Relación entre meteorología y contaminación en la ciudad de Madrid

Relationship between meteorology and pollution in the city of Madrid

Alberto Collado Mamblona

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENDE DE MADRID

Trabajo Fin de Grado

en

Ingeniería Informática

2021-2022

Directores:

Rafael Caballero Roldán

Carlos Román Cascón

RESUMEN

Ahora más que nunca se están intentando establecer medidas para reducir los índices de contaminación en las grandes ciudades y así poder adaptarse a las nuevas normativas establecidas por la Unión Europea, cuyo objetivo además de favorecer el medio ambiente es mejorar la salud de las personas que las habitan.

En este trabajo se pretende estudiar las relaciones entre los distintos contaminantes, el tráfico y la meteorología, empleando datos históricos recogidos por las distintas estaciones repartidas por la ciudad de Madrid. A su vez, mediante métodos de *Machine Learning*, que no tienen en cuenta series temporales, poder determinar si se pueden realizar predicciones válidas con el objetivo de poder impulsar medidas que palien los niveles de contaminación e informar a la población con mayor antelación.

PALABRAS CLAVE

Contaminación, meteorología, tráfico, datos abiertos, aprendizaje automático, ciencia de datos.

ABSTRACT

Now more than ever, efforts are being made to establish measures to reduce pollution levels in large cities in order to adapt to the new regulations established by the European Union, whose main objective in addition to favouring the environment, is to improve the health of the people who live there.

The aim of this work is to study the relationships between the different pollutants, road traffic and meteorology. To do this, it has been necessary to use historical data collected by the different stations spread throughout the city of Madrid and machine learning methods, which do not take into account time series. These methods will make it possible to determine if valid predictions can be made, in order to promote measures that reduce pollution levels and inform the population in advance.

KEYWORDS

Pollution, meteorology, traffic, open data, machine learning, data science

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

1.1 Planteamiento del problema

A continuación, se explica cómo se ha planteado el problema, desde el punto de partida, hasta las distintas técnicas utilizadas.

Debido a los problemas que ocasiona en la salud de las personas convivir junto a unos niveles altos de contaminación atmosférica, es importante conocer con antelación cuando van a alcanzarse estos niveles y tomar medidas adecuadas. En este caso, utilizaremos técnicas de *Data Science* para conocer la relación entre los distintos contaminantes junto a la meteorología y el tráfico rodado, para así poder prever el comportamiento de los elementos más perjudiciales y actuar en consecuencia. De este modo, se puede tratar de evitar que se lleguen a sobrepasar los límites establecidos por la OMS y que suponen un riesgo alto para la salud de las personas.

Para este trabajo se han obtenido datos reales proporcionados por el Ayuntamiento de Madrid, categorizados en tres tipos: datos de calidad del aire, datos meteorológicos, datos de tráfico y festividades. El registro de estos datos se lleva a cabo en intervalos horarios. Estos han tenido que ser unificados en un mismo fichero para poder analizarlos correctamente. Desde ese punto, se realiza el estudio y aplicación de las distintas técnicas de *Data Science* para extraer información relevante. El análisis se basa en el uso de algoritmos de *Machine learning,* que actúa transformando los datos de entrada, de tal forma que aproveche mejor la naturaleza de serie temporal que presentan estos.

Podremos ver el comportamiento de los diferentes contaminantes para un mismo modelo y determinar si es posible realizar predicciones satisfactorias con este método, así como la degradación que experimentan las predicciones cuando se realizan para futuros más lejanos.

Para la interpretación de los resultados se ha contado con el apoyo del grupo de micrometeorología y variabilidad climática de la Universidad Complutense de Madrid, METCLIM-UCM. En particular con los consejos de su director, Carlos Yagüe, y la codirección de Carlos Román Cascón.

1.2. Objetivos

El objetivo de este proyecto es ilustrar cómo las diferentes y novedosas técnicas de análisis de datos pueden utilizarse para predecir y reducir los niveles de contaminación.

Los objetivos serían los siguiente:

O1. Mostrar que los datos de calidad del aire, meteorológicos y tráfico están disponibles para el uso público y a su vez mostrar dónde se encuentran disponibles, como descargarlos y tratarlos. De esta manera quien los necesite para otros estudios podrá disponer de ellos.

O2. Utilizar los métodos más populares y actuales que se encuentran en el mundo del análisis de datos, para comprender y preprocesar los datos, generando un fichero único con toda la información que necesitaremos para el objetivo de predicción.

O3. Ilustrar a través de herramientas de visualización las relaciónes entre los distintos conjuntos de datos.

O4. Predecir el comportamiento de los contaminantes mediante la utilización de métodos de aprendizaje automático.

O5. Obtener conclusiones acerca de si es posible realizar estas predicciones y observar qué contaminantes se ven más afectados por factores concretos.

Además, y como objetivo personal se han obtenido conocimientos de programación en *Python* y en el uso de las librerías más relevantes para poder tratar y analizar los datos.

Estos objetivos se corresponden con las fases clásicas del *Data Science*: descarga, preprocesamiento y visualización.

1.3. Contenido de la memoria

La presente memoria consta de los siguientes apartados:

1. **Introducción:** en este capítulo se plantean los problemas, objetivos y mecanismos para alcanzarlos.

2. **Estado del arte:** durante este capítulo se presentarán los principales gases contaminantes y el estado actual de diversos trabajos que abordan su estudio. Además, se especificarán las concentraciones máximas recomendades por la Organización Mundial de la Salud. También se presentarán las principales diferencias de este proyecto con otros proyectos relacionados.

3. **Origen y obtención de los datos**: en este apartado se presentará la fuente de donde se han extraído los datos, y la metodología utilizada para su obtención. Además, también se explica la descripción de estos en su forma original, espacio que ocupan en disco, número de ficheros, etc. Este capítulo corresponde al objetivo O1.

4. **Preprocesamiento y limpieza de los datos**: durante este apartado se explicarán los métodos utilizados para su tratamiento y limpieza, para poder lograr un formato adecuado para su estudio, y de este modo facilitar la extracción de información. También se muestra el resultado obtenido tras la transformación.

5. **Valores nulos y *Outliers*:** se muestran las distintas técnicas y métodos para el tratamiento de los valores nulos y valores inesperados o *outliers,* con el objetivo de no perder demasiada información relevante y eliminar aquella que pueda introducir ruido en los resultados, respectivamente. Durante el desarrollo de este capítulo se lograrán alcanzar los objetivos O2 y O3.

6. **Análisis de datos y extracción de información:** en este apartado se pretenden exponer los algoritmos utilizados para la creación y selección de modelos, además de las técnicas que se pueden utilizar para mejorar las predicciones de estos.
   Asimismo, se presentará la comparación entre los distintos modelos. A lo largo de este capítulo se obtendrán los objetivos O4 y O5.

7. **Conclusiones y futuros trabajos:** en esta sección se enumeran las conclusiones alcanzadas tras los resultados obtenidos durante el apartado anterior. Además, se describen posibles trabajos futuros que se podrían realizar a raíz de este proyecto.

A continuación, se adjunta un enlace al repositorio de *GitHub* que contiene el código creado durante este trabajo:

- https://github.com/albercol/TFG

1.4. Herramientas utilizadas

En este apartado se detallan las herramientas y librerías más importantes, y que más se han utilizado para la realización del proyecto.

I. Python

Lenguaje de programación de alto nivel, cuyo objetivo es centrarse en la legibilidad del código. Se trata de un lenguaje de programación multiparadigma, ya que soporta la programación orientada a objetos, programación imperativa y programación funcional.

A su vez, es el lenguaje de programación más utilizado en el análisis y *Data Science* por su amplia comunidad, la gran cantidad de herramientas estadísticas y matemáticas, y por su facilidad de aprendizaje.

II. Jupyter Notebook

Jupyter Notebook es una aplicación web de código abierto que permite a los científicos de datos crear y compartir documentos que integran código en vivo, ecuaciones, resultados computacionales, visualización de gráficos y otros recursos, junto con texto explicativo. Pudiendo volver a ejecutar fragmentos de código sin necesidad de ejecutar todo el *script*.

El mayor atractivo para el análisis y *Data Science* es la posibilidad de generar reportes, gráficos y resultados en una misma interfaz.

Fue lanzado en 2015 por la organización sin ánimo de lucro Proyecto Jupyter.

III. Librerías

- Librería *os*: es un módulo que provee de una manera fácil y versátil el uso de funcionalidades dependientes del sistema operativo.

  Con dicho módulo se ha accedido a la carpeta donde se almacenan los datos obtenidos previamente sobre contaminación atmosférica.

- Librería *glob*: es un módulo que encuentra todos los nombres de rutas que se asemejan a un patrón especificado, de acuerdo con las reglas usadas en un terminal *Unix*.

  Con dicho módulo se ha conseguido identificar el nombre de los ficheros que han sido descargados y se les ha asignado un nombre identificativo acorde con el contenido de este.

- Librería *Pandas*: es una librería especializada en el manejo y análisis de estructuras de datos.

  Con esta librería se ha conseguido abrir y leer los ficheros, así como guardarlos en una variable *dataframe*, además de almacenarlos en una lista para posteriormente poder concatenar todo el contenido de esta en un único fichero que almacena toda la información.

- Librería *Beautifulsoup*: es una librería de Python que permite extraer información de contenido en formato HTML o XML. Para usarla es necesario especificar un *parser*, responsable de transformar un documento HTML o XML en un árbol complejo de objetos Python. Esto permite que podamos interactuar con los elementos de una página web como si usáramos las herramientas de desarrollador de un navegador.

Esta herramienta ha sido utilizada para examinar los elementos de la página web del Ayuntamiento y automatizar el proceso de descarga de todos los ficheros necesarios para la realización de este trabajo.

- Librería *request*: Es una librería permite realizar peticiones HTTP y FTP y ha sido utilizada para la descarga de parte de los ficheros necesarios para este proyecto.

- Librería *Zipfile*: esta librería proporciona herramientas para crear, leer, escribir, agregar y mostrar un fichero ZIP.

