formato json y se han seleccionado las variables que se consideraban más relevantes:

- Fecha: formato dd/mm/yyyy
- Hora: formato hh:mm:ss
- Velocidad del viento: velocidad media en km/h
- Temperatura: temperatura media en °C
- Humedad relativa: en %
- Precipitación: precipitación de lluvia o nieve en mm
- Velocidad máxima del viento: velocidad máxima en km/h
- Radiación: en vatios por metro cuadrado ( $w/m^2$ )
- NO<sub>2</sub>: óxido de nitrógeno que puede ocasionar fatiga ( $\mu g/m^3$ )
- O<sub>3</sub>: ozono que puede ocasionar irritación de las vías respiratorias. No se especifica la unidad de medida
- CO: emisiones de los vehículos que en altas concentraciones puede afectar a la visibilidad ( $mg/m^3$ )

De todas las estaciones meteorológicas que recogen los datos, se ha seleccionado la estación del centro de la ciudad como la estación de referencia por su ubicación más céntrica que el resto de estaciones. Además, este conjunto de datos únicamente cuenta con registros de los años 2021 y 2022.

Para ver la calidad de los datos se ha realizado un mapa en escala de grises de las distintas variables, ya que se ha observado una alta concentración de valores nulos en el año 2022. Como se puede observar en la figura 3.1, en las variables meteorológicas: humedad relativa, precipitación, radiación, temperatura, velocidad del viento y velocidad máxima del viento, todos o gran parte de los registros horarios de 2022, son valores nulos. Esto puede deberse a la interrupción

del proceso de mediciones de estos valores durante el año 2022. Para las variables atmosféricas CO y O3 podemos ver que las mediciones son dispares por meses en ambos años, y para la variable NO2, podemos ver que los datos nulos se distribuyen equitativamente en cada mes, siendo diciembre de 2021 el mes con mayor concentración de valores nulos.

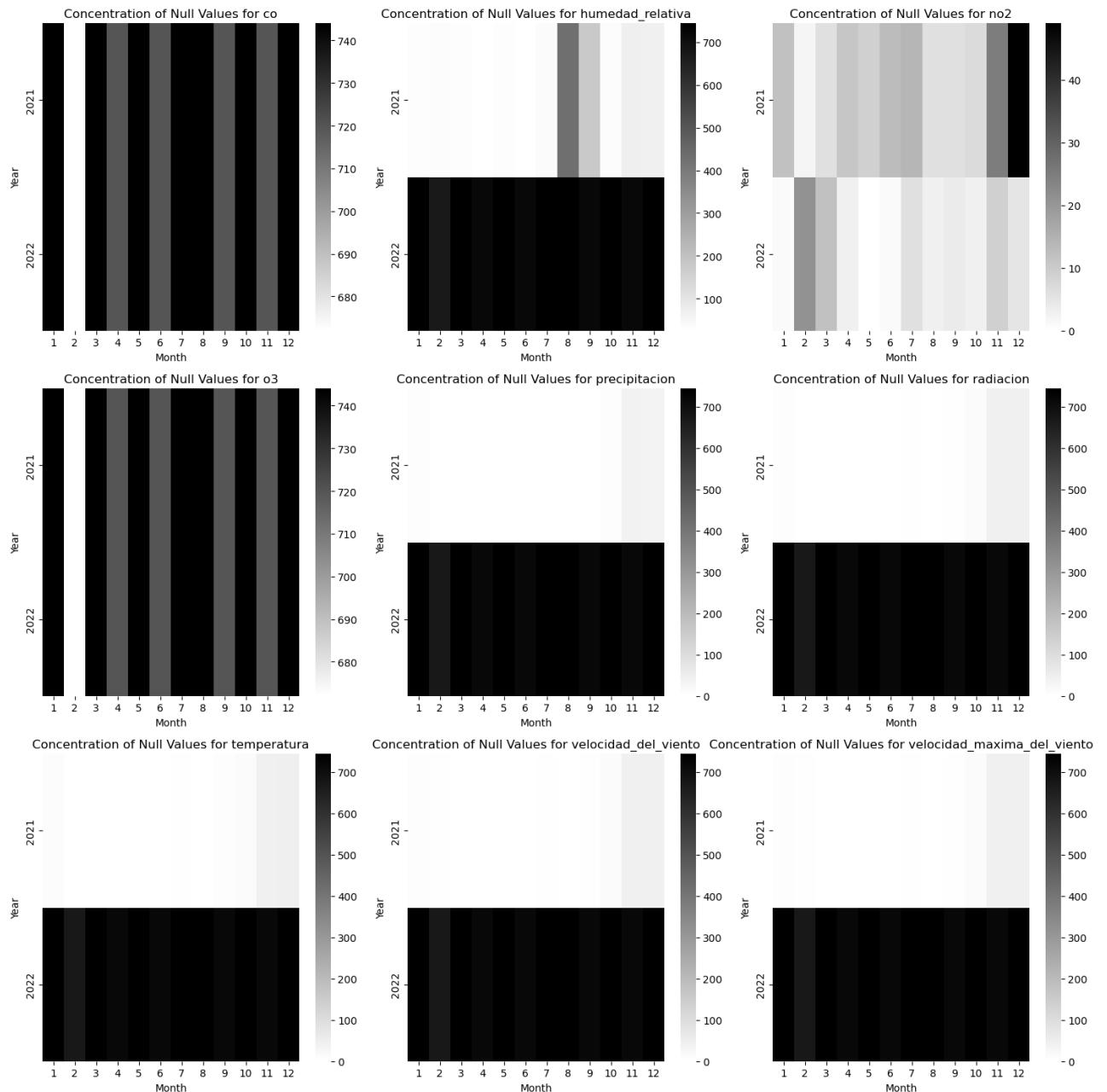


Figura 3.1: Mapa de concentración de valores nulos sobre datos atmosféricos

Fuente: elaboración propia. Notebook: ETL\_other\_data.ipynb

Ante estos resultados, y dado el gran volumen de datos de los que disponemos de las estaciones de Valenbisi (4 años con una granularidad de minuto para 276 estaciones, suponen unos 580 millones de registros apróx.), podemos concluir que los datos con los que trabajaremos serán los correspondientes a 2021 para el histórico de las estaciones y para los datos atmosféricos.

### Datos geográficos de las estaciones

Para obtener los datos de localización de las estaciones, haremos una llamada a la API del sistema de **JCDecaux** y filtraremos para obtener los siguientes datos:

- ID de la estación: número entero en el intervalo [1,276]
- Dirección de la estación: p. ej. C/GUILLEM DE CASTRO esquina con C/NA JORDANA
- Coordenadas geográficas. Se descargan conjuntas, las separamos en latitud y longitud; p. ej. 'lat': 39.48001, 'lng': -0.38302
- Bornetas totales de la estación: número entero

### Datos económicos de los barrios de Valencia

Los datos económicos que se extraen hacen referencia al precio medio del metro cuadrado en los distintos barrios de la ciudad. Para ello, empleamos el mismo portal del Ayuntamiento de Valencia, que dispone de un conjunto de datos de la web de Fotocasa. Estos datos se van a combinar con los datos geográficos de las estaciones. Para abordar el problema, se sigue la siguiente estrategia:

- Extraer los datos en formato json con las coordenadas de los distintos barrios y su forma geométrica (delimitaciones geográficas sobre el mapa).
- Comprobar en qué forma geométrica del fichero json de Fotocasa se encuentra cada estación de Valenbisi.
- Extraer las coordenadas geográficas. Se descargan conjuntas, las separamos en latitud y longitud.
- Añadir la información del barrio y del precio por metro cuadrado en el año 2022 (año con el dato más actualizado en la página del ayuntamiento).

### Datos de zonas verdes de la ciudad

Siguiendo una estrategia similar a la de los datos económicos, queremos incorporar al dataset con las coordenadas de las estaciones información referente a las zonas verdes de la ciudad.

Para ello, lo que vamos a hacer es descargar los distintos polígonos geométricos con zonas verdes de la ciudad (datos proporcionados por el Ayuntamiento de valencia), y veremos cuál es la zona verde más cercana a cada estación, para añadir así los metros cuadrados de zona verde.

### 3.2.2. Tratamiento de datos históricos de Valenbisi

Los datos de los que disponemos del sistema de Valenbisi son de los últimos 4 años y han sido proporcionados en carpetas comprimidas siguiendo la siguiente estructura:

- Año - Mes - Día - Estación (txt)
  - Fecha y hora, Bicicletas disponibles, Bornetas disponibles

Una vez los datos han sido descomprimidos, vamos a realizar una transformación sobre los mismos para poder obtener una base de datos sólida sobre la que trabajar. La principal transformación que vamos a realizar será centralizar todos los registros separados en ficheros por estaciones en la jerarquía de carpetas Año/Mes/Día/Estación, añadiendo la información de la estación a cada registro con el siguiente formato:

- ID de la estación
- Fecha y hora
- Bicicletas disponibles
- Bornetas disponibles

Teniendo en cuenta que solo vamos a procesar los datos del año 2021, y que la granularidad de los datos es del minuto, contamos con aproximadamente medio millón de registros para cada estación. Dado que tenemos los datos de 276 estaciones, eso implica que el procesamiento de los datos necesita de una herramienta de *Big Data* para poder procesar todos los registros. Sin embargo, la granularidad que queremos es más bien horaria (la ocupación de las estaciones no cambiará mucho cada minuto), por lo que para poder procesar los datos con Python, en una primera instancia se decide recorrer los registros de los ficheros de texto cada 60 líneas, tomando así como referencia un único valor de cada hora. Para procesar todos los registros, no tenemos los recursos locales suficientes, por lo que esto lo haremos en un servidor externo.

### 3.2.3. Tratamiento de datos históricos de Valenbisi con Spark

Como se ha comentado en la sección [3.2.2](#), dado que nos hemos encontrado con un problema de procesamiento por la gran cantidad de registros, vamos a desarrollar el proceso en un

entorno con Spark, para así poder procesar los valores medios por estación/día/hora. Se adjunta brevemente el pseudocódigo en el algoritmo 1; se puede observar el código detallado en el notebook **ETL\_Spark.ipynb**.

En este algoritmo, lo primero que haremos será obtener los directorios de los datos almacenados y descomprimirlos. Después se procesará cada línea de cada fichero dándole el formato necesario para su tratamiento. Mediante una estructura de datos distribuidos en Spark (RDD), procesaremos los valores medios de bicicletas y parking por horas, y la suma de los viajes totales.