  Dicha librería, ha sido utilizada debido a que uno de los ficheros descargados y que utilizaremos durante este trabajo se encontraba en este formato. Esta herramienta, nos permitirá extraer los componentes individuales

- Librería *Scikit-Learn*: esta librería cuenta con algoritmos de clasificación, regresión, *clustering* y reducción de dimensionalidad. Además, presenta la compatibilidad con otras librerías como *NumPy, SciPy y Matplotlib*, que serán utilizadas durante la etapa de predicción.

- Librería *NumPy*: es una librería especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos, esta herramienta sirve como apoyo a *Pandas* para representar vectores de datos.

  Incorpora objetos llamados *array*s, que permiten representar colecciones de datos de un mismo tipo en varias dimensiones y funciones eficientes para su manipulación.

- Librería *Matplotlib*: es una librería especializada en la creación de gráficos en dos dimensiones. Esta herramienta, será utilizada durante este trabajo para representar las gráficas y poder visualizar los resultados obtenidos.

- Librería *Seaborn*: al igual que *Matplotlib*, es una herramienta que proporciona una interfaz de alto nivel para la representación de gráficos estadísticos. Esta librería, será utilizada para la representación visual de los valores obtenidos.

2. ESTADO DEL ARTE

En este apartado se muestran cuáles son los principales gases contaminantes y las principales características de estos. Asimismo, también se exponen los límites establecidos por la Organización Mundial de la Salud para considerar que una zona tiene niveles altos de contaminación, y el criterio para actuar sobre un determinado territorio.

Igualmente, en este capítulo se plantean las principales características de este trabajo respecto a otros realizados, y se exponen las distintas herramientas utilizadas.

2.1. Principales gases contaminantes

Se ha realizado un estudio previo para determinar cuáles son los principales gases contaminantes, desde el punto de vista de la afección sobre la salud humana e impacto en el ecosistema. En él se determinó que dichos gases son el Dióxido de azufre (SO_2), Monóxido de Carbono (CO), Óxidos de Nitrógeno (NO_2, NO) y Ozono (O_3); además, de las partículas menores de 2.5 micras y partículas menores de 10 micras. (Enviraiot, 2020) (ecologistasenaccion, n.d.)

i. Dióxido de azufre

Este contaminante tiene un periodo de vida de alrededor de 72 horas, y su principal fuente es la quema de combustibles fósiles ricos en azufre, así como las erupciones volcánicas.

Su inhalación repercute en afecciones del sistema respiratorio y en el funcionamiento de los pulmones. Asimismo, puede producir irritación ocular.

ii. Monóxido de Carbono

Este contaminante tiene un periodo de vida de unos 30-90 días, su principal fuente es la quema de combustibles fósiles y biomasa. En la ciudad de Madrid viene derivado principalmente de los vehículos de combustión.

La exposición a este contaminante reduce la capacidad de transportar oxígeno de la sangre a los tejidos corporales.

iii. Óxidos de Nitrógeno

Se pueden identificar varios tipos de Óxidos de Nitrógeno.

El Monóxido de Nitrógeno (NO) es un tóxico que se oxida con rapidez convirtiéndose en Dióxido de Nitrógeno. El origen es tanto natural, debido a la descomposición bacteriana e incendios, como derivado de la actividad humana, debido a vehículos motorizados y quema de combustibles fósiles.

A su vez, el Dióxido de Nitrógeno (NO_2) resulta de la combustión efectuada a altas temperaturas, tanto de origen natural como antropogénico. Este tóxico, es irritante y precursor de la formación de contaminantes secundarios como el Ozono y las partículas menores de 2.5 micras.

Además, ambos ($NO_{x)}$ tienen un efecto corrosivo sobre la piel y el sistema respiratorio, pudiendo ser causante de un edema pulmonar cuando el sujeto se expone a concentraciones elevadas.

iv. Ozono troposférico

Se genera por la reacción fotoquímica de los precursores NO_x y CO, sustancias emitidas de forma directa que reaccionan con la luz solar en condiciones atmosféricas estables.

Su impacto para la salud es notable, debido a que posee un elevado carácter oxidativo que le proporciona la capacidad de destruir organismos completos.

v. PM2.5

La principal fuente de las partículas menores de 2,5 micras es la emisión producida por los vehículos diésel, derivados del tráfico rodado de las ciudades.

Su impacto sobre la salud está relacionado con enfermedades de tipo respiratorio, tales como la bronquitis y dolencias de tipo cardiovascular. Además, de provocar asma y alergias entre la población infantil.

Gracias a su facilidad para ser respirado y su reducido tamaño, puede viajar profundamente en los pulmones. Lo que lo hace que sea más perjudicial que las partículas menores de 10 micras, y tenga un mayor impacto en las personas.

vi. PM10

Las fuentes de emisión de estas partículas pueden ser móviles o estacionarias, un 77,9% del total procede del polvo resuspendido existente en la atmósfera. La industria, la construcción y los comercios suponen el 7,6%, mientras que el tráfico rodado supone un 6,5%.

La exposición prolongada puede provocar efectos nocivos en el sistema respiratorio. No obstante, como se ha mencionado anteriormente, es menos perjudicial que las partículas menores de 2.5 micras, ya que al tener un mayor tamaño estas no logran atravesar los alveolos pulmonares, quedando retenidas en la mucosa que recubre las vías respiratorias, siendo expulsadas de manera relativamente eficaz mediante la tos.

2.2. Directrices mundiales de calidad del aire

En este apartado se pretende mostrar los niveles establecidos por la OMS en el año 2005, y la última actualización en 2021, de los niveles máximos establecidos por esta organización, que establece el criterio para considerar que una zona contiene niveles altos de contaminación. «tabla 1.»

Tabla 1. Comparación de los niveles recomendados de la calidad del aire entre 2005 y 2021. (WHO)

Contaminante	Promedio de tiempo	Directriz de calidad del aire de 2005	Directriz de calidad del aire de 2021
PM2.5, µg/m^3	Anual	10	5
	24-horas[a]	25	15
PM10, µg/m^3	Anual	20	15
	24-horas[a]	50	45
O_3, µg/m^3	Temporada alta[b]	-	60
	24-horas[a]	100	100
NO_2, µg/m^3	Anual	40	10
	24-horas[a]	-	25
SO_2, µg/m^3	24-horas[a]	20	40
CO, mg/m^3	24-horas[a]	-	4

([a]) Percentil 99 (es decir, 3-4 días de excedencia por año)

([b]) Promedio de la concentración media de O_3 máxima diaria en ocho horas en los seis meses consecutivos con la concentración más alta de seis meses**.**

2.3. Criterio para determinar que una zona está contaminada

La evaluación de una zona se realiza para cada contaminante y de acuerdo con la situación de la estación, de modo que cuando se evalúa más de una estación en la misma situación prevalecerá la información de la estación con niveles más altos.

2.4. Trabajos relacionados

En este apartado se expondrán algunos trabajos similares, los cuales utilizan diferentes técnicas de *Machine Learning* y *Deep Learning* para realizar predicciones sobre los distintos contaminantes.

El primer trabajo que se ha tenido en cuenta es el trabajo de fin de grado de Daniel García Sousa de la UC3M, Universidad Carlos III de Madrid, cuyo título es "*Data Science, explotación y análisis de la calidad del aire en la ciudad de Madrid"*. (Sousa, 2019) Este trabajo utiliza modelos de *Machine Learning* de regresión como el ARIMA y redes

neuronales LSTM, tanto simples como multicapa, cuyo objetivo es explorar la precisión, alcance y utilidad de los distintos algoritmos de *Machine Learning*, tanto de aprendizaje supervisado, como no supervisado para todos los contaminantes. Utilizando datos horarios de contaminación proporcionados por el Ayuntamiento de Madrid y datos meteorológicos diarios proporcionados por la AEMET.

En segundo lugar, se ha tenido en cuenta el trabajo de fin de máster de Gabriel Villalba Pintado de la UOC, Universitat Oberta de Catalunya, cuyo título es *"Predicción de la calidad del aire de Madrid mediante modelos supervisados"* (Pintado, 2019). El cual, ha tenido en cuenta datos horarios de calidad del aire disponibles en la web del Ayuntamiento de Madrid, datos climatológicos proporcionados por la AEMET y datos sobre el calendario laboral, disponibles también en la web del Ayuntamiento de Madrid. El objetivo es crear un modelo de predicción de calidad del aire basado en tres tipos distintos de redes neuronales, *Multilayer Perceptron* (MLP), *Long Short-Term Memory* (LSTM), *Convolutional Neural Network* (CNN) y *Support Vector Machines* (SVM).

Por último, se ha considerado el trabajo de fin de máster de Lorena García Fernández de la UNED, Universidad Nacional de Educación a Distancia, cuyo título es *"Predicción de la contaminación por dióxido de nitrógeno en la ciudad de Madrid mediante modelos de Inteligencia artificial"* (Fernandez, 2018-2019). Este, utiliza datos de calidad del aire proporcionados por el Ayuntamiento de Madrid y datos meteorológicos proporcionados por la Agencia Estatal de Meteorología, centrándose exclusivamente en el NO_2.

A diferencia de los proyectos mencionados anteriormente, se ha utilizado el modelo *Random Forest Regressor* junto a los datos horarios de calidad del aire, para todos los contaminantes disponibles, los datos horarios meteorológicos para la misma estación en la que también se están midiendo los datos de calidad del aire, así como los datos de tráfico recogidos por las estaciones que miden la densidad de tráfico, y el calendario laboral, disponibles también en la web del Ayuntamiento de Madrid. Durante la realización de este proyecto se ha evitado utilizar algoritmos que sean específicos para el tratamiento de series temporales, véase el algoritmo ARIMA o las redes neuronales LSTM mencionados anteriormente, forzando los datos de entrada del modelo añadiendo variables del pasado. De este modo, se consiguen paliar las consecuencias de no tratar el conjunto de datos como una serie temporal.

3. ORIGEN Y OBTENCIÓN DE LOS DATOS

En este capítulo se pretende encontrar una fuente fiable para la obtención de los datos, en nuestro caso datos de calidad del aire, datos meteorológicos, datos de tráfico y festividades, así como otros datos de interés referentes a las estaciones de medición. Con el objetivo de su posterior preparación y limpieza para que puedan ser procesados de forma adecuada.