---

**Algorithm 1** Pseudocódigo del algoritmo *process\_data\_Spark*

**Entrada:** *data\_years*: Lista de años

```
spark ← SparkSession.builder.appName("DataProcessing").getOrCreate()
start_directories ← get_directories(data_years)
output ← data_descomp/data_avg.txt

def process_line(line, station_id):
    data = line.strip().split(',')
    ts_str = data[0]
    bikes, parking = float(data[1]), float(data[2])
    ts = datetime.strptime(ts_str, '%Y/%m/%d %H:%M:%S')
    return (station_id, ts, bikes, parking)

rdd ← spark.sparkContext.parallelize(start_directories)
    .flatMap(for file in os.listdir(directory))
    .filter(lambda station: 'checkpoints' not in station[1])
    .flatMap(lambda station: for line in station)

rdd ← rdd.map(lambda x: process_line(x[1], x[0]))
df ← spark.createDataFrame(rdd, ['station_id', 'timestamp', 'bikes', 'parking'])
df ← df.withColumn('date', date_format('timestamp', 'yyyy-MM-dd'))
    .withColumn('hour', date_format('timestamp', 'HH'))

avg_df ← df.groupBy("station_id", "date", "hour").agg(
    avg("bikes").alias(avg_bikes),
    avg("parking").alias(avg_parking),
    max("total_trips").alias(max_trips))

avg_df.write.mode(overwrite).csv(output, header=False)
spark.stop()
```

---

# Capítulo 4

## Analítica descriptiva y calidad de los datos

En el capítulo anterior se han procesado los datos con los que vamos a trabajar para definir el modelo predictivo. Para garantizar la calidad de los datos procesados, en ese capítulo vamos a realizar un estudio descriptivo de los mismos. Además, finalizaremos con el resultado del conjunto de datos final, que será el que usaremos en el siguiente capítulo para definir el mejor modelo de predicción. Seguiremos el siguiente orden:

- **Datos meteorológicos:** análisis de los datos, valores nulos y *outliers*<sup>1</sup>, imputación de estos valores. Análisis de correlación e impacto entre las variables numéricas.
- **Datos de las estaciones: geográficos, económicos y zonas verdes:** realizaremos un estudio de los posibles *outliers* de las variables numéricas. Imputación de outliers de las bornetas de las paradas y evaluación de zonas verdes de grandes dimensiones.
- **Datos de los movimientos en las estaciones:** unión de los datos de valores medios con los datos de referencia de cada hora. Análisis de estaciones con mayor impacto para el proyecto: estaciones con mayor número de viajes, mayor tasa de ocupación, menor tasa de disponibilidad de bicicletas y estaciones con la mayor desviación con respecto del valor medio.

### 4.1. Datos meteorológicos

#### 4.1.1. Análisis de los datos

Para estudiar la calidad de los datos meteorológicos, tal como se puede observar en el notebook `data_analysis.ipynb`, vamos a realizar un estudio de valores nulos y *outliers*. Podemos

---

<sup>1</sup>**outliers:** en estadística, tales como muestras estratificadas, un valor atípico (en inglés outlier) es una observación que es numéricamente distante del resto de los datos. [17].

observar los resultados del estudio en la siguiente tabla:

	Valores Nulos		Valores Outlier	
	Cantidad	%	Cantidad	%
velocidad_del_viento	128	1.46 %	300	3.42 %
temperatura	130	1.48 %	0	0.00 %
humedad_relativa	1005	11.47 %	0	0.00 %
precipitacion	90	1.03 %	297	3.39 %
velocidad_maxima_del_viento	128	1.46 %	257	2.93 %
radiacion	124	1.42 %	325	3.71 %
no2	160	1.83 %	575	6.56 %
o3	8760	100.00 %	0	0.00 %
co	8760	100.00 %	0	0.00 %

Cuadro 4.1: Valores nulos y outliers de los datos meteorológicos.

Fuente: elaboración propia; notebook: data\_analysis.ipynb

### 4.1.2. Imputación de valores nulos

Con los resultados de la tabla 4.1 tomaremos las siguientes decisiones:

- Las columnas **O3** y **CO** pueden ser eliminadas ya que no contienen valores.
- Para imputar los valores nulos de las variables con porcentaje bajo (<5%) usaremos el método kNN de vecinos más cercanos, con un total de 5 vecinos. Como todas tienen porcentajes muy bajos de valores nulos, no daremos importancia al orden de imputación. Las variables a tener en cuenta en la imputación serán las siguientes:
  - Para **velocidad del viento**: temperatura, velocidad máxima del viento y fecha y hora.
  - Para **temperatura**: velocidad del viento, precipitación y radiación.
  - Para **precipitación**: fecha y hora.
  - Para **velocidad máxima del viento**: temperatura, velocidad del viento, fecha y hora.
  - Para **radiación**: NO2, temperatura, fecha y hora.
  - Para la variable **NO2** decidimos imputar los valores medios de los 23 vecinos más próximos en función de la hora, es decir, tomaremos como referencia las 23 horas más cercanas.

- La mayoría de los registros con **humedad relativa** nula se localizan en el mes de agosto. Para imputar los valores nulos de humedad relativa vamos a considerar los valores medios de humedad por mes y hora.

#### 4.1.3. Imputación de valores outliers

Para evaluar la existencia de outliers, se realiza un estudio visual de los datos numéricos mediante un diagrama de cajas, que muestra los valores que se salen de los cuantiles. Podemos visualizar el resultado en la siguiente figura 4.1.

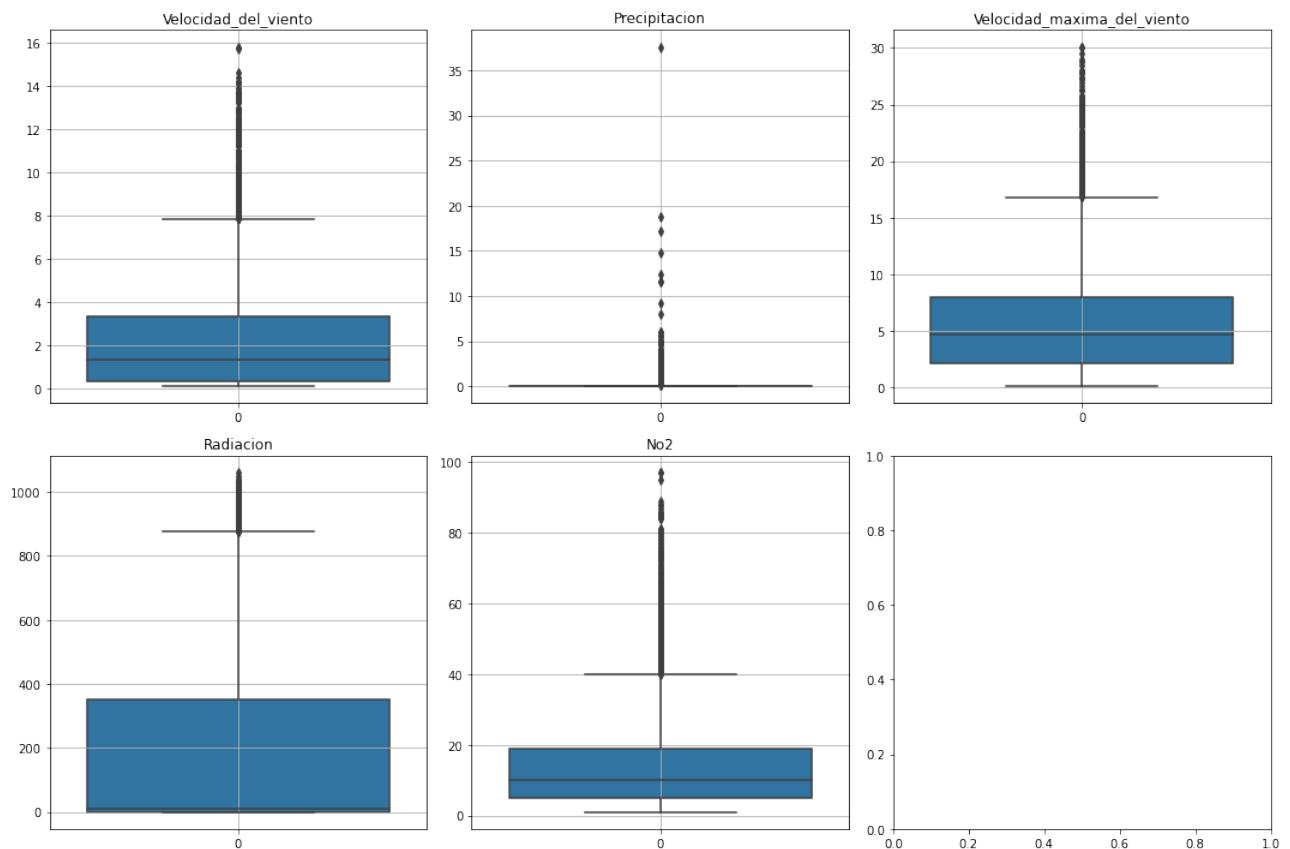


Figura 4.1: Valores outliers de los datos meteorológicos.

Fuente: elaboración propia; notebook: *data\_analysis.ipynb*

Del estudio visual en la figura 4.1, podemos deducir que las dos variables referentes al viento, son variables sin *outliers* puesto que no se salen de un rango posible de los valores de velocidad del viento. Precipitación, aparentemente tiene un valor que es bastante mayor que el resto, pero también descartaremos que sea un *outlier* porque no se trata de un valor anómalo, que además en la zona del mediterráneo podría explicarse con una dana puntual. Respecto a los valores de NO<sub>2</sub>, en la [página del ministerio de medioambiente](#) se indica que los valores medios se sitúan

en 40, siendo el límite 200. Luego también descartamos que se trate de *outliers*, al estar los valores por debajo de 100. Para la variable radiación, vamos a realizar un estudio de cuándo se producen valores superiores a 400, por hora, mes y día. Podemos ver el resultado en la siguiente figura.

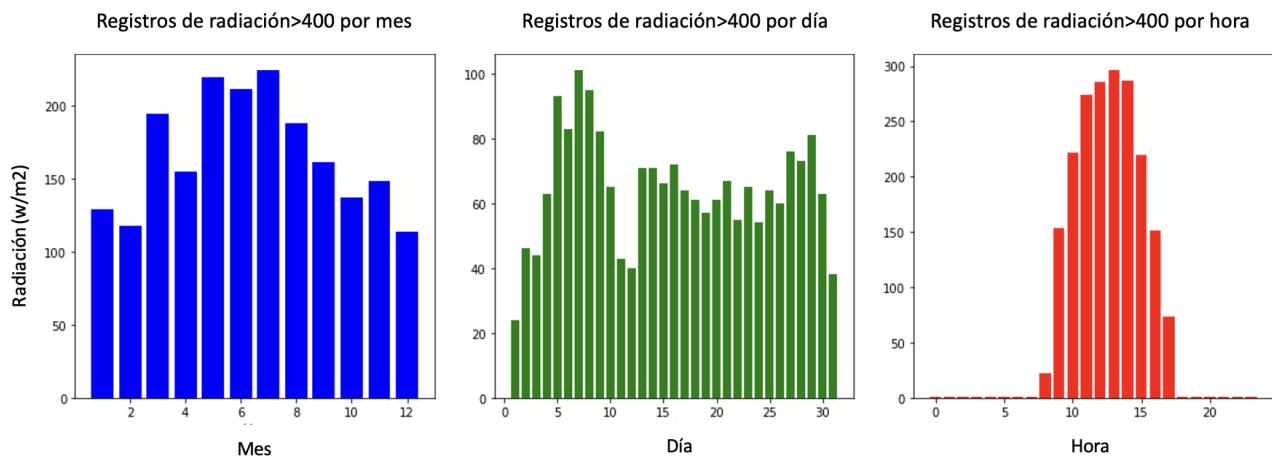


Figura 4.2: Distribución de los valores mayor de 400 de la variable radiación por mes, día y hora.

Fuente: elaboración propia; notebook: *data\_analysis.ipynb*

Observando la figura 4.2, podemos ver que los valores se distribuyen de forma normal. El único gráfico que no nos da información útil es el gráfico por día, pero debido a que es muy variable, dependiendo de la luz solar, es normal que el resultado no se distribuya de forma normal. Sin embargo los meses y horas, al ser variables estacionarias, si que nos dan información de la distribución de la radiación solar. Luego, decidimos no imputar ninguno de estos valores, al mostrar una distribución normal.

#### 4.1.4. Correlación entre variables

Para finalizar, se estudia mediante la matriz de correlación si podemos descartar algunas de las variables por estar altamente correlacionadas. Como se observa en la figura 4.3, existe una alta correlación entre la variable que mide la velocidad del viento y la variable que mide la velocidad máxima del viento, por lo que decidimos eliminar del conjunto la velocidad máxima al considerarse menos representativa.

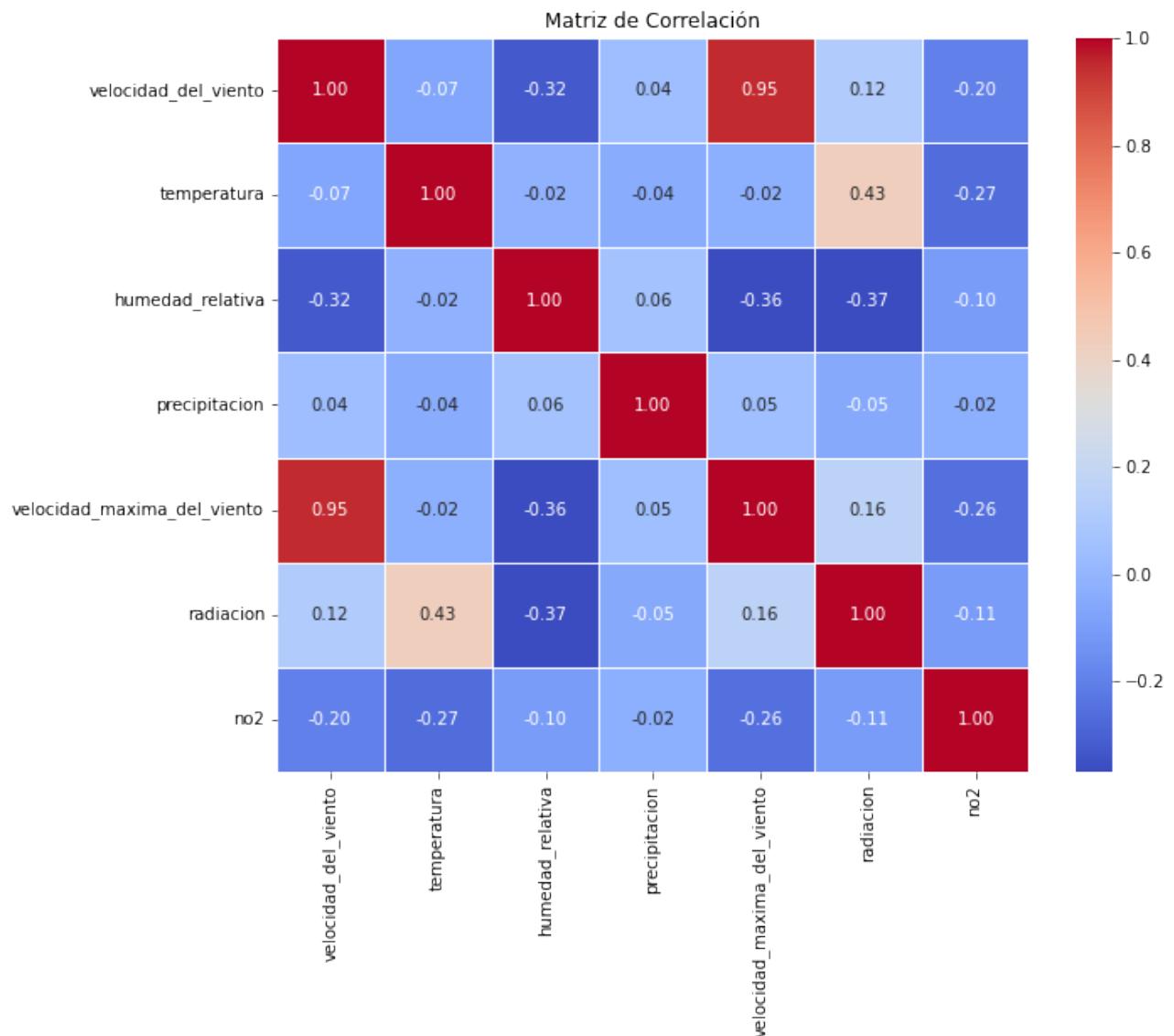


Figura 4.3: Matriz de correlación de las variables atmosféricas.

*Fuente: elaboración propia; notebook: data\_analysis.ipynb*

## 4.2. Datos de las estaciones: geográficos, económicos y zonas verdes

### 4.2.1. Análisis de valores outliers

Para realizar el estudio de posibles valores atípicos en las variables numéricas de las estaciones, podemos observar la tabla 4.2 y la figura 4.4

	Valores Outlier	Porcentaje de valores outliers
bike_stands	24	8.70 %
lat	0	0.00 %
lng	0	0.00 %
precio_2022_euros_m2	6	2.17 %
close_green_area_shape	37	13.41 %

Cuadro 4.2: Valores outliers y porcentaje de valores outliers de datos de estaciones.

Fuente: elaboración propia; notebook: data\_analysis.ipynb

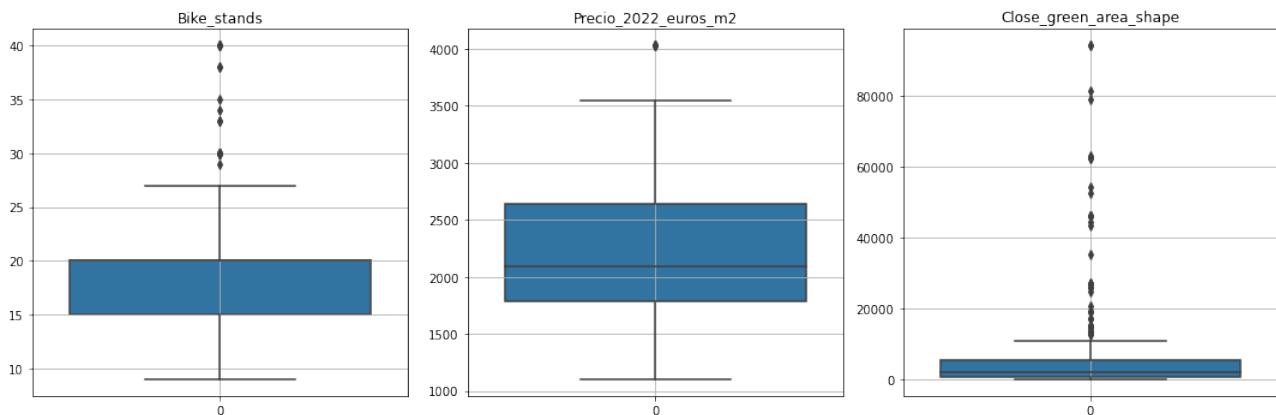


Figura 4.4: Estudios de outliers en datos geográficos.

Fuente: elaboración propia; notebook: data\_analysis.ipynb

Con los resultados anteriores, vamos a proceder a:

- **Estaciones con 40 parkings:** revisión las estaciones que tienen exactamente 40 parkings y determinar si realmente se trata de *outliers*. Es posible que exista una razón específica para esta cantidad de parkings en algunas estaciones.
- **Precio del metro cuadrado:** no se considera un *outlier* al contrastarlo con otras fuentes. Se asume que los valores proporcionados están dentro del rango esperado o son consistentes con otras fuentes de datos [18].
- **Zonas verdes:** visualizar las zonas verdes en un mapa para evaluar si son *outliers* debido a un posible error en los decimales o si se trata de zonas verdes de gran dimensión, como los tramos del río Turia. Esta visualización ayudará a determinar si las áreas verdes son representativas o si requieren una revisión más detallada.

#### 4.2.2. Evaluación de outliers en zonas verdes de grandes dimensiones

Para verificar si los valores de zonas verdes con grandes dimensiones son realmente *outliers*, vamos a visualizar el área de las zonas verdes de gran dimensión sobre un mapa para evaluar visualmente si se trata de errores en los datos (por ejemplo con las unidades de medida).



Figura 4.5: Mapa de las zonas verdes consideradas outliers.

Fuente: elaboración propia; notebook: [data\\_analysis.ipynb](#)

Se puede visualizar el mapa interactivo en el notebook [data\\_analysis.ipynb](#). De la figura 4.5, concluimos que los datos de las zonas verdes consideradas como valores atípicos, se trata de datos sin error al referirse a zonas ajardinadas de mayor dimensión (jardines del antiguo cauce del río Turia) o zonas que incluyen grupos de zonas verdes. Luego, no imputamos ninguno de estos valores.

### 4.3. Reducción de la dimensionalidad de los datos

Por último, nos queda evaluar la calidad de los datos de los viajes de las estaciones. Tenemos dos grandes conjuntos de datos: los datos que registran un único valor por hora, día y estación, y los datos que muestran los valores medios y el total de viajes, con la misma granularidad: hora, día y estación. Además, tenemos 276 estaciones, y nos gustaría tomar algunas como referencia para poder reducir la dimensión de los datos. Para ello, procederemos de la siguiente forma:

- Uniremos ambos conjuntos de datos usando como clave de unión la fecha y hora, y la estación.

- Realizaremos un estudio de las estaciones con mayor desviación del valor medio por hora.
- Analizaremos cuáles son las estaciones de referencia: mayores tasas de ocupación media, menor disponibilidad de bicicletas, y mayor número de viajes.
- Una vez hemos evaluado las métricas anteriores y visualizando sobre un mapa, con el conocimiento de la ciudad, elegiremos las 19 estaciones más representativas. El motivo de seleccionar 19 estaciones, es meramente orientativo y se fija ese número porque es aproximadamente un 7 % del total de estaciones, y consideramos que es una cantidad óptima para generalizar el sistema.

#### 4.3.1. Evaluación de los datos sobre movimientos: registros erróneos

Para evaluar la calidad de los datos del conjunto resultante, lo primero que haremos será corroborar que los datos que tenemos sobre las bicicletas y parkings de cada estación, son datos correctos. Para ello, vamos a verificar que la suma de bicicletas y parkings coincide con el campo *bike\_stands* que indica la capacidad de cada estación. Esto puede deberse a diversos motivos: aumento del numero de parkings en la estación, errores del sistema, bicicletas no reconocidas en el parking, etc. Aquellas estaciones donde el porcentaje de error sea mayor a un 20 % serán eliminadas del estudio.

Para el resto de las estaciones, aquellas con un error menor al 20 %, se realizará la siguiente imputación:

- Si la estación tiene el número de parking inferior o igual al campo *bike\_stands*, entonces modificaremos el campo *bikes* como *bike\_stands - parking*.
- Si la estación tiene el número de parking mayor a *bike\_stands* y *bikes* es menor o igual a *bike\_stands*, entonces modificaremos el campo *parking* como *bike\_stands - bikes*.
- En cualquier otro caso, eliminaremos el registro.

Podemos observar los detalles del pseudocódigo en el algoritmo 2.