Al final de cada subapartado se podrá encontrar una copia de la fuente original de los datos, los cuales pueden variar en el tiempo, y un enlace al repositorio donde se encuentra una copia del conjunto de datos, pudiéndose consultar en cualquier momento pese a la variación de los datos originales, cumpliendo así con el objetivo «O1» del trabajo.

3.1. Origen y formato de los datos

Los datos que han sido utilizados en este proyecto provienen principalmente del portal de Datos Abiertos del Ayuntamiento de Madrid, donde este pone a disposición del usuario datos del gobierno municipal para así poder impulsar el desarrollo de herramientas.

El Ayuntamiento de Madrid facilita los datos sobre contaminación, así como de meteorología, cuyos formatos son tabla de Excel «.xlsx», fichero «.csv» o fichero de texto «.txt» y datos sobre los aforos de tráfico, cuyos formatos son tabla de Excel «.xlsx» y fichero «.csv».

Asimismo, también se proporciona información sobre la localización de cada una de las estaciones de medición, que parámetros mide cada una de ellas, su latitud y longitud entre otros.

3.1.1. Datos sobre calidad del aire

Los datos horarios de contaminación atmosférica son proporcionados por el Ayuntamiento de Madrid y actualizados diariamente para un conjunto determinado de contaminantes. De igual forma, la fuente de los datos proporciona otro fichero de información sobre las diferentes estaciones de calidad del aire repartidas por Madrid, donde podemos encontrar: el código de estación, nombre de la ubicación, altitud, latitud

y longitud, además de los gases contaminantes que recoge (marcados con una «X»), entre otros datos. (Madrid, Calidad del aire. Datos horarios desde 2001, n.d.) (Madrid, Calidad del aire. Estaciones de control, n.d.)

Los datos de calidad del aire están recogidos desde el año 2001 hasta 2021 en formato «.zip», de la siguiente forma.

El campo PUNTO DE MUESTREO incluye el código de la estación completo, formado por: provincia, municipio y estación, más la magnitud y la técnica de muestreo. «tabla 2.»

PROVICIA	MUNICIPIO	ESTACIÓN	MAGNITUD	PUNTO DE MUESTREO	ANO	MES	DÍA	H01	V01	H02	V02
28	79	4	1	28079004_1_38	2019	1	1	23	V	17	V

Tabla 2. Representación de los datos horarios de calidad del aire en su formato original.

Fuente: calidad del aire, estaciones.

Repositorio: GitHub. Se ha añadido como «.zip» en el repositorio debido al tamaño del conjunto de datos.

3.1.2. Datos meteorológicos

Los datos horarios sobre meteorología han sido recogidos y proporcionados por el Ayuntamiento de Madrid y actualizados diariamente para las variables meteorológicas recogidas por cada estación. La principal diferencia respecto al fichero de calidad del aire es que el Ayuntamiento de Madrid no dispone de datos meteorológicos anteriores al 1 de enero de 2019, debido a que la infraestructura de la red meteorológica se puso en marcha en 2018.

Al igual que en el apartado anterior, la fuente de los datos proporciona otro fichero con información sobre las diferentes estaciones meteorológicas repartidas por la ciudad, donde podemos encontrar: el código de estación, nombre de la ubicación, altitud, latitud y longitud, además de los gases contaminantes que recoge (marcados con una «X»), entre otros datos. (Madrid, Datos meteorológicos. Datos horarios desde 2019, n.d.) (Madrid, Datos meteorológicos. Estaciones de control, n.d.)

El campo PUNTO DE MUESTREO incluye el código de la estación completo, formado por: provincia, municipio y estación, más la magnitud y la técnica de muestreo. «tabla 3.»

PROVICIA	MUNICIPIO	ESTACIÓN	MAGNITUD	PUNTO DE MUESTREO	ANO	MES	DÍA	H01	V01	H02	V02
28	79	104	82	28079104_82_98	2019	1	1	23	V	17	V

Tabla 3. Representación de los datos horarios meteorológicos en su formato original.

Fuente: meteorología, estaciones.

Repositorio: GitHub.

3.1.3. Datos sobre el tráfico

Los datos horarios sobre el tráfico han sido recogidos y proporcionados por el Ayuntamiento de Madrid y actualizados diariamente para todas las estaciones desde enero de 2018, en los cuales se recogen el número de vehículos que pasan por las estaciones permanentes de medición de tráfico. Al igual que ocurre con los conjuntos de datos anteriores, la fuente también proporciona información específica de las diferentes estaciones de tráfico repartidas por Madrid, desde el código de estación, nombre de esta, latitud y longitud, el sentido y orientación. (Madrid, Aforos de tráfico en la ciudad de Madrid permanentes, n.d.)

Las horas están comprendidas en formato de 12H. (01:00-12:00). el parámetro FSEN indica el sentido del tráfico 1 ó 2 y la franja horaria, «-», para datos tomados de 1:00-12:00 h y «=» para datos tomados de 13:00-24:00 h. «tabla 4.»

FDIA	FEST	FSEN	HOR1	HOR2
01/11/18	ES01	1-	808	721
01/11/18	ES01	1=	1134	1168
01/11/18	ES01	2-	735	589
01/11/18	ES01	2=	1154	1143

Tabla 4. Representación de los datos horarios sobre tráfico en su formato original.

Fuente: tráfico

Repositorio: GitHub.

3.1.4. Datos sobre festividades

Los datos de los días festivos han sido recogidos y proporcionados igualmente por el Ayuntamiento de Madrid.

A partir de la fecha 07/01/2019, correspondiente a la columna Día, se realiza un cambio de formato. Dejando de contarse los laborables/festivos diariamente para solo contabilizar los festivos. «tabla 5.»

Día	Día_semana	Laborable/festivo	Tipo de festivo	Festividad
01/01/2013	Martes	Festivo	Festivo nacional	Año Nuevo
02/01/2013	Miércoles	Laborable		

Tabla 5. Representación de los datos sobre festividades en su formato original.

Fuente: festividades.

Repositorio: GitHub.

3.2. Obtención de los datos

En este apartado se especifica los pasos que se han llevado a cabo para obtener los conjuntos de datos que han sido utilizados en este proyecto.

3.2.1. Obtención de datos sobre calidad del aire

Los datos proporcionados, están disponibles para descargarse en formato «.zip», uno por año. Como se tienen registros desde el año 2001, se ha realizado un programa que filtra por el año, obteniendo exclusivamente los datos hasta el 2019. Este programa, utiliza *BeautilfulSoup* para examinar la página y descargar los ficheros con dicha extensión en una carpeta creada con *os*, estos posteriormente han sido descomprimidos en esta con *Zipfile*, generando una gran cantidad de ficheros basura que han sido borrados con *os* para no saturar la memoria. Se entiende como ficheros basura a los ficheros descargados «.xml», «.txt» y «.zip» que no serán utilizados durante la realización de este proyecto.

En total se han descargado 34.7MB, correspondientes a treinta y seis ficheros con extensión «.csv», estos ficheros están repartidos de doce en doce en cada «.zip» y representan los datos de las mediciones de cada contaminante para cada mes del año.

Después, se procedió a concatenar todos los ficheros obtenidos en uno único, cuyo espacio en disco es de 31.5MB.

A su vez se ha descargado mediante *request* el fichero que contiene los datos de las estaciones de calidad del aire, el cual ocupa un total de 5.23KB.

3.2.2. Obtención de datos meteorológicos

Para la descarga de los datos meteorológicos, se ha escrito un programa que mediante *Beautifulsoup* inspecciona el código HTML de la página, para extraer los enlaces que nos permitan descargar todos los ficheros en una carpeta creada con *os*. En total se han descargado 21.6MB, estos corresponden a treinta y seis ficheros, cuya información representan las mediciones para cada mes del año desde el 2019. A partir de aquí se procedió a concatenar todos los ficheros obtenidos en uno único, cuyo espacio en disco es de 20.3MB.

Posteriormente, se ha procedido a realizar la descarga del fichero que contiene los datos de las estaciones meteorológicas, de la misma forma que en apartado anterior «sección 3.2.1.». Cuyo espacio total en disco es de 4.87 KB.

3.2.3. Obtención de datos sobre tráfico

La descarga de este conjunto de datos se ha realizado de forma muy similar a la del apartado anterior «sección 3.2.2.». El espacio en disco del conjunto de datos total es de 19.3MB, correspondientes a un total de cuarenta y cinco ficheros que corresponden a las mediciones de las estaciones de tráfico por cada mes, Estos datos están disponibles desde el año 2018, por lo que se ha tomado la decisión de descargarlos sin filtrar el año, para posteriormente obtener los datos a partir de esta fecha.

El conjunto de ficheros descargados debería ser de cuarenta y ocho en vez de cuarenta y cinco obtenidos, esto es debido a un retraso de tres meses en la publicación de los datos por el Ayuntamiento de Madrid.

Al igual que en los apartados mencionados anteriormente, se ha procedido a concatenar todos los ficheros generados en uno único, para poder ser limpiado y procesado de forma más eficiente. Se ha generado un único fichero de 32.8MB.

3.2.4. Obtención de festividades

La página impide descargar ficheros mediante *request*, por lo que se ha utilizado las librerías *Beautifulsoup* para examinar la página y *os* para que cree una carpeta donde almacenar el fichero descargado.

En total, se ha descargado únicamente un fichero con un peso total de 109KB.

A modo resumen se adjunta una tabla en la que se engloban el total de lo nombrado anteriormente, de forma que se muestre de una forma más visual el conjunto de archivos descargados. «tabla 6.»

	Calidad del Aire	**Meteorología**	**Tráfico**	**Festividades**	**Estaciones Calidad del Aire**	**Estaciones Meteorológicas**	**Total**
Ficheros Unificados	31,5MB	20,3MB	32,8MB	109KB	4,87 KB	5,23KB	84,7MB
Conjunto de ficheros	36 ficheros	36 ficheros	45 ficheros	_	_	_	75,6MB
	34,7MB	21,6MB	19,3MB				

Tabla 6. Tabla con el espacio que ocupa cada uno de los ficheros descargados.