---

**Algorithm 2** Corrección de registros en DataFrame de bicicletas

---

**Entrada:**  $df$ : DataFrame de bicicletas

```

for cada fila  $fila$  en  $df$  do
    if  $fila['parking'] \leq fila['bike_stands']$  then
         $fila['bikes'] \leftarrow fila['bike_stands'] - fila['parking']$ 
    else if  $fila['parking'] > fila['bike_stands']$  and  $fila['bikes'] \leq fila['bike_stands']$  then
         $fila['parking'] \leftarrow fila['bike_stands'] - fila['bikes']$ 
    else
        Eliminar la fila  $fila$  de  $df$ 
    end if
end for
return  $df$ 
```

---

#### 4.3.2. Estaciones de referencia

Para finalizar, como ya se ha comentado al inicio del capítulo, elegiremos cuatro métricas para evaluar nuestras estaciones de referencia:

- ¿Cuáles son las estaciones con mayor desviación estándar en el número de bicicletas?
- ¿Cuáles son las estaciones con mayor número de movimientos?
- ¿Cuáles son las estaciones con una mayor tasa de ocupación?
- ¿Cuáles son las estaciones con una menor disponibilidad de bicicletas?

Una vez hemos procesado los 4 conjuntos de estaciones que responden a nuestras preguntas, representamos sobre el mapa las estaciones resultantes (4.6) y vemos si se trata de estaciones dispersas en la ciudad y si se cubren los distintos tipos de barrios: negocios, residencial, ocio, académico, etc.



Figura 4.6: Mapa de las estaciones resultantes del estudio.

Fuente: elaboración propia; notebook: *data\_analysis.ipynb*

De la figura 4.6, concluimos la elección de las siguientes 19 estaciones:

- Barrios del extraradio de Valencia: 217, 240, 167
- Zona de estudiantes: 85, 92, 97, 110, 114
- Zona residencial: 79, 75, 66, 101, 50, 41 (esta última se añade por ser la estación que corresponde a mi residencia).
- Zonas turísticas y de ocio: 14, 148, 40, 28, 149

# Capítulo 5

## Análisis predictivo de los datos. Atributos, separación de conjuntos y modelos de predicción

### 5.1. Tratamiento de los atributos del conjunto de datos final

#### 5.1.1. Elección de los atributos

Tras el análisis realizado en el capítulo 4, podemos categorizar nuestros datos como:

- **Datos de las estaciones:** recogen los parámetros de estudio para las predicciones, esto es el número de bicicletas, parking y bicicletas medias por hora de cada estación del sistema.
- **Datos geolocalizados:** estos datos hacen referencia a los datos geográficos que tenemos sobre las estaciones de bicicletas, en referencia a su ubicación (latitud y longitud), a las zonas verdes cercanas o al precio por metro cuadrado del barrio donde se encuentran.
- **Datos atmosféricos:** se trata de los datos de variables atmosféricas que hacen que trabajemos con datos estacionales.

En la siguiente tabla podemos observar los atributos seleccionados, el tipo de dato y una breve descripción de cada atributo.

Fuente	Campo	Tipo de dato	Descripción
<b>Datos Estaciones</b>	station_id	int64	Identificador único de la estación
	parking_avg	float64	Promedio de espacios de aparcamiento disponibles
	total_trips	int64	Número total de viajes
	timestamp	object	Marca temporal del dato
	hora	int64	Hora del dato
	dia	int64	Día del dato
	festivo	bool	Festivo o no
<b>Datos Geográficos</b>	mes	int64	Mes del dato
	bikes	int64	Número de bicicletas disponibles
	bike_stands	int64	Número total de puestos para bicicletas
	bike_avg	float64	Valor medio de bicicletas disponibles
	barrio	object	Barrio donde se encuentra la estación
	distrito	object	Distrito donde se encuentra la estación
<b>Datos Atmosféricos</b>	lat	float64	Latitud de la estación
	lng	float64	Longitud de la estación
	precio_2022_euros_m2	float64	Precio por metro cuadrado en 2022
	close_green_area_shape	float64	Área verde cercana más próxima
	velocidad_del_viento	float64	Velocidad del viento en la ciudad
<b>Datos Atmosféricos</b>	temperatura	float64	Temperatura en la ciudad
	humedad_relativa	float64	Humedad relativa en la ciudad
	precipitacion	float64	Precipitación en la ciudad
	radiacion	float64	Radiación en la ciudad
	no2	float64	Nivel de NO2 en la ciudad

Cuadro 5.1: Tipos de datos y descripciones por categoría.

Fuente: elaboración propia.

## 5.1.2. Atributos adicionales de los datos

### Codificación de las variables categóricas

Si observamos la tabla 5.1, podemos ver que hay datos no numéricos: algunos datos de las estaciones son campos de texto, esto es el barrio o el distrito de la estación, así como el dato de festivo es un booleano. Como son datos que queremos usar en los modelos predictivos, vamos a codificarlos, ya que alguno de los modelos solo trabajan con datos numéricos. Añadimos de esta forma los siguientes campos de datos:

Fuente	Campo	Tipo de dato	Descripción
Datos Codificados	barrio_encoded	int64	Identificador de barrio
	distrito_encoded	int64	Identificador de distrito
	festivo_encoded	int64	0 si no festivo; 1 si festivo
	dia_semana	int64	Identificador del día de la semana

Cuadro 5.2: Datos codificados.

*Fuente: elaboración propia.*

## Escalado de los datos

Para evaluar las variables a escalar, realizamos un estudio de los rangos de las variables numéricas. podemos observarlo en la tabla siguiente:

Variable	Valor Mínimo	Valor Máximo
station_id	14.0	240.00
bike_stands	20.0	40.00
total_trips	0.0	560.00
hora	0.0	23.00
dia	1.0	31.00
mes	1.0	12.00
velocidad_del_viento	0.1	15.80
temperatura	0.7	36.80
humedad_relativa	13.0	100.00
precipitacion	0.0	37.60
radiacion	0.0	1060.00
no2	1.0	97.00
precio_2022_euros_m2	1602.0	4029.00
close_green_area_shape	34.7	6893.89

Cuadro 5.3: Valores mínimos y máximos de las variables

Viendo los valores de la tabla, tomamos la decisión de escalar los datos de precio\_2022\_euro\_m2 y close\_green\_area\_shape. Dado que no necesariamente siguen una distribución normal, elegimos escalarlos por sus valores mínimo y máximo, mediante la función *MinMaxScaler* de Python. Podemos verlo ver el pseudocódigo del algoritmo en [3](#).

---

**Algorithm 3** MinMaxScaler()

**Entrada:** Conjunto de datos  $X$  con  $n$  características y  $m$  muestras

**Salida:** Conjunto de datos escalado  $X'$

- 1: Inicializar un objeto MinMaxScaler()
  - 2: Ajustar el objeto MinMaxScaler() al conjunto de datos  $X$  para calcular el mínimo ( $\min_i$ ) y el máximo ( $\max_i$ ) de cada característica  $i$
  - 3: **for** cada característica  $i$  en  $X$  **do**
  - 4:    Escalar la característica  $i$  en  $X$  usando la fórmula:
  - 5:    
$$X'_i = \frac{X_i - \min_i}{\max_i - \min_i}$$
  - 6: **end for**
  - 7: **Devolver:**  $X'$
- 

Fuente: [19]

### Variable a predecir: precisa balanceo

Dado que el objetivo principal de este proyecto es predecir cuándo una estación precisará ser atendida por un camión de reparto cuando presente un problema de desbalanceo, vamos a definir un criterio de atención que se definirá de la siguiente forma:

- **Necesidad de bicicletas:** habrá una necesidad de llevar bicicletas a una estación cuando presente menos del 5 % del total de sus aparcamientos. Esta elección se debe a que todas las estaciones del estudio tienen al menos 20 aparcamientos, por lo que una cantidad de bicicletas inferior al 5 % indica que es una estación con una bicicleta o ninguna.
- **Necesidad de parking:** será necesario liberar una estación cuando presente al menos el 95 % o más de ocupación de su totalidad.

Definimos de este modo la variable *precisa\_balanceo*, que será un valor booleano y será imprescindible en los modelos de clasificación.

## 5.2. Preparación de los conjuntos de entrenamiento y testeo

A la hora de realizar la división del conjunto de datos para entrenar los distintos modelos, es muy importante tener en cuenta las siguientes cuestiones que van a ser fundamentales para desarrollar modelos no sesgados:

- Los datos que disponemos son temporales. Tenemos información de las estaciones de bicicletas para cada día y hora de 2021, y también información de las condiciones climatológicas por hora. Si los datos de un día y hora en concreto se incluyen en el conjunto de entrenamiento, se deben incluir los datos de todas las estaciones para ese mismo día y hora, para evitar tener datos climatológicos repetidos en los conjuntos de entrenamiento y testeo.
- Al incluir datos del clima, estamos hablando de datos estacionales. Es importante que se incluyan datos para el entrenamiento de todas las estaciones del año, ya que no es lo mismo la climatología en verano que en invierno. Además hay que tener en cuenta que no es lo mismo un día de fin de semana que un día de diario.
- Los datos también son espaciales. En el caso de que se incluyan, en los conjuntos de entrenamiento y test, datos de las mismas estaciones, las variables que dependen de la geoposición de las estaciones (zonas verdes, precio del metro cuadrado, latitud, longitud, barrio, distrito) también van a aparecer de forma repetida en ambos conjuntos.

Para poder lograr una división insesgada de los datos, vamos a centrarnos en las siguientes técnicas:

- **Estratificación:** queremos tener de forma equilibrada los datos de las distintas estaciones climatológicas (meses), de los días de la semana y de los festivos. Es decir queremos una estratificación temporal-estacional. Además, para evitar el sesgo en las cuestiones de localización, como tratamos con datos espaciales queremos valorar la posibilidad de elegir distintas estaciones de bicicletas entre los conjuntos de entrenamiento y de testeo.
- **Tratamiento de series temporales en modelos supervisados:** considerando que los datos son una serie temporal, es importante tener en cuenta la secuencia del tiempo en el que ocurren los hechos. Para ello, lo que haremos será ordenarlos por ocurrencia y partiremos los conjuntos con la división 80/20. Además, dado que es una serie temporal donde habrá una importante influencia de los datos de las horas previas, vamos a crear un algoritmo que añada los datos de bicicletas de cada estación en las  $n$  horas previas (según se le indique al algoritmo). Podemos observar el pseudocódigo de este algoritmo en [4](#).

**Algorithm 4** series\_to\_supervised**Entrada:** *data, features, target, n\_in, dropnan***Salida:** *agg*

- 1: Inicializar *agg* como DataFrame vacío
  - 2: **for** cada fila en *data* **do**
  - 3:   Obtener características pasadas para *target*
  - 4:   Aregar características pasadas y actuales a *agg*
  - 5: **end for**
  - 6: **if** *dropnan* **then**
  - 7:   Eliminar filas con valores nulos en *agg*
  - 8: **end if**
  - 9: **return** *agg*
- 

*Fuente: [20]*

Luego queremos aplicar estratificación sobre una serie temporal para poder asegurar que haya:

- Proporciones equilibradas de cada mes del año (resolvemos así el problema de la estacionalidad de los datos).
- Proporciones equilibradas en los festivos = *True*.
- Proporciones equilibradas en los dia\_semana.

Además, se van a definir dos tipos de conjuntos: conjuntos con estaciones comunes entre testeo y entrenamiento y conjuntos con estaciones dispares entre testeo y entrenamiento. De este modo tenemos que:

**1. Conjunto de datos que comparten estaciones entre train y test:**

- **Ventajas:**
  - Permite al modelo capturar características específicas de cada estación, lo que podría mejorar la precisión de las predicciones para esas estaciones específicas.
  - Ayuda a evaluar cómo el modelo se desempeña en datos de estaciones similares a las que ha visto durante el entrenamiento.
- **Desventajas:**
  - Existe el riesgo de sobreajuste si hay una variabilidad considerable entre las estaciones.