4. PREPROCESADO DE DATOS

A continuación, se detallan los pasos seguidos para cada uno de los conjuntos de datos, para transformarlos, limpiarlos y unificarlos, con el objetivo de darles un formato más adecuado para su análisis.

4.1. Selección de la estación de calidad del aire y meteorología y tráfico

Para llevar a cabo esta selección, se han tenido que descargar los ficheros sobre las estaciones de calidad del aire y las estaciones meteorológicas proporcionados por el Ayuntamiento de Madrid. A diferencia de otros trabajos, se ha procedido a descartar el conjunto de datos proporcionado por la AEMET, debido a que, aunque estos contengan registros disponibles desde el año 2013, han sido tomados de forma diaria, por lo que se ha considerado que no aprovecha las características del resto del conjunto de datos, cuyos registros están tomados de forma horaria. Además, como se verá posteriormente, algunas de las estaciones que miden la calidad del aire coinciden en latitud y longitud con las estaciones meteorológicas, al contrario de lo que ocurre con los datos proporcionados por AEMET. Esta última característica, puede resultar importante para predecir el comportamiento de los elementos recogidos por la estación de calidad del aire y ha resultado decisiva para optar por el conjunto de datos meteorológicos proporcionados por el Ayuntamiento de Madrid.

Posteriormente, se ha procedido a analizar la intersección entre ambos conjuntos para detectar cuales de las estaciones tanto meteorológicas, como de calidad del aire, se encuentran situadas en el mismo lugar mediante la columna LATITUD y LONGITUD disponibles en ambos ficheros.

Tras la obtención de dichas estaciones se ha procedido a seleccionar la estación que recoje más datos de interés para el proyecto.

	ESTACION	CODIGO_CORTO	NO2	SO2	CO	PM10	PM2_5	O3	BTX
0	Pza. de España	4	X	X	X	NaN	NaN	NaN	NaN
1	Escuelas Aguirre	8	X	X	X	X	X	X	X
2	Arturo Soria	16	X	NaN	NaN	NaN	NaN	X	NaN
3	Farolillo	18	X	X	X	X	NaN	X	X
4	Casa de Campo	24	X	NaN	NaN	X	X	X	X
5	Pza. del Carmen	35	X	X	X	NaN	NaN	X	NaN
6	Moratalaz	36	X	X	NaN	X	NaN	NaN	NaN
7	Cuatro Caminos	38	X	NaN	NaN	X	X	NaN	X
8	Barrio del Pilar	39	X	NaN	NaN	NaN	NaN	X	NaN
9	Ensanche de Vallecas	54	X	NaN	NaN	NaN	NaN	X	NaN
10	Pza. Elíptica	56	X	NaN	X	X	X	X	NaN
11	El Pardo	58	X	NaN	NaN	NaN	NaN	X	NaN
12	Juan Carlos I	59	X	NaN	NaN	NaN	NaN	X	NaN

Figura 1. Captura de las estaciones de calidad del aire cuyo indicador X en las variables indica que mide la concentración del contaminante y NaN indica que no realiza la medición.

Fuente: elaboración propia mediante librería pandas de python.

	ESTACION	CODIGO_CORTO	VV (81)	DV (82)	T (83)	HR (86)	PB (87)
0	Plaza España	4	NaN	NaN	X	NaN	NaN
1	Escuelas Aguirre	8	NaN	NaN	X	X	NaN
2	Arturo Soria	16	NaN	NaN	X	X	NaN
3	Farolillo	18	NaN	NaN	X	NaN	NaN
4	Casa de Campo	24	X	X	X	X	X
5	Plaza del Carmen	35	NaN	NaN	X	X	NaN
6	Moratalaz	36	NaN	NaN	X	X	NaN
7	Cuatro Caminos	38	NaN	NaN	X	X	NaN
8	Barrio del Pilar	39	NaN	NaN	X	X	NaN
9	Ensanche de Vallecas	54	X	X	X	X	NaN
10	Plaza Elíptica	56	X	X	X	X	X
11	El Pardo	58	NaN	NaN	X	X	NaN
12	Juan Carlos I	59	X	X	X	X	X

Figura 2. Captura de las estaciones de meteorológicas cuyo indicador X en las variables indica que mide la variable meteorológica y NaN indica que no realiza la medición.

Fuente: elaboración propia mediante librería pandas de python.

Como se puede observar en las anteriores figuras «figura 1.» y «figura 2.», la estación de Plaza Elíptica, correspondiente a la fila diez, es una de las estaciones más interesantes para realizar nuestro estudio, debido a la cantidad de variables recogidas.

Después de realizar el primer filtrado se ha procedido determinar las estaciones de tráfico más cercanas a la estación de calidad del aire/meteorológica y la relevancia de estas. Obteniéndose, que cercanas a esta estación seleccionada se encuentran tres estaciones de medición de tráfico, situadas en las calles: Av. de Oporto cuyo código de estación es el 10, Av. Rafaela Ibarra cuyo código de estación es el 54 y Calle de Marcelo Usera cuyo código de estación es el 53. «figura 3.»

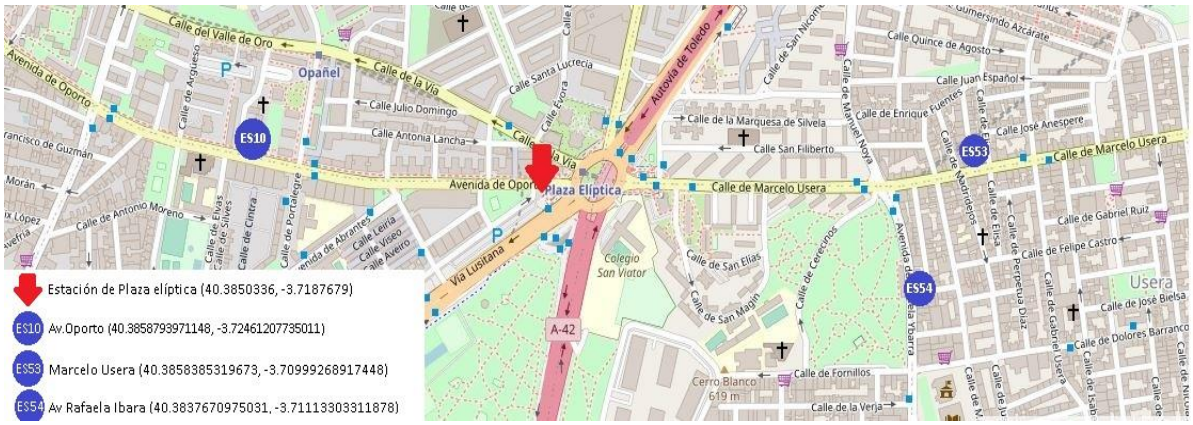


Figura 3. Disposición de la estación de contaminación y meteorología y estaciones de tráfico.

Fuente: callejero-madrid.es

El código de estación se encuentra en el fichero *Ubicación de sentidos permanentes y sentidos de calles*, en la misma página donde se han descargados los datos de tráfico.

4.2. Limpieza de datos

Tras analizar cuál de todas las estaciones de calidad del aire/meteorología es más interesante, se ha procedido a eliminar la información extra que se encuentra en los ficheros de calidad del aire y meteorológicos, dejando solo los datos referentes a la estación de Plaza Elíptica. Asimismo, se han eliminado del fichero de tráfico las estaciones que se encuentran lejos del punto de muestreo, dejando solo las tres estaciones seleccionadas durante el apartado anterior. «sección 4.1.»

El resultado de la limpieza de los datos se puede encontrar en el siguiente enlace al repositorio de GitHub:

- *https://github.com/albercol/TFG/tree/main/Fase%201.3/PlazaEliptica*

4.2.1 Limpieza de datos sobre calidad del aire y meteorología

Durante este apartado, se ha procedido a la extracción del conjunto de datos de calidad del aire y meteorología de todas las variables que son irrelevantes para el objetivo de este proyecto, quedándose exclusivamente con las variables MAGNITUD «tabla 7.», que identifica de forma numérica el tipo de dato que recoge, el rango de horas (H01, …, H24),

que almacena el valor para cada magnitud a lo largo del día. Por último, nos hemos quedado con las variables DIA, MES y AÑO, cuyo objetivo es identificar la fecha para todo el conjunto de horas de cada magnitud.

Magnitud Calidad del Aire		**Magnitud Calidad del Aire**	
06	Monóxido de Carbono	81	VELOCIDAD VIENTO
07	Monóxido de Nitrógeno	82	DIR. DE VIENTO
08	Dióxido de Nitrógeno	83	TEMPERATURA
09	Partículas < 2.5 µm	86	HUMEDAD RELATIVA
10	Partículas < 10 µm	87	PRESION BARIOMETRICA
12	Óxidos de Nitrógeno	89	PRECIPITACIÓN
14	Ozono		

Tabla 7. Representación numérica para las variables magnitud en los archivos de calidad del aire y meteorología.

A continuación, se muestra un ejemplo del resultado de la limpieza de los datos para el archivo de calidad el aire, para el archivo de meteorología obtendremos el mismo resultado. «tabla 8.»

ANO	**MES**	**DIA**	**MAGNITUD**	**H01**	**…**	**H24**
2019	4	1	6	0.3	…	0.2
2019	4	2	6	0.3	…	0.3

Tabla 8. Ejemplo del resultado de la extracción de las variables irrelevantes para el archivo de calidad del aire. Aplicable a la table de meteorología.

4.2.3. Limpieza de datos sobre tráfico

En nuestro fichero correspondiente a los datos de tráfico, podemos encontrar la variable FEST, que se encarga de identificar la estación que está recogiendo los datos. Como se menciona al principio de este apartado nosotros utilizaremos las estaciones más cercanas a la Plaza Elíptica, por lo que se procederá a la eliminación del resto de estaciones al terminar este apartado. También podemos encontrar la variable FSEN que ha sido explicada durante «sección 3.1.3.», Al igual que en los ficheros de calidad del aire y meteorología, también se dispone de una variable que identifica la fecha, los datos recogidos para cada hora (HOR1, …, HOR12) y estación, esta variable se denomina FDIA. «tabla 9.»