- La capacidad del modelo para generalizar a nuevas estaciones puede ser limitada.

**2. Conjunto de datos sin compartir estaciones entre train y test:** las estaciones del conjunto de entrenamiento, serán filtradas de modo que el 20 % de las estaciones (aprox.) no estén en entrenamiento pero sí en testeo.

■ **Ventajas:**

- Evitas que el modelo aprenda características específicas de una estación en particular, lo que podría llevar a una mejor generalización a nuevas estaciones.
- Ayuda a evaluar la capacidad del modelo para capturar patrones temporales y climáticos de manera más general.

■ **Desventajas:**

- Si las estaciones tienen características únicas que afectan significativamente a la variable objetivo (precisa\_balanceo o bikes\_avg), el modelo podría no capturar adecuadamente estas variaciones.
- La generalización del modelo a nuevas estaciones puede ser menos precisa.

En la tabla 5.4 podemos observar los detalles de los conjuntos de datos sobre los que desarrollaremos los modelos predictivos.

Dataset	Descripción	Tipo	Filas	Columnas	Estratificación	Estaciones	Horas previas
<b>Dataset Global</b>	Es el conjunto de datos obtenido del procesamiento, limpio y preparado para los modelos.	-	166.364	26	-	19	-
<b>DataSet1</b>	Es el conjunto de datos resultante de aplicar estratificación, con estaciones comunes en entrenamiento y testeo.	Entrenamiento	133.095	28	día semana, festivo, mes	19	NO
		Testeo	33.269			19	
<b>DataSet2</b>	Es el conjunto de datos resultante de aplicar estratificación, con estaciones disjuntas en entrenamiento y testeo.	Entrenamiento	131.340	28	día semana, festivo, mes	15	NO
		Testeo	35.024			4	
<b>DataSet3</b>	Es el conjunto de datos resultante de añadir datos de las horas previas, con estaciones comunes en entrenamiento y testeo.	Entrenamiento	133.000	depende de las horas previas	-	19	Sí
		Testeo	33.250			19	
<b>DataSet4</b>	Es el conjunto de datos resultante de aplicar estratificación y datos de las horas previas, con estaciones comunes en entrenamiento y testeo.	Entrenamiento	133.057	depende de las horas previas	día semana, festivo, mes	19	Sí
		Testeo	33.269			19	
<b>DataSet5</b>	Es el conjunto de datos resultante de aplicar estratificación y datos de las horas previas, con estaciones disjuntas en entrenamiento y testeo.	Entrenamiento	131.310	depende de las horas previas	día semana, festivo, mes	15	Sí
		Testeo	35.016			4	

Cuadro 5.4: Conjuntos de entrenamiento y testeo

## 5.3. Modelos predictivos

De acuerdo a la división de los conjuntos en el apartado 5.2, tenemos 5 grupos de conjuntos sobre los que realizar el entrenamiento y testeo de nuestros modelos. Recordamos que el objetivo de nuestro proyecto será determinar cuándo una estación estará desbalanceada y precisará ser atendida por un camión de reparto en una ruta optimizada. Para lograr este objetivo, podemos usar el valor de bicicletas que tendrá la estación (o en su defecto el valor medio por hora) o también nos sirve el nuevo atributo añadido a nuestros conjuntos: `precisa_balanceo`. En función de qué queramos predecir, tenemos dos grupos de modelos:

- **Modelos de regresión:** son aquellos modelos estadísticos que tratan de explicar el comportamiento de una variable dependiente continua en función de una o más variables independientes (continuas o categóricas).
- **Modelos de clasificación:** son los modelos que se usan para predecir una clase en una variable categórica dependiente, en función de una serie de características independientes. Se diferencian de los modelos de regresión en que no predicen una variable continua si no una característica.

### 5.3.1. Regresión multipolinomial

La regresión polinómica múltiple es un tipo de modelo de regresión que permite modelar la relación entre una variable dependiente y varias variables independientes mediante un polinomio de grado superior a uno. Es una extensión de la regresión lineal múltiple, donde en lugar de asumir una relación lineal entre las variables independientes y la dependiente, se asume una relación polinómica. Esto permite capturar relaciones no lineales más complejas [21].

Para una regresión polinómica de grado  $d$  con  $n$  variables independientes  $(x_1, x_2, \dots, x_n)$ , la relación puede expresarse como:

$$y = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \beta_{ii} x_i^2 + \sum_{i=1}^n \sum_{j=i+1}^n \beta_{ij} x_i x_j + \dots + \beta_{n\dots n} x_n^d + \epsilon$$

donde:

- $y$  es la variable dependiente.
- $\beta_0, \beta_i, \beta_{ii}, \beta_{ij}, \dots, \beta_{n\dots n}$  son los coeficientes del modelo.
- $x_i$  y  $x_j$  son las variables independientes.

- $\epsilon$  es el término de error.

En nuestro caso, vamos a emplear la regresión polinómica múltiple para estimar el valor de la variable continua `bikes_avg`. Para ello procedemos del siguiente modo:

- Elección de las variables independientes
- Definimos los grados del polinomio
- Elección de los conjuntos de datos de entrenamiento y testeo
- Entrenamiento del modelo
- Evaluación del modelo de acuerdo al error cuadrático medio (MSE) y al coeficiente de determinación ( $R^2$ )

### MSE (Mean Squared Error)

- **Definición:** El MSE es la media de los cuadrados de los errores, es decir, la media de las diferencias al cuadrado entre los valores observados y los valores predichos por el modelo.

- **Fórmula:**

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

donde  $y_i$  son los valores observados,  $\hat{y}_i$  son los valores predichos, y  $n$  es el número de observaciones.

- **Interpretación:**

- Un MSE más bajo indica un mejor ajuste del modelo a los datos observados.
- El MSE siempre es un valor positivo y no tiene un límite superior fijo.
- Debido a que está en unidades cuadradas de la variable dependiente, puede ser difícil de interpretar directamente. Sin embargo, compararlo entre diferentes modelos puede ayudar a determinar cuál tiene un mejor rendimiento.

### $R^2$ (Coeficiente de Determinación)

- **Definición:** El  $R^2$  mide la proporción de la varianza en la variable dependiente que es explicada por el modelo.

- **Fórmula:**

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

donde  $y_i$  son los valores observados,  $\hat{y}_i$  son los valores predichos,  $\bar{y}$  es la media de los valores observados, y  $n$  es el número de observaciones.

- **Interpretación:**

- El  $R^2$  varía entre 0 y 1 (aunque puede ser negativo si el modelo es peor que la media).
- Un  $R^2$  de 1 indica que el modelo explica toda la varianza de la variable dependiente.
- Un  $R^2$  de 0 indica que el modelo no explica ninguna de las variaciones en la variable dependiente.

Se adjuntan los resultados del modelo en la tabla 5.5

Variable a predecir	Dataset	Grados del polinomio	Métricas de Evaluación		Tiempos de ejecución (s)	$CO_2$ emitido (kg)
			MSE	$R^2$		
bikes_avg	DataSet1	1	55.49	0.14	0.06	3.03e-08
		2	51.68	0.20	0.50	2.40e-07
		3	42.00	0.35	5.01	2.42e-06
	DataSet2	1	48.23	-0.64	0.06	2.76e-08
		2	79.50	-1.71	0.55	2.64e-07
		3	159.34	-4.42	4.77	2.31e-06

Cuadro 5.5: Resultados de regresión de bikes\_avg con diferentes grados de polinomio

### Interpretación de los resultados

Consideremos el caso de *DataSet1* con un polinomio de grado 1:

- **MSE = 55.49:** Esto significa que, en promedio, el cuadrado de los errores (diferencias entre los valores observados y los predichos) es 55.49. Comparado con otros modelos (polinomio de grado 2 o 3), podemos ver si este error es relativamente alto o bajo.
- **$R^2 = 0.14$ :** Esto indica que el modelo con polinomio de grado 1 explica el 14 % de la varianza en los datos observados. No es un ajuste muy bueno, ya que el modelo solo explica una pequeña parte de la variación total.

Comparando esto con el polinomio de grado 3 para *DataSet1*:

- **MSE = 42.00:** Este valor es menor que el MSE del polinomio de grado 1, indicando un mejor ajuste a los datos.

- **R<sup>2</sup> = 0.35:** El modelo con polinomio de grado 3 explica el 35 % de la varianza en los datos observados, lo cual es mejor que el modelo de grado 1, aunque aún hay una cantidad significativa de varianza que no se explica.

En el caso de *DataSet2* obtenemos incluso peores resultados que usando directamente el valor medio. Luego concluimos que un modelo multipolinomial no es la mejor opción para estimar nuestros datos.

### 5.3.2. Regresión logística

En estadística, la regresión logística es un tipo de análisis de clasificación utilizado para predecir el resultado de una variable categórica en función de las variables independientes o predictoras. Es útil para modelar la probabilidad de un evento que ocurre en función de otros factores [22].

En nuestro caso, usaremos un modelo de regresión logística para predecir los datos de la variable *precisa\_balanceo*, que es una variable categórica con dos clases: verdadero o falso.

#### Métricas de evaluación en algoritmos de clasificación

Las distintas métricas de evaluación en un algoritmo de clasificación para modelos supervisados son:

- **Precisión Global:** es el porcentaje de predicciones correctas sobre el total de predicciones. Se calcula como:

$$\text{Precisión Global} = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}}$$

- **Precisión:** es el porcentaje de verdaderos positivos (TP) sobre el total de predicciones positivas (TP + FP, siendo FP los falsos positivos). Indica qué tan precisas son las predicciones positivas del modelo. Se calcula como:

$$\text{Precisión} = \frac{TP}{TP + FP}$$

- **Recall (Sensibilidad o Tasa de Verdaderos Positivos):** es el porcentaje de verdaderos positivos sobre el total de instancias que realmente son positivas. Indica qué tan bien el

modelo captura los casos positivos reales. Se calcula como:

$$\text{Recall} = \frac{TP}{TP + FN}$$

donde FN son los falsos negativos.

- **F1-Score:** Es la media armónica de la precisión y el *recall*. Es una métrica que busca un balance entre precisión y *recall*, especialmente útil cuando hay una distribución desequilibrada de clases. Se calcula como:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}}$$

Respecto al tipo de medias, se calculan del siguiente modo:

- **Macro Average** calcula la métrica individualmente para cada clase y luego toma el promedio (sin considerar el número de instancias en cada clase). Es útil para evaluar el rendimiento general del modelo cuando las clases están equilibradas.

$$\text{Macro Precision} = \frac{1}{N} \sum_{i=1}^N \text{Precisión}_i$$

$$\text{Macro Recall} = \frac{1}{N} \sum_{i=1}^N \text{Recall}_i$$

$$\text{Macro F1-Score} = \frac{1}{N} \sum_{i=1}^N \text{F1-Score}_i$$

donde  $N$  es el número de clases.