FDIA	FEST	FSEN	HOR1	HOR2
01/11/18	ES01	1-	808	721
01/11/18	ES01	1=	1134	1168
01/11/18	ES01	2-	735	589
01/11/18	ES01	2=	1154	1143

Tabla 9. Formato del fichero de Tráfico sin limpiar.

Para la limpieza de este fichero, se ha tenido que realizar un trabajo más elaborado en comparación con el resto del conjunto de datos. En primer lugar, se han borrado todas las claves nulas para posteriormente separar los datos en dos *dataframes* distintos. En uno se han almacenado los datos referentes a las horas de 1-12, que equivalen a los datos con el identificador «-», en otro se han incluido todos los datos de 13-24, que equivalen a los datos con el identificador «=». Ambos identificadores se encuentran en la columna FSEN. «tabla 9.»

A continuación, se muestra el resultado obtenido tras la división realizada para ambos conjuntos de horas. «tabla 10.» «tabla 11.»

FDIA	FEST	FSEN	HOR1	…	HOR12
1/11/18	ES01	1-	808	…	953
1/11/18	ES01	2-	735	…	1012
1/11/18	ES02	1-	528	…	703
1/11/18	ES02	2-	689	…	917

Tabla 10. Conjunto de datos de 1 a 12 para ambas direcciones.

FDIA	FEST	FSEN	HOR1	…	HOR12
1/11/18	ES01	1=	1134	…	588
1/11/18	ES01	2=	1154	…	509
1/11/18	ES02	1=	779	…	360
1/11/18	ES02	2=	1013	…	435

Tabla 11. Conjunto de datos de 13 a 24 para ambas direcciones.

En segundo lugar, se procederá a realizar una unión entre los dos *dataframes* para unir ambos conjuntos de datos correspondientes a las tablas anteriormente mencionadas. El resultado de realizar esta unión es un conjunto de datos unificado para veinticuatro horas. También se han modificado los nombres de las columnas generados automáticamente como resultado de dicha unión.

A continuación, se ha procedido a extraer el identificador del sentido del tráfico 1 ó 2 de la columna FSEN, generando una nueva columna llamada sentido con dicho identificador. Tras esta transformación se ha procedido a la eliminación de las columnas generadas con el nombre FSEN, dando lugar a la siguiente tabla. «tabla 12.»

FDIA	FEST	sentido	HOR1	…	HOR24
1/11/19	ES01	1	1527	…	348
1/11/19	ES01	2	1093	…	260
1/11/19	ES02	1	649	…	225
1/11/19	ES02	2	1167	…	402

Tabla 12. Ejemplo del resultado de la unión de los dataframes de 1a12 con 13a24.

Por último, se procedió a ordenar por FDIA, FEST y sumar ambas direcciones correspondientes a los valores 1 y 2 de la columna sentido que fue resultado de la operación anterior. «tabla 12.»

Como resultado final, se obtiene una fila por estación. Además, se ha procedido a eliminar la columna con el nombre sentido, debido a que esta ya carece de utilidad dentro del conjunto de datos. «tabla 13.»

FDIA	FEST	HOR1	…	HOR24
01/01/19	ES01	2116	…	608
01/01/19	ES02	1816	…	451

Tabla 13. Suma de ambas direcciones para el fichero de tráfico.

4.3. Homogeneización de los datos

En esta fase se ha procedido la modificación de los conjuntos de datos tratados durante el apartado anterior «sección 4.2.», con el objetivo de unificarlos en un único fichero a partir de la fecha y la hora. Para lograrlo es conveniente realizar una serie de transformaciones, las cuales se explicarán a continuación.

4.3.1. Datos sobre calidad del aire y meteorología

En nuestros ficheros, actualmente los parámetros de las horas vienen distribuidos en columnas diferentes asociados a una única columna de magnitud. Una de las características de los algoritmos de *Machine Learning* es la obtención de los datos fila a fila. Por esta razón, la distribución de los datos de los ficheros actuales -Tabla 14- tuvo que ser modificada, utilizando la función *melt*, distribuyendo de este modo las columnas (H01, …, H24) en una única. «tabla 15»

ANO	MES	DIA	MAGNITUD	H01	…	H24
2019	4	1	6	0.3	…	0.2
2019	4	2	6	0.3	…	0.3

Tabla 14. Datos de calidad del aire antes de realizar la transformación de la columna H1...H24.

ANO	MES	DIA	MAGNITUD	HORA	Value
2019	1	1	6	H01	0.6
2019	1	1	6	H02	0.8

Tabla 15. Ejemplo del resultado de la transformación de los datos de calidad del aire tras realizar la transformación de la columna H1...H24 en una sola columna.

A su vez, los datos correspondientes a la columna MAGNITUD, se tuvieron que distribuir en formato horizontal, utilizando para ello la función *pivot_table*. Obteniendo de este modo la relación deseada, correspondiente a fecha, hora y magnitud.

Por último, se procedió a la modificación de los valores correspondientes a la columna MAGNITUD, debido a la codificación numérica de los mismos, siendo poco clarificadores. Como resultado de estas modificaciones se obtuvo un fichero cuyo formato corresponde a la siguiente tabla. «tabla 16.»

DIA	MES	ANO	HORA	CO	NO	NO2
1	1	2019	H01	0.6	81.0	73.0
1	1	2019	H02	0.8	124.0	82.0
1	1	2019	H03	0.7	93.0	72.0

Tabla 16.Ejemplo del resultado de la transformación de columna a fila para las magnitudes de los contaminantes para el fichero de calidad del aire.

A continuación, se adjuntan los enlaces correspondientes a los datos de calidad del aire y los datos de meteorología.

- *https://github.com/albercol/TFG/blob/main/Fase2/Fase%202.1/DatosLimpios/Contaminacion.csv*
- *https://github.com/albercol/TFG/blob/main/Fase2/Fase%202.1/DatosLimpios/Meteorologia.csv*

4.3.2. Datos horarios sobre Tráfico

Para dar el formato adecuado al fichero correspondiente al conjunto de datos de Tráfico, se realizó el mismo proceso que en el apartado anterior «sección 4.3.1.» La única diferencia destacable, es el cambio en el nombre de las variables y la separación de la fecha en tres columnas diferentes para asemejarse al formato de los datos de los ficheros mencionados anteriormente. «tabla 17.»

DIA	MES	ANO	HORA	ES10	ES53
1	1	2019	H01	304	387
1	1	2019	H02	297	383

Tabla 17. Ejemplo del resultado de la transformación de fila a columna de las horas para el fichero de tráfico.

A continuación, se adjuntan los enlaces correspondientes a los datos de tráfico:

- *https://github.com/albercol/TFG/blob/main/Fase2/Fase%202.1/DatosLimpios/Trafico.csv*

4.3.3. Datos sobre festividades

Se ha procedido a renombrar las columnas para mantener la coherencia con el resto de los ficheros, cuyos nombres Día, Dia_semana, Laborable/festivo, Tipo de festivo y Festividad, han sido sustituidos por FECHA, DIA, TIPO, FESTIVIDAD, NOM_FESTIVIDAD.

Debido al cambio de formato mencionado anteriormente, se han puesto como laborables todos los días respetando los festivos. Posteriormente, con los datos de la columna DIA los coincidentes con el valor «sábado» y «domingo» se han modificado por dicho valor en la columna TIPO. Solucionando el problema del cambio de lectura de los datos. «tabla 18.»

FECHA	DIA	TIPO	FESTIVIDAD	NOM_FESTIVIDAD
01/01/2013	Martes	Festivo	Festivo nacional	Año Nuevo
02/01/2013	Miércoles	Laborable		

Tabla 18. Ejemplo del resultado de la transformación de la columna tipo para el fichero festividades.

5. ESTUDIO DE LOS DATOS

Durante este capítulo se pretende realizar una limpieza de los datos y quitar en la medida de lo posible valores nulos y *outliers* para finalmente obtener un conjunto de datos unificado y listo para ser utilizado, alcanzando así los objetivos propuestos. «O2» y «O3»

En el siguiente enlace se muestra el fichero con los datos unificados tras la detección de *outliers*:

- *https://github.com/albercol/TFG/tree/main/Fase2/Fase%202.3/DatosUnificados*

5.1. Valores nulos

En este apartado se pretende analizar la cantidad de datos nulos que contienen nuestros conjuntos de datos, para determinar qué hacer en cada caso.

No se da la misma situación en todos los casos, por ejemplo, el conjunto de metodología presenta una cantidad ínfima de valores nulos, al igual que sucede con el conjunto de datos de tráfico, los cuales no contienen ningún valor nulo. Por consiguiente, durante este apartado se tratará exclusivamente el fichero de calidad del aire. «figura 4.»

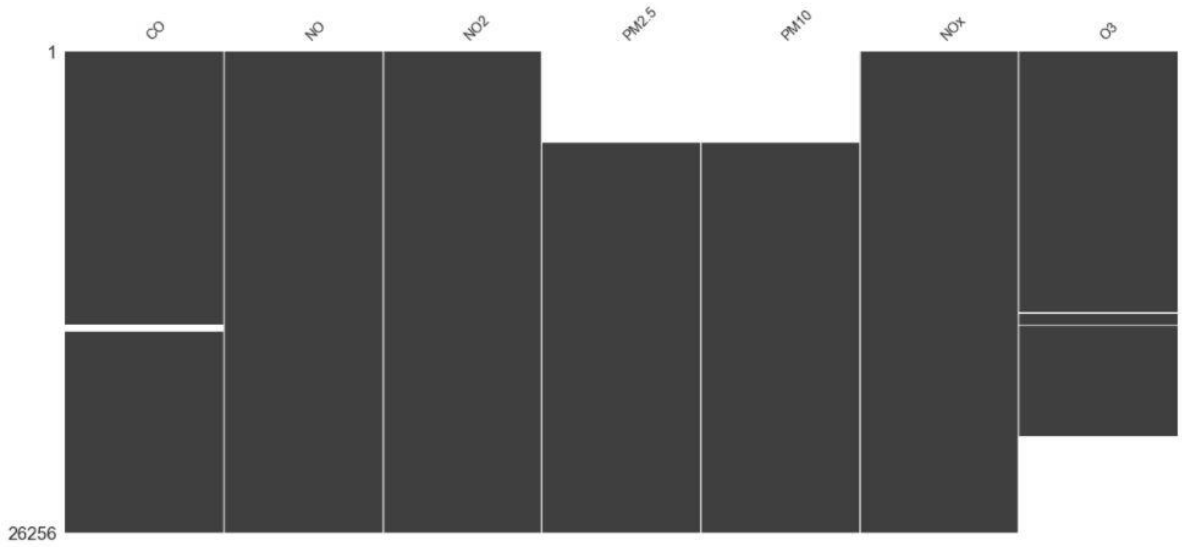


Figura 4. Representación gráfica de valores nulos para todo el conjunto de los elementos de calidad del aire. El eje de ordenadas muestra la cantidad de filas, en el de abscisas cada uno de los contaminantes que tratamos durante este trabajo.