- **Weighted Average** también calcula la métrica individualmente para cada clase, pero luego toma un promedio ponderado basado en el número de instancias en cada clase. Esto es útil cuando las clases están desequilibradas y se desea dar más importancia a las clases con más instancias.

$$\text{Weighted Precision} = \sum_{i=1}^N \left( \frac{\text{Número de instancias de la clase } i}{\text{Total de instancias}} \cdot \text{Precisión}_i \right)$$

$$\text{Weighted Recall} = \sum_{i=1}^N \left( \frac{\text{Número de instancias de la clase } i}{\text{Total de instancias}} \cdot \text{Recall}_i \right)$$

$$\text{Weighted F1-Score} = \sum_{i=1}^N \left( \frac{\text{Número de instancias de la clase } i}{\text{Total de instancias}} \cdot \text{F1-Score}_i \right)$$

En la tabla 5.6 se adjunta los resultados del modelo de regresión logística.

Variable a predecir	Dataset	Precisión Global	Métricas de evaluación			Tiempos de ejecución (s)	Emisiones de CO2
			Tipo de media	Métrica	Valor		
precisa_balanceo	DataSet1	0.79	Macro average	Precisión	0.39	5.18	2.50e-06
				Recall	0.50		
				F1-score	0.44		
	DataSet2	0.75	Weighted average	Precisión	0.62	3.27	1.58e-06
				Recall	0.79		
				F1-score	0.69		
			Macro average	Precisión	0.58	3.27	1.58e-06
				Recall	0.50		
				F1-score	0.43		
			Weighted average	Precisión	0.66	3.27	1.58e-06
				Recall	0.75		
				F1-score	0.64		

Cuadro 5.6: Resultados de predicción con regresión logística

### Conclusiones de los resultados

Como se puede ver en la tabla 5.6 este modelo clasifica con una precisión global muy similar los datos del conjunto que comparte estaciones y los datos del conjunto que no las comparte. Esto podemos interpretarlo como un buen modelo para generalizar a toda la red de estaciones. Sin embargo, si observamos los resultados de las métricas correspondientes a la media *Macro Average*, aunque para el *DataSet2* son ligeramente mejores, concluimos que no son las ideales.

### 5.3.3. Árboles de decisión

La idea principal de los árboles de decisión es subdividir el espacio de datos de entrada para generar regiones disjuntas, de forma que todas las muestras que pertenezcan a la misma región sean de la misma clase. Si una región tiene muestras de diferentes clases, será subdividida en regiones más pequeñas siguiendo el mismo criterio. El proceso finaliza cuando se han partido las muestras de entrada en regiones de forma que para todas se cumple que solo tienen mues-

tras de una única clase [23].

Dentro de los parámetros que pueden afectar a la construcción y definición del árbol, vamos a estudiar principalmente los siguientes:

- ***max\_depth***: se trata del valor de profundidad máxima del árbol. Se emplea para controlar el sobreajuste del modelo haciendo el modelo más sencillo cuando es un valor bajo y más complejo cuando es alto.
- ***max\_features***: indica el número de características a considerar para la división de los nodos. Limitando este valor podemos conseguir una mejor generalización del modelo y evitar el sobreajuste.

En base a la variación de estos dos parámetros, obtenemos los siguientes resultados.

#### Resultados del modelo para ***max\_depth=10*** y ***max\_features=sqrt***

Variable a predecir	Dataset	Precisión Global	Métricas de evaluación			Tiempos de ejecución (s)	Emisiones de CO2
			Tipo de media	Métrica	Valor		
precisa_balanceo	DataSet1	0.76	Macro average	Precisión	0.69	0.16	8.15e-08
				Recall	0.56		
				F1-score	0.56		
	DataSet2	0.72	Weighted average	Precisión	0.73	0.18	8.79e-08
				Recall	0.76		
				F1-score	0.71		
			Macro average	Precisión	0.54		
				Recall	0.51		
				F1-score	0.49		
			Weighted average	Precisión	0.65		
				Recall	0.72		
				F1-score	0.66		

Cuadro 5.7: Resultados de predicción con árboles de decisión con máxima profundidad de 10

**Resultados del modelo para  $\text{max\_depth}=20$  y  $\text{max\_features}=\text{sqrt}$**

Variable a predecir	Dataset	Precisión Global	Métricas de evaluación			Tiempos de ejecución (s)	Emisiones de CO2
			Tipo de media	Métrica	Valor		
precisa_balanceo	DataSet1	0.86	Macro average	Precisión	0.85	0.22	1.09e-07
				Recall	0.78		
				F1-score	0.80		
	DataSet2	0.68	Weighted average	Precisión	0.86	0.23	1.14e-07
				Recall	0.87		
				F1-score	0.86		

Cuadro 5.8: Resultados de predicción con árboles de decisión con máxima profundidad de 20

**Resultados del modelo para  $\text{max\_depth}=30$  y  $\text{max\_features}=\text{sqrt}$**

Variable a predecir	Dataset	Precisión Global	Métricas de evaluación			Tiempos de ejecución (s)	Emisiones de CO2
			Tipo de media	Métrica	Valor		
precisa_balanceo	DataSet1	0.94	Macro average	Precisión	0.92	0.28	1.38e-07
				Recall	0.92		
				F1-score	0.92		
	DataSet2	0.66	Weighted average	Precisión	0.94	0.24	1.16e-07
				Recall	0.94		
				F1-score	0.94		

Cuadro 5.9: Resultados de predicción con árboles de decisión con máxima profundidad de 30

### Conclusiones de los resultados

Como se puede ver en las tablas anteriores, cuando aumentamos la profundidad de los árboles, podemos ver una notable mejora en la precisión global para el conjunto de datos *DataSet1*. Además los valores métricos tienen una alta precisión en este conjunto de datos, tanto para la media macro como para la media ponderada. Sin embargo, podemos ver que este modelo de árboles de decisión, no funciona tan bien para el conjunto de datos *DataSet2*, donde se obtienen unas métricas similares para todas las variaciones de profundidad.

#### 5.3.4. Bosques aleatorios de árboles de decisión

Cuando los clasificadores base son árboles de decisión y se utiliza un muestreo tanto de los elementos del conjunto original de entrenamiento como de sus variables, el clasificador combinado se conoce como *Random Forest* (RF), dado que se trata precisamente de un conjunto (o bosque) de árboles que han sido creados mediante un proceso aleatorio.

La práctica habitual consiste en generar versiones diferentes del conjunto de entrenamiento usando muestreo con reemplazo, de forma que, durante el proceso de construcción de cada árbol de decisión se selecciona aleatoriamente un subconjunto de las variables del conjunto de datos, dando opciones a variables que normalmente quedarían eclipsadas por otras que tuvieran mayor relevancia. Este procedimiento permite medir la importancia relativa de cada variable, estimando el error cometido por el clasificador combinado cuando se altera dicha variable, permutando aleatoriamente sus valores en el conjunto de test [24].

Podemos variar distintos parámetros en un bosque aleatorio. Entre ellos se encuentra el valor máximo de profundidad y el número de estimadores (árboles que formarán el bosque). En la tabla 5.10 se adjuntan los resultados de los modelos con los parámetros que mejores resultados generan.

Variable a predecir	Dataset	Número de estimadores	Profundidad	Precisión Global	Métricas de evaluación			Tiempos de ejecución (s)	Emisiones de CO2
					Tipo de media	Métrica	Valor		
precisa_balanceo	DataSet1	100	30	0.94	Macro average	Precisión	0.92	0.28	1.38e-07
						Recall	0.92		
						F1-score	0.92		
	DataSet2	100	10	0.66	Weighted average	Precisión	0.94	0.24	1.16e-07
						Recall	0.94		
						F1-score	0.94		

Cuadro 5.10: Mejores resultados para un bosque aleatorio de árboles de decisión

## 5.4. Modelo de predicción seleccionado

Tal como se detalla en el notebook **data\_prediction.ipynb**, se realizan otros modelos predictivos como por ejemplo árboles de regresión sobre los conjuntos de datos *DataSet4* y *DataSet5*. Sin embargo, como el objetivo del proyecto es solucionar el problema de balanceo del sistema, con la variable que indica si una estación precisa balanceo, nos sirve para lograr el objetivo. En este apartado vamos a visualizar los resultados de aplicar un bosque de árboles de decisión al conjunto de datos que contiene información de las horas previas con los datos ordenados (*DataSet3*). Se adjuntan los resultados en la tabla 5.11

Dataset	Número de estimadores	Profundidad	Horas previas	Precisión Global	Tiempo de ejecución (s)	Emisiones de CO2
DataSet3	100	30	1	0.882	2.33	1.12e-6
			2	0.884		1.39e-6
			3	0.886		1.47e-6
			6	0.886		1.57e-6

Cuadro 5.11: Resultados de predicción para DataSet3 con 100 estimadores y profundidad de 30

A pesar de que el valor de la precisión global es algo inferior que el obtenido en el apartado 5.3.4, elegimos este modelo para predecir los datos ya que se generaliza más a todo el sistema al incluir datos de todas las estaciones del conjunto. Además, como se detalla en [25], en series temporales es importante considerar los datos de las horas previas como en los modelos ARIMA.

# Capítulo 6

## Optimización de la ruta de reparto

En este capítulo se va a profundizar en el objetivo del proyecto: encontrar una ruta óptima de reparto para balancear las estaciones del sistema. Para lograr este objetivo, vamos a seguir los siguientes pasos:

- Predicción de los datos para saber si una estación necesitará ser balanceada en un momento determinado.
- Elección de los horarios de reparto y los criterios de salida del camión de balanceo.
- Búsqueda de la ruta óptima: algoritmo de Dijkstra. Este algoritmo también conocido como el algoritmo de los caminos mínimos, minimiza la distancia a recorrer, de este modo se conseguirá, no solo reducir el coste de la ruta de reparto, si no también las emisiones de CO<sub>2</sub> [26].

### Predicción de los datos

Para predecir el campo que indica si una estación precisa balanceo, usaremos el modelo descrito en el apartado 5.4: bosques aleatorios de árboles de decisión sobre el conjunto *Data-Set3*. Dado que los resultados de la tabla 5.11 son muy similares usando distintos valores para el parámetro de horas previas, usaremos solo el valor de la última hora que es el que menos tiempo de ejecución y menor emisiones de CO<sub>2</sub> conlleva.

### Elección de las horas de reparto

Para determinar las horas a las que se hará el reparto de bicicletas en las estaciones desbalanceadas, se realiza un estudio visual de las estaciones que precisan balanceo por hora y día de la semana. Se pueden observar los datos en la siguiente figura.