Fuente: elaboración propia mediante librería Seaborn de Python.

En primer lugar, se ha realizado una representación gráfica para determinar el número de datos nulos «Nans» del que se disponen. Se puede observar una cantidad considerable de estos datos en las partículas PM2.5, PM10 y O3, quizás por fallo de los sensores o paradas de mantenimiento. Además, al ser datos que proceden de dispositivos de medición, estos pueden registrar valores anómalos, demasiado elevados o bajos, que pueden influir en las predicciones, los cuales se tratarán durante el siguiente apartado.

Si las filas que tuvieran valores nulos en el conjunto de los datos fueran eliminadas de este, obtendríamos un conjunto de datos muy pequeño, eliminando a su vez datos correctos de otros contaminantes. Por esta razón, se ha decidido seguir trabajando con ellos a pesar de los nuevos generados, eliminándolos en fases futuras cuando se vayan a tratar.

5.1.1 Valores cero

Al tratarse de sensores que miden la densidad de las partículas en suspensión, estos toman exclusivamente valores positivos, por lo que se ha llegado a la conclusión de que los valores detectados como ceros absolutos actúan como fondos planos, que además de suponer unos valores irreales no aportan información relevante al modelo. Por este motivo se decidió transformar estos datos en valores nulos.

Si tenemos en cuenta la cantidad de datos cuyo valor es «0» junto al conjunto de los nulos, encontramos una cantidad considerable de datos que tendríamos que rechazar, por lo que al igual que se explicó durante el apartado «sección 5.1.», se tomó la decisión de eliminarlos cuando tengamos que procesarlos en fases futuras.

5.2. Outliers

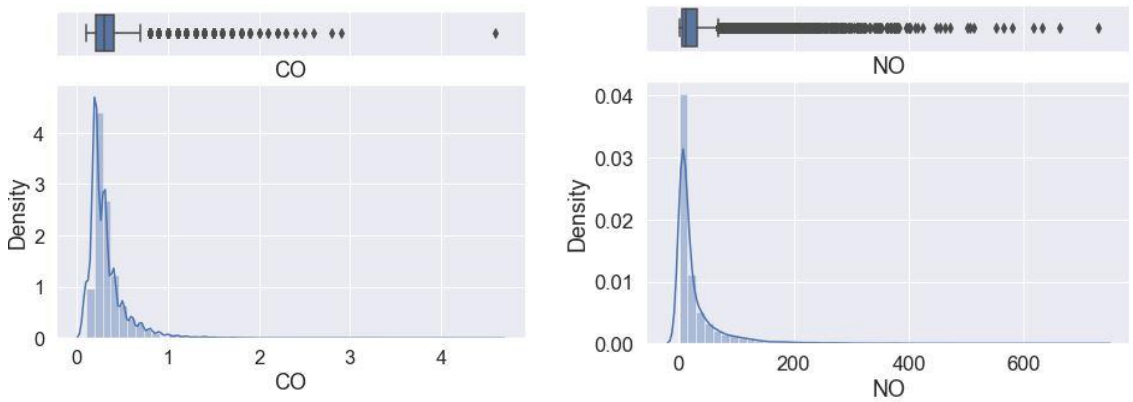
En este apartado se pretende detectar valores atípicos para posteriormente eliminarlos del conjunto de datos, y que estos no afecten los resultados estadísticos ni al entrenamiento de nuestro modelo. (Navas, 2020)

A efectos de este trabajo, podemos encontrar dos tipos de *outliers*. En primer lugar, podemos encontrar valores altos o bajos pero que se corresponden con la realidad, estos suelen considerarse valores cuya distancia supera en más de tres veces el *mean absolute deviation* de la mediana (B., 2014). En segundo lugar, podemos encontrar valores anómalos debido a errores. Estos últimos serán los que se tratarán durante la realización de este apartado para el fichero de calidad del aire, marcando y eliminando los valores que se encuentran en más de quince veces el *mean absolute deviation*, que corresponde a multiplicar el valor considerado para primer caso por cinco. Para el resto se ha usado el valor estándar del primer caso, debido a que no encontraban valores anómalos.

Para la detección de *outliers* es habitual la utilización de la mediana y su *mad,* en lugar de la media y la desviación típica, debido a que la mediana es más robusta ante la presencia de *outliers*. (Ripley, 2004)

5.2.1. Datos sobre calidad del aire

Si realizamos una representación gráfica mediante histogramas para cada uno de los contaminantes recogidos por la estación, como se muestra en la siguiente imagen «figura 5.» se observa la obtención de una función normal desplazada hacia la izquierda en la mayoría de los casos, excepto en el Ozono, que obtiene una función bimodal, debido a que este contaminante aumenta conforme lo hace la temperatura, lo cual tiene relación directa entre los días y las noches, aunque también esta última se encuentra desplazada hacia la izquierda.



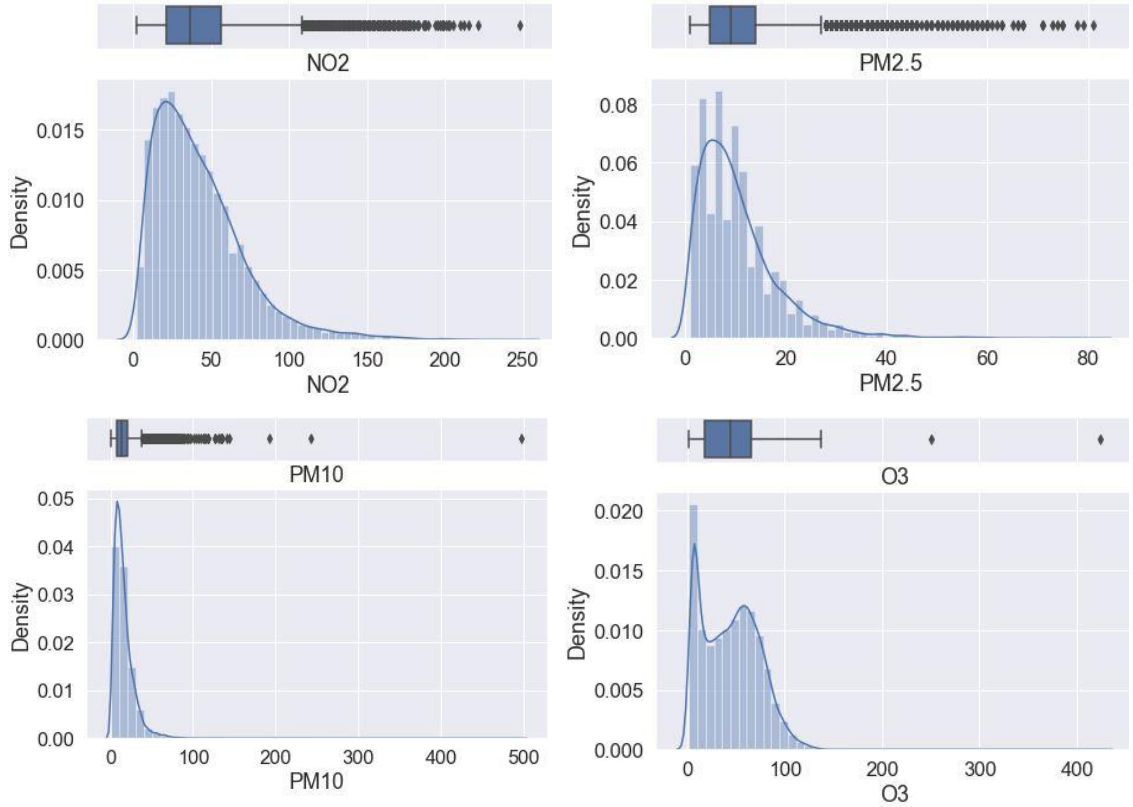


Figura 5. Histogramas para cada contaminante junto con su boxplot asociado, antes de quitar outiliers.

Fuente: elaboración propia mediante librería Matplotlib de Python.

Por lo tanto, a la hora de eliminar los valores atípicos se ha tenido que realizar con cuidado para no eliminar variaciones debidas a acontecimientos meteorológicos o situaciones anómalas que se hayan podido producir.

Como se tenía la sospecha de que los contaminantes se comportarían de forma diferente según el tráfico, se ha procedido a añadir para cada fecha un identificador de tipo día: laborable, sábado, domingo y festivos. De este modo se ha podido filtrar por cada uno de ellos y así obtener los máximos diarios, la mediana y *mean absolute deviation (mad)* de esos máximos, y marcar como *outliers* los valores cuya distancia a la mediana supera en una determinada cantidad de veces el *mad*. Como se ha especificado al principio de este apartado, este valor será de quince debido a las características del conjunto de datos y la necesidad de eliminar valores erróneos como consecuencia de fallos durante la recopilación de los datos.

En este caso, como se puede observar en las gráficas anteriores se representa un conjunto de funciones con una desviación hacia la izquierda, por lo que, si se dejase el factor de

multiplicación utilizado para una función normal estaríamos eliminando un exceso de valores válidos.

De esta forma, se consiguió reducir ligeramente la cantidad de valores atípicos, sin afectar a una cantidad muy grande del conjunto de datos.

A continuación, se muestra un ejemplo de dos histogramas junto a sus *boxplots*, comparando ambas gráficas tras la detección y eliminación de *outliers* para el contaminante CO. «figura 6.» Además, un ejemplo gráfico que permite observar estos valores para el contaminante PM10 muestran a lo largo del tiempo los valores que va tomando esta variable, mostrando los *outliers* detectados con un punto verde. «figura 7.» «figura 8.»