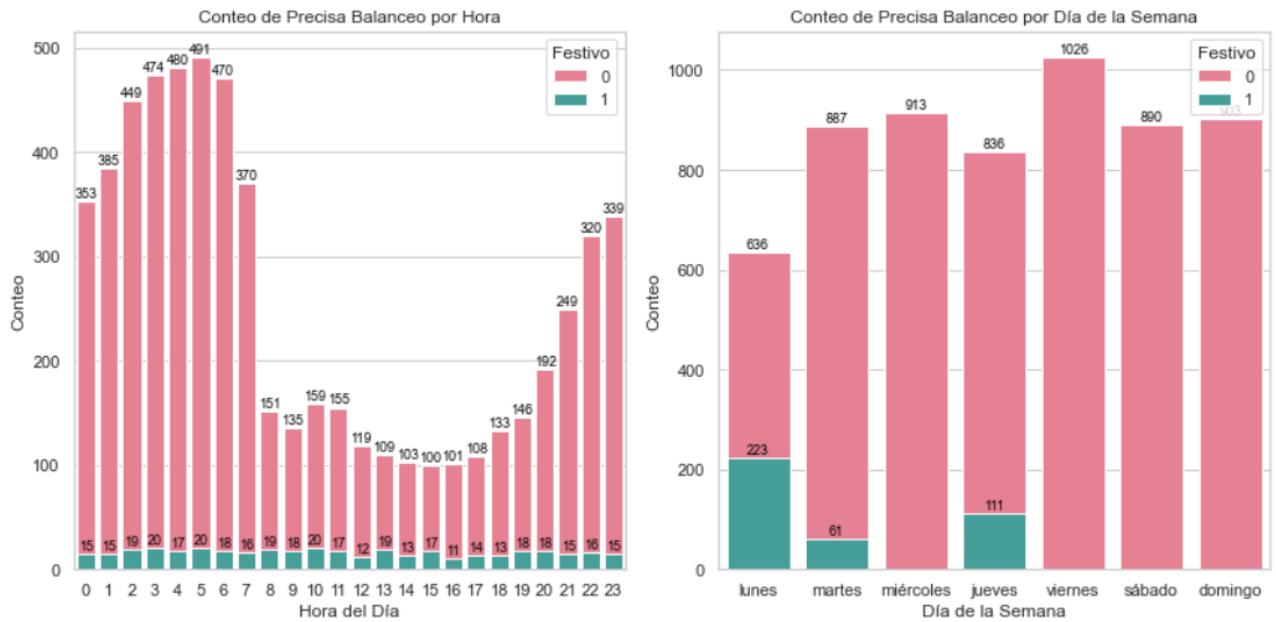


Figura 6.1: Conteo de estaciones que precisan balanceo.

Fuente: elaboración propia; notebook: [data\\_analysis.ipynb](#)

De la figura 6.1 se deduce que:

- Entre las 00:00 y las 07:00 se produce una alta concentración de desbalanceo en las estaciones. Esto puede deberse a las altas ocupaciones en los barrios dormitorio o la escasa disponibilidad en la zona nocturna los días de fin de semana. Haremos un estudio del número de viajes que se producen en cada hora, para ver si balanceando a primera hora de la noche y primera hora de la mañana podemos reducir el desbalanceo en esas horas.
- El segundo pico de desbalanceos se da a las 10:00. Esto puede deberse al aumento de movimientos en la zona de negocios de la ciudad desde los barrios residenciales. Se produce otro aumento de demanda a partir de las 21:00.
- El día con mayor número de estaciones desbalanceadas es el viernes y el menor el lunes.

Del estudio de viajes realizados por hora y día de la semana en el notebook [data\\_prediction.ipynb](#) en el apartado 4.2, se concluye que los mejores horarios para realizar la ruta de balanceo son:

- **Lunes-Viernes:** 00:00, 07:00, 10:00, 13:00, 17:00
- **Sábados:** 06:00, 12:00, 18:00, 23:00
- **Domingos:** no se prestará el servicio de balanceo

## Optimización de la ruta de reparto. Algoritmo de Dijkstra: ruta más corta entre estaciones.

Para elaborar un caso de uso, elegimos un día en concreto y una hora en concreto, en la que al menos 5 estaciones necesiten ser balanceadas. Para determinar la ruta óptima, elegimos la ruta con las distancias más cortas, o lo que es lo mismo, el algoritmo de Dijkstra, cuyo pseudocódigo puede observarse en 5

---

### Algorithm 5 Algoritmo de Dijkstra

---

```

1: Entrada: Grafo  $G = (V, E)$ , nodo origen  $s$ 
2: Salida: Distancias mínimas desde  $s$  a todos los demás nodos en  $V$ 
3: for cada nodo  $v$  en  $V$  do
4:    $dist[v] \leftarrow \infty$ 
5:    $prev[v] \leftarrow \text{null}$ 
6: end for
7:  $dist[s] \leftarrow 0$ 
8:  $Q \leftarrow V$  {Cola de prioridad de todos los nodos en  $V$ }
9: while  $Q$  no está vacía do
10:    $u \leftarrow$  nodo en  $Q$  con  $dist[u]$  mínima
11:   remover  $u$  de  $Q$ 
12:   for cada vecino  $v$  de  $u$  do
13:      $alt \leftarrow dist[u] + w(u, v)$  { $w(u, v)$  es el peso de la arista  $(u, v)$ }
14:     if  $alt < dist[v]$  then
15:        $dist[v] \leftarrow alt$ 
16:        $prev[v] \leftarrow u$ 
17:     end if
18:   end for
19: end while
20:
21: return  $dist, prev$ 
```

---

Fuente: [26]

Con este algoritmo, será necesario definir cuál es el punto inicial, es decir, cuál es la primera estación que será balanceada. Para ello, nos fijaremos en la estación con menor disponibilidad de bicicletas, para así poder llevar el camión de reparto lleno y descargarlo. En caso de que haya más de una estación en la ruta con el mismo número mínimo de bicicletas, entonces miraremos cuál de las estaciones se sitúa en el barrio con menor precio por metro cuadrado. Esta elección, no trivial, se debe a que si queremos instalar almacenes de reparto por la ciudad, conviene que sean en zonas donde el metro cuadrado sea más barato y por lo tanto permita ahorrar costes a la empresa de reparto en la instalación de los almacenes.

En las siguientes figuras, se adjuntan capturas de pantalla de las rutas óptimas para días aleatorios en horas aleatorias.

### Ruta de reparto Lunes a las 07:00 A.M

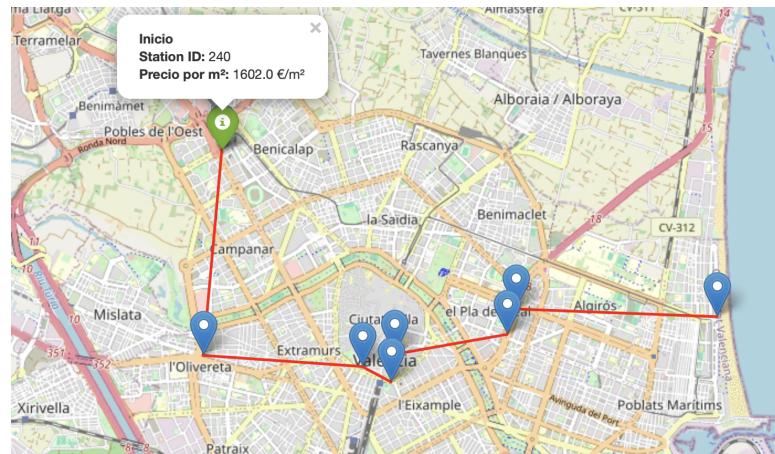


Figura 6.2: Ruta de reparto optimizada para un Lunes a las 07:00 A.M.

Fuente: elaboración propia; notebook: data\_analysis.ipynb

### Ruta de reparto Martes a las 00:00 A.M

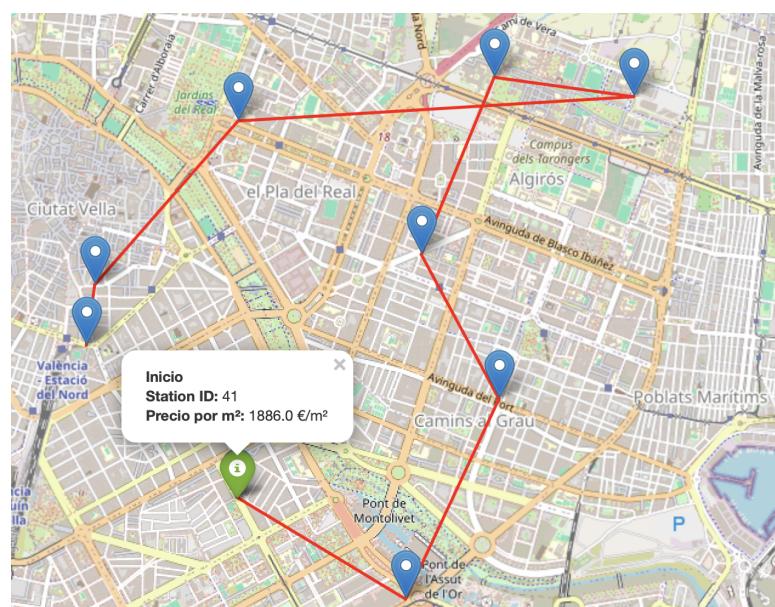


Figura 6.3: Ruta de reparto optimizada para un Martes a las 00:00 A.M.

Fuente: elaboración propia; notebook: data\_analysis.ipynb

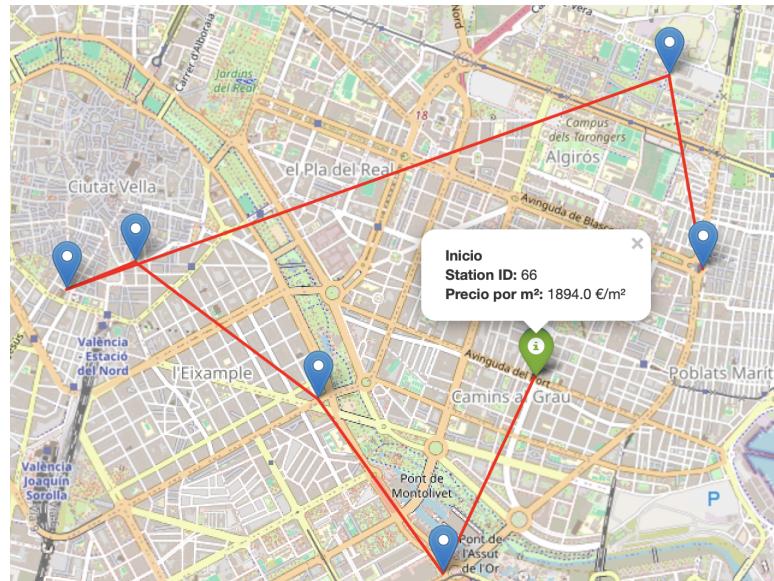
**Ruta de reparto Jueves a las 07:00 A.M**

Figura 6.4: Ruta de reparto optimizada para un Jueves a las 07:00 A.M.

Fuente: elaboración propia; notebook: data\_analysis.ipynb

## 6.1. Conclusiones sobre la optimización de la ruta

Con los ejemplos estudiados en el proyecto podríamos realizar un análisis de las estaciones que más veces precisan el balanceo y definir un control de almacenes por la ciudad. Dado que se desconoce si una estación necesitará bicicletas o bornetas, en base a las estaciones con mayor necesidad de balanceo, podemos definir los puntos estratégicos de la ciudad en los que establecer los almacenes, siempre teniendo en cuenta que el punto de inicio de la ruta va a tener en cuenta el precio por metro cuadrado del barrio en el que se localiza una estación.

Se plantea la optimización de la ruta sin unos horarios fijados, puesto que dependiendo del día y de los parámetros estimados, habrá unas horas distintas en las que haya que realizar la ruta. Sin embargo, desde una perspectiva más realista, se podría definir unos horarios semanales, en base a la necesidad de balanceo y a los turnos de los trabajadores.

# **Capítulo 7**

## **Conclusiones**

Una vez realizado este proyecto de investigación y tras una amplia búsqueda de la literatura previa, podemos concluir unos buenos resultados de predicción para la finalidad con la que se ha planteado el proyecto. Siendo el balanceo uno de los principales problemas de un sistema de bicicletas compartidas, se propone una solución a la empresa que gestiona un sistema real, desde el enfoque de la analítica de datos.