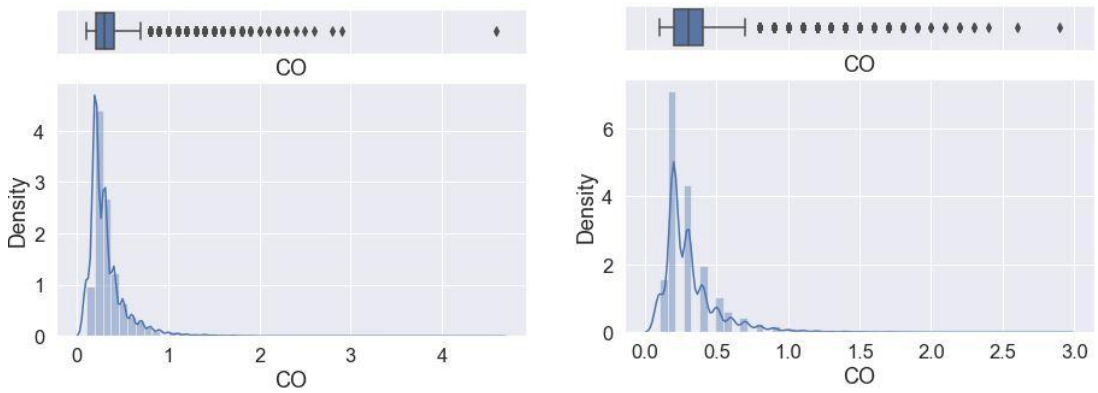


Figura 6. Comparativa del histograma para el contaminante CO, antes y después de quitar los outliers.

Fuente: elaboración propia mediante librería Matplotlib de Python.

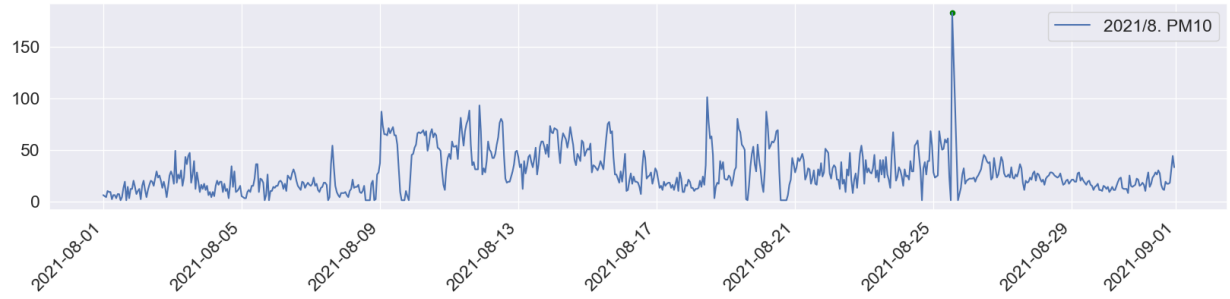


Figura 7. Detección de outliers para el PM10, correspondiente a un día laborable a lo largo del mes de agosto del 2021.

Fuente: elaboración propia mediante librería Matplotlib de Python.

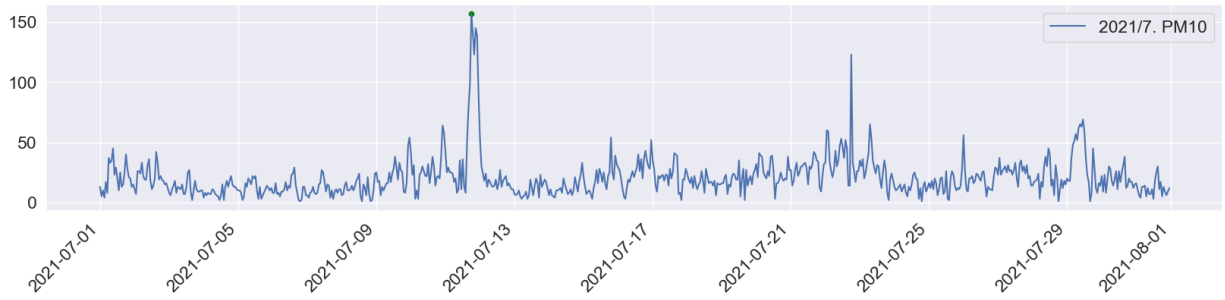
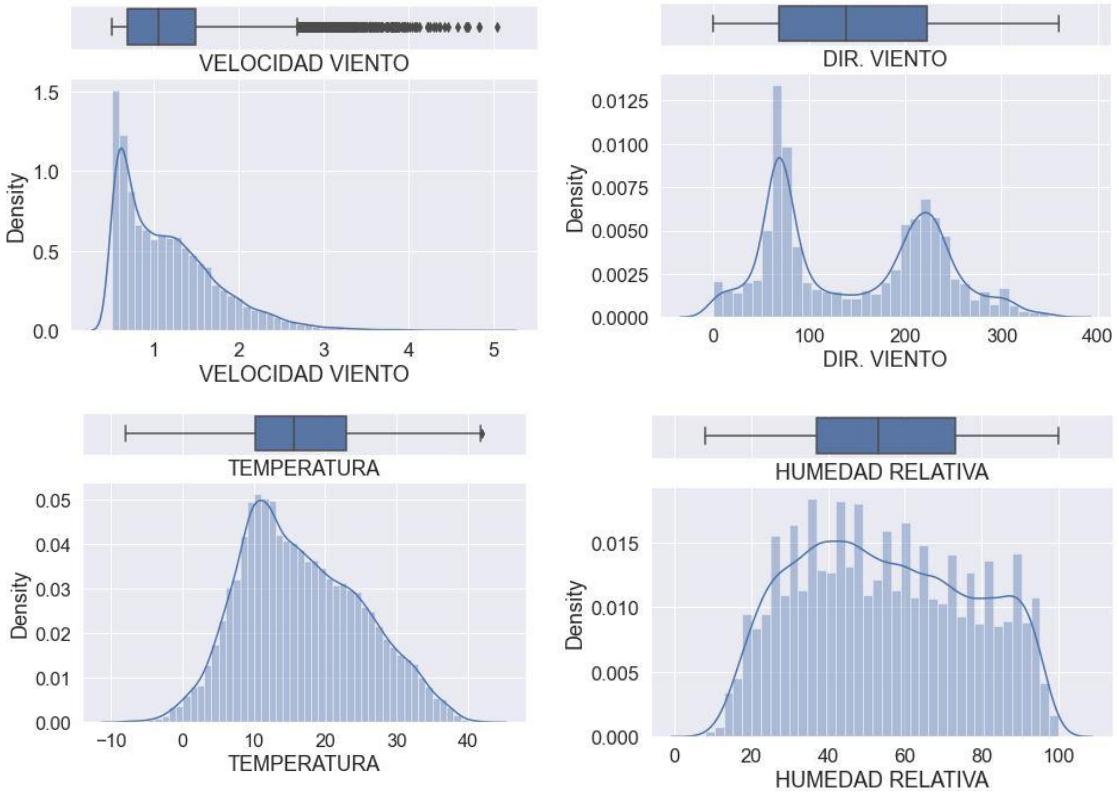


Figura 8. Detección de outliers para el PM10, correspondiente a un domingo a lo largo del mes de julio del 2021.

Fuente: elaboración propia mediante librería Matplotlib de Python.

5.2.2. Datos meteorológicos

Si realizamos una representación gráfica mediante histogramas, para cada una de las variables meteorológicas recogidas por la estación «figura 9.», obtenemos una función normal, asimétrica y desplazada hacia la izquierda en la variable VELOCIDAD DEL VIENTO, en el resto de las variables meteorológicas, podemos observar que tienen una distribución normal, excepto en las variables HUMEDAD RELATIVA que presenta una asimetría considerable, y en la variable DIR. VIENTO que obtenemos una función bimodal muy pronunciada.



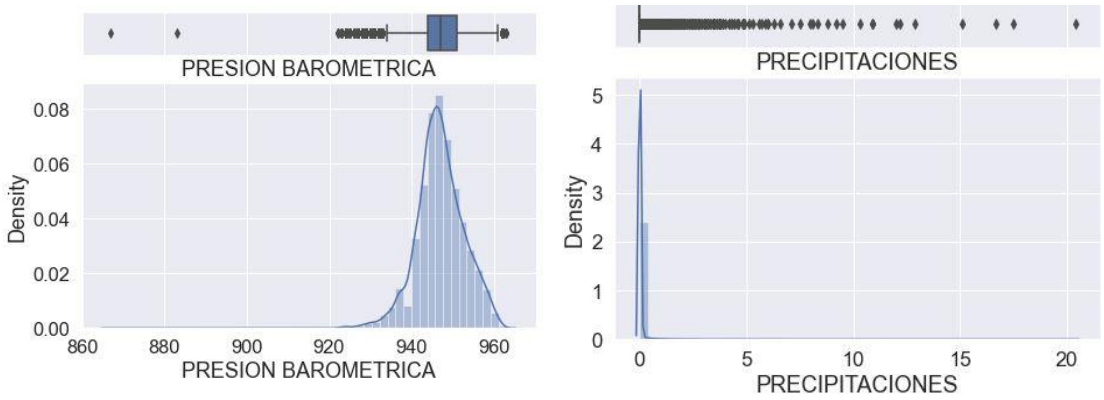


Figura 9. Histogramas para cada variable meteorológica junto con su boxplot asociado.

Fuente: elaboración propia mediante librería Matplotlib de Python.

Esta última, se produce debido a la relativa cercanía de la Sierra de Guadarrama respecto a Madrid, generándose vientos del Oeste en condiciones no estables y brisas en condiciones estables, soplando en una dirección por la noche y en otra por el día. (C.Román-Cascón, s.f.) Aunque en el caso de Madrid, estas brisas se reorientan dependiendo de la topografía local, pero su dirección predominante viene determinada por el diferente calentamiento de las zonas más altas y el llano durante el día. Además, los datos obtenidos en las mediciones también pueden estar condicionados debido a la reorientación producida por los edificios, especialmente si el punto de muestreo se encuentra muy próximo al suelo, como resulta en este caso. «figura 10.»