Con las tareas de tratamiento de los datos, hemos podido observar que se produce una alta tasa de estaciones desbalanceadas a lo largo del día, siendo finalmente la granularidad horaria la que se ha elegido para abordar el problema. En un inicio del proyecto el tratamiento de los datos se había planteado con granularidad de minuto, que era como se habían obtenido los datos, pero sin embargo, debido a las herramientas disponibles y a los recursos, a parte de las necesidades de balanceo del sistema, se ha optado por llevarlo a granularidad horaria como en la mayoría de los artículos de la literatura consultada.

Se han combinado los datos con fuentes de interés personal y que se creen que aportan valor al proyecto, destacando los datos climatológicos como datos de mayor aportación y los datos de las zonas verdes como datos de mayor interés. El proyecto ha sido reducido a 19 estaciones, por facilidad de procesamiento, con la idea de poder generalizarlo para toda la red de estaciones. Además se ha reducido la cantidad de años de los que se disponía en un principio.

Las tareas de pre-procesamiento de los datos han tenido un importante peso en el proyecto, destacando la imputación de valores anómalos y la distribución de los datos, así como la correlación de algunas de las variables. Todo ello ha servido para poder definir un conjunto de datos consistente que combina diversas fuentes. Además se ha realizado un importante análisis sobre cómo separar los conjuntos de entrenamiento y testeo, valorando que se trata de una serie tem-

poral donde es importante considerar valores previos en el tiempo y también teniendo en cuenta la estacionalidad de los datos.

La parte referente a los modelos predictivos, en un principio se plantea el problema de estimación del valor de bicicletas medias por hora de una estación mediante regresión multipolinomial obteniendo unos resultados bastante malos, que nos lleva a orientar el problema a la perspectiva de los modelos de clasificación, estableciendo un umbral de balanceo. Para poder determinar si una estación precisa ser balanceada, hemos empleado modelos de regresión logística obteniendo buenos resultados (con una precisión global cercana al 80 %); árboles de decisión (con una precisión global cercana al 75 %); bosques aleatorios de árboles de regresión y de decisión, siendo este último el modelo elegido para concluir, con una precisión que supera el 90 %.

Por último, se ha diseñado una ruta de reparto optimizada para casos concretos en los que se necesita balanceo de las estaciones, aportando un gran valor a la literatura previa consultada, donde solo uno de los proyectos consultados dedica parte del análisis a la optimización de la ruta.

## **7.1. Valoración personal**

Con este trabajo de investigación se ha desarrollado un proyecto de mejora a un sistema de bicicletas compartidas, en una ciudad con un creciente uso del servicio debido al aumento de la población en los últimos años y a las buenas condiciones climáticas medias durante todo el año. Este proyecto de fin de máster, pretendía hacer uso de las herramientas aprendidas en las distintas asignaturas impartidas en el máster y a su vez, ha sido una motivación personal para mejorar un sistema que yo mismo estoy usando desde Enero de 2024, fecha en la que se inicia este proyecto.

El proyecto ha supuesto un reto, dado que la parte de tratamiento de los datos ha sido más extensa de lo que se pensaba. A pesar de tener los datos en un formato ordenado y listo para el tratamiento, no se contempló la cantidad de datos de los que se disponía y la necesidad de hacer uso de herramientas *Big Data* para su procesamiento. Además, se ha optado por añadir fuentes secundarias proporcionadas por el propio Ayuntamiento de la ciudad, algo que se ha sumado a la motivación personal.

A pesar de que la parte referente a la optimización de la ruta no se ha desarrollado de forma extensa por falta de tiempo, creo que aporta un gran valor a otros proyectos desarrollados

en la universidad y podría considerarse el ofrecer esta medida a la empresa que gestiona el BSS.

Respecto a la metodología seguida en la elaboración del proyecto, considero que ha sido clave para llevar un desarrollo continuo y no quedarse atascado en la elaboración del proyecto. Además, la forma en la que se ha estructurado el proyecto ha ayudado a tener una visión clara desde el inicio.

## 7.2. Trabajos futuros

Entre algunas de las mejoras a futuro que se podrían plantear en otros proyectos, cabe destacar:

- Emplear búsqueda de hiperparámetros para mejorar los resultados en las elecciones de los parámetros de los modelos seleccionados.
- Emplear la API proporcionada por el BSS para predecir datos en tiempo real.
- Usar otros modelos de predicción como puede ser ARIMA o redes neuronales.
- Generalizar el proyecto a toda la red de estaciones y no focalizarlo solo en algunas. Para ello sería necesario emplear herramientas con procesadores más potentes que el que se dispone localmente.
- Mejorar la parte de optimización de la ruta: considerar una ruta realista por la ciudad, valorando calles prohibidas y midiendo la distancia de Manhattan en lugar de la euclídea.
- Diseñar el algoritmo de optimización de modo que permita definir si una estación precisa bicicletas o bornetas, y en base a ello definir la ruta optimizada llevando las bicicletas de una estación a otra.
- Realizar un estudio de las emisiones de CO<sub>2</sub> de la actual ruta de reparto y compararlas con las emisiones que generaría la ruta optimizada.
- Integración de todo el proyecto en una aplicación que permita estimar la demanda de las estaciones a futuro y elabore horarios inteligentes de reparto: selección de las horas de reparto y elaboración del plan de ruta optimizada.

# Capítulo 8

## Glosario

- **API:** *Application Programming Interface* o Interfaz de Programación de Aplicaciones
- **BSS:** *Bike Sharing System* o Sistema de Bicicletas Compartidas
- **ETL:** *Extract, Transform and Load* o Extracción, Transformación y Carga
- **ML:** *Machine Learning* o Aprendizaje Automático
- **MSE:** *Mean Squared Error* o Error Cuadrático Medio
- **ODS:** Objetivos de Desarrollo Sostenible
- **SVCPDP** *Single Vehicle One-commodity Capacitated Pickup and Delivery Problem* o Problema de recogida y entrega capacitado para un solo vehículo y un solo producto
- **TFG:** Trabajo Final de Grado
- **UOC:** Universitat Oberta de Catalunya

# Bibliografía

- [1] Directorate-General for Environment. Valencia kicks off 2024 as new european green capital. [https://environment.ec.europa.eu/news/valencia-kicks-2024-new-european-green-capital-2024-01-11\\_en](https://environment.ec.europa.eu/news/valencia-kicks-2024-new-european-green-capital-2024-01-11_en), 2024. Accessed: 2024-06-02.
- [2] JCDecaux Open Data API. <https://developer.jcdecaux.com/#/opendata/vls?page=getstarted>. Accessed: March 2024.
- [3] Wikipedia contributors. Cross industry standard process for data mining. [https://es.wikipedia.org/wiki/Cross\\_Industry\\_Standard\\_Process\\_for\\_Data\\_Mining](https://es.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining), September 2023.
- [4] Yan Chen, Xinlu Sun, Muhammet Deveci, and D'Maris Coffman. The impact of the covid-19 pandemic on the behaviour of bike sharing users. *Sustainable Cities and Society*, 84:104003, 2022.
- [5] Belén Astrid Neumann-Saavedra, Teodor Gabriel Crainic, Bernard Gendron, Dirk C. Mattfeld, and Mirjam Römer. Service network design of bike sharing systems with resource constraints. In *Computational Logistics*, pages 352–366. Springer International Publishing, 2016.
- [6] Güise Lorenzo Rodríguez Aguiar. Análisis y mejora de un sistema de bicicletas compartidas para el balanceo de estaciones. Master's thesis, Universitat Oberta de Catalunya (UOC), Las Palmas de Gran Canaria, junio 2021.
- [7] Zhibin Yang, Jing Chen, Jia Hu, Yantai Shu, and Peng Cheng. Mobility modeling and data-driven closed-loop prediction in bike-sharing systems. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4488–4499, 2019.
- [8] OpenAI. ChatGPT: A large-scale pretrained language model. <https://openai.com/research/chatgpt>, 2022. Accessed: 2024-03-23.

- [9] Jiawei Zhang, Xiao Pan, Moyin Li, and Philip S. Yu. Bicycle-sharing system analysis and trip prediction. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, volume 1, pages 174–179, 2016.
- [10] Jorge Sainero Valle. Estudio y predicción del estado de las estaciones de un sistema de bicicletas compartido. Master's thesis, Universitat Oberta de Catalunya (UOC), Madrid, enero 2023.
- [11] I. Cebrián Martínez. Estudio de la aplicación de algoritmos de enrutado al balanceo de vehículos en sistemas de compartición de bicicletas. Master's thesis, Universitat Politècnica de València, Valencia, 2021. <http://hdl.handle.net/10251/171476>.
- [12] Ingeniería Industrial Online. ¿qué es y para qué sirve google or tools? Fecha de consulta: 23 de marzo de 2024. <https://ingenieriaindustrialonline.com/investigacion-de-operaciones/que-es-y-para-que-sirve-google-or-tools/>.
- [13] Leonardo Caggiani, Rosalia Camporeale, Michele Ottomanelli, and Wai Yuen Szeto. A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transportation Research Part C: Emerging Technologies*, 87:159–182, 2018.
- [14] Daniel Chemla, Frédéric Meunier, and Roberto Wolfler Calvo. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2):120–146, 2013.
- [15] Wikipedia contributors. Latex. <https://en.wikipedia.org/wiki/LaTeX>, 2024. Consultado el 3 de junio de 2024.
- [16] Codecarbon: Estimación de las emisiones de co2 de modelos de aprendizaje automático. <https://github.com/mlco2/codecarbon>. Accedido el 7 de junio de 2024.
- [17] Wikipedia contributors. Valor atípico. <https://en.wikipedia.org/wiki/Outlier>, 2024. Consultado el 1 de mayo de 2024.
- [18] Las Provincias. Los barrios de valencia más caros para comprar una vivienda. <https://www.lasprovincias.es/valencia-ciudad/barrios-valencia-caros-comprar-vivienda-20240223010225-nt.html>, 2024. Consultado en mayo de 2024.
- [19] Scikit learn developers. Preprocesamiento de datos - sklearn. <https://qu4nt.github.io/sklearn-doc-es/modules/preprocessing.html>, s.f.
- [20] Jason Brownlee. How to convert a time series to a supervised learning problem in python. <https://machinelearningmastery.com>, note=Accessed: 2024-05-21, 2017.

- [21] 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 E. Duchesnay. *sklearn.preprocessing.PolynomialFeatures*, 2023. Accessed: 2024-05-21.
- [22] Wikipedia contributors. Regresión logística. [https://en.wikipedia.org/wiki/Logistic\\_regression](https://en.wikipedia.org/wiki/Logistic_regression), 2024. Accedido el 22 de mayo de 2024.
- [23] Raúl Montoliu Colás. *Modelos supervisados*. Fundació Universitat Oberta de Catalunya (FUOC), Av. Tibidabo, 39-43, 08035 Barcelona, primera edition, 2021. Tiempo mínimo de dedicación recomendado: 1 hora.
- [24] Jordi Gironés Roig, Jordi Casas Roma, Julià Minguillón Alfonso, and Ramon Caihuelas Quiles. *Minería de datos: modelos y algoritmos*. Manuales (Tecnología). Editorial UOC, Barcelona, 1<sup>a</sup> edición en lengua castellana, julio 2017, 1<sup>a</sup> edición digital, septiembre 2017 edition, 2017.
- [25] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. *Machine Learning Strategies for Time Series Forecasting*, volume 138. 01 2013.
- [26] M. Noto and H. Sato. A method for the shortest path search by extended dijkstra algorithm. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0, volume 3, pages 2316–2320 vol.3, Oct 2000.*