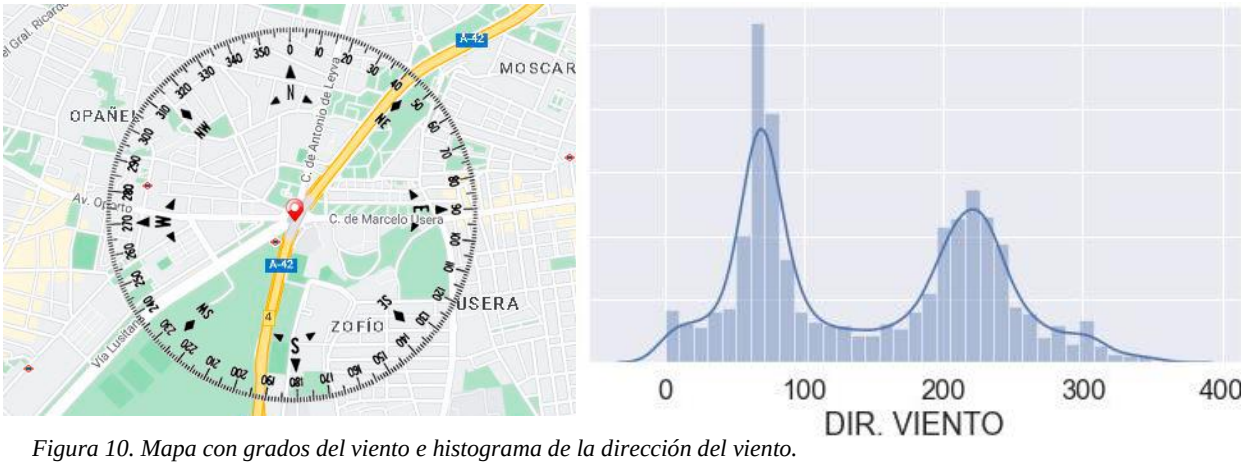


Figura 10. Mapa con grados del viento e histograma de la dirección del viento.

Fuente: mapa de Google con la representación de los grados.

A la hora de eliminar los *outliers*, se ha decidido proceder de manera similar al fichero referente a los datos de calidad del aire, con la única diferencia de que los valores cuya distancia a la mediana superan en tres la del *mad*. «sección 5.2.1.»

Además, se ha omitido la columna PRECIPITACIONES debido a sus características particulares.

5.2.3. Datos sobre Tráfico

Tras hacer una representación mediante histogramas de los datos de las tres estaciones seleccionadas, podemos observar en las siguientes gráficas «figura 11.» la existencia de una clara relación entre las estaciones ES53 y ES54, correspondientes a la calle Marcelo Usera y la calle Av. Rafaela Ybarra, por lo que se concluyó que para dicha localización, las dos estaciones situadas al este de Plaza Elíptica absorben el tráfico una de la otra «figura 12.», por lo que se podrían unificar estas dos estaciones y tener la suma de ambas, debido a la semejanza en ambos histogramas.

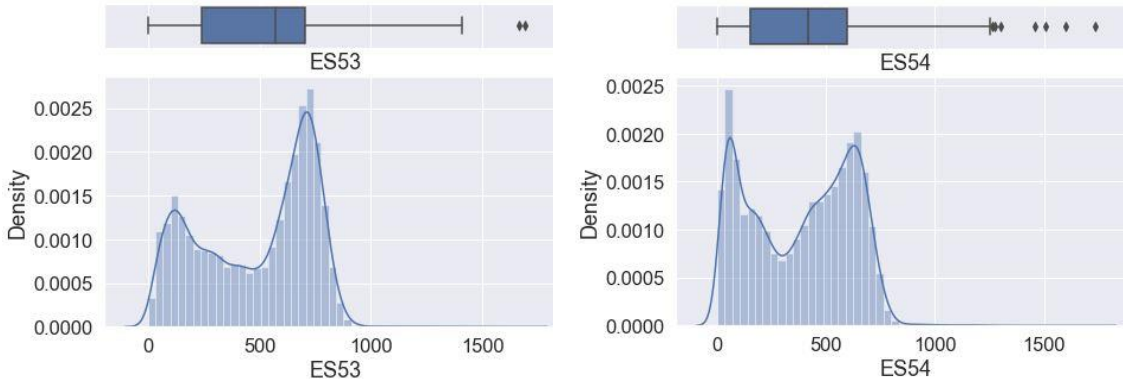


Figura 11. Representación gráfica de outliers para las estaciones de tráfico ES53 y ES54.

Fuente: elaboración propia mediante librería Matplotlib de Python.

Figura 12. Localización de estaciones de tráfico.

Fuente: Ayuntamiento de Madrid. Aforos de tráfico disponible en datos.madrid.es

En este caso, para eliminar los *outliers* se ha procedido de una manera similar a la sección de datos meteorológicos «sección 5.2.2.»

5.3. Estacionariedad de los datos

Disponemos de un conjunto de datos que se recopilan secuencialmente en el tiempo formando una serie temporal**.** Durante este apartado se pretende examinar la estacionalidad de los datos, tratados como series temporales, mediante los métodos estadísticos, como la prueba aumentada de *Dickey-Fuller* y *Rolling Statistic*.

Una serie temporal es estacionaria, cuando resulta estable a largo plazo en el tiempo, esto quiere decir que los valores de esta tienden a oscilar alrededor de una media constante en el tiempo, sin que se aprecien aumentos o disminuciones sistemáticos en sus valores.

Por otro lado, las series temporales no estacionarias, son series las cuales cambian en el tiempo, estos cambios marcan una tendencia clara a crecer o decrecer a lo largo de toda la serie temporal, por contraposición a lo mencionado en el párrafo anterior, esta no oscila alrededor de un valor constante.

La implicación de que una serie temporal sea estacionaria o no estacionaria es la facilidad de modelado, puesto que las series temporales no estacionarias deben tratarse para la identificación y eliminación de la tendencia de esta, además de la eliminación de los efectos estacionales. (Brownlee, 2016) (Nishtha, 2021)

5.3.1. Prueba aumentada de Dickey-Fuller

La prueba aumentada de *Dickey-Fuller*, es un tipo de prueba estadística llamada prueba de raíz unitaria cuyo objetivo es determinar qué tan fuerte está definida una serie temporal por una tendencia. Esta prueba se ha realizado para cada una de las variables seleccionadas para este proyecto.

La hipótesis nula de la prueba es que la serie temporal puede ser representada por una raíz unitaria, eso quiere decir que dicha serie no es estacionaria. Por el contrario, la hipótesis alternativa es que la serie es estacionaria.

Interpretamos este resultado utilizando el p-value de la prueba. Si este valor está por debajo de un umbral del 5%, sugiere que rechazamos la hipótesis nula, lo que querría decir que la serie temporal es estacionaria, de lo contrario, un valor por encima de este umbral nos sugeriría que no rechazamos la hipótesis nula por lo que dicha serie sería no estacionaria.

Para: **CO** ADF Statistic: -8.768627 p-value: 2.5567708522140714e-14 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **NO2** ADF Statistic: -10.264429 p-value: 4.1548809304732636e-18 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **PM2.5** ADF Statistic: -8.970265 p-value: 7.788156002450207e-15 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567
Para: **O3** ADF Statistic: -7.852976 p-value: 5.537579246293036e-12 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **VELOCIDAD VIENTO** ADF Statistic: -11.303394 p-value: 1.2908663610363708e-20 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **DIR. VIENTO** ADF Statistic: -9.838135 p-value: 4.833494648182885e-17 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567
Para: **TEMPERATURA** ADF Statistic: -3.493716 p-value: 0.008149965517626399 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **HUMEDAD RELATIVA** ADF Statistic: -5.501909 p-value: 2.0622289782079447e-06 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **PRESION BAROMETRICA** ADF Statistic: -8.602765 p-value: 6.79850677364883e-14 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567
Para: **PRECIPITACIONES** ADF Statistic: -37.067117 p-value: 0.0 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **ES10** ADF Statistic: -7.296702 p-value: 1.3719796313738182e-10 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567	Para: **ES53** ADF Statistic: -4.691919 p-value: 8.714383779887315e-05 Critical Values: 1%: -3.431 5%: -2.862 10%: -2.567

Tabla 19. Resultados de realizar la prueba aumentada de Dickey-Fuller para todos los contaminantes.

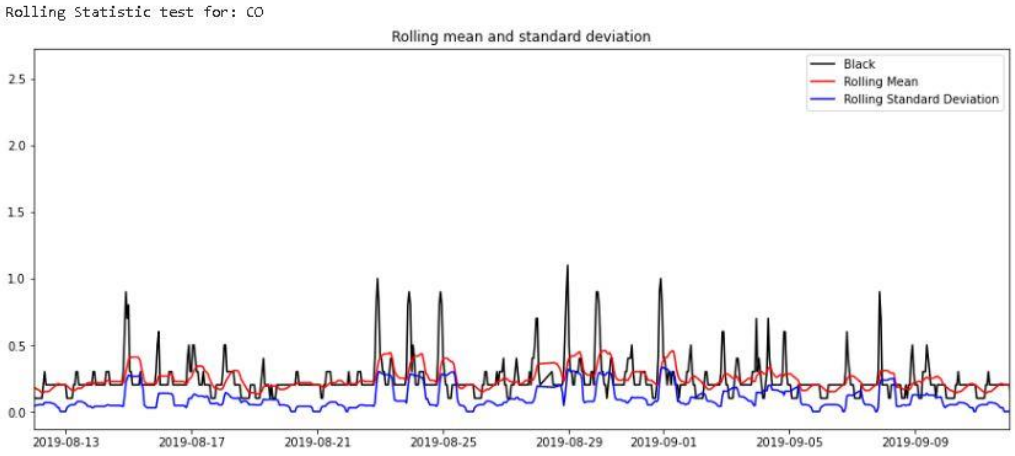
Tras realizar la prueba «tabla 19.», se imprime el valor de la variable ADF Statistic negativo en todas las variables que estamos utilizando, cuanto más negativo sea esta estadística, más probable es de que rechacemos la hipótesis nula por lo que podríamos concluir que nuestro conjunto de datos es estacionario.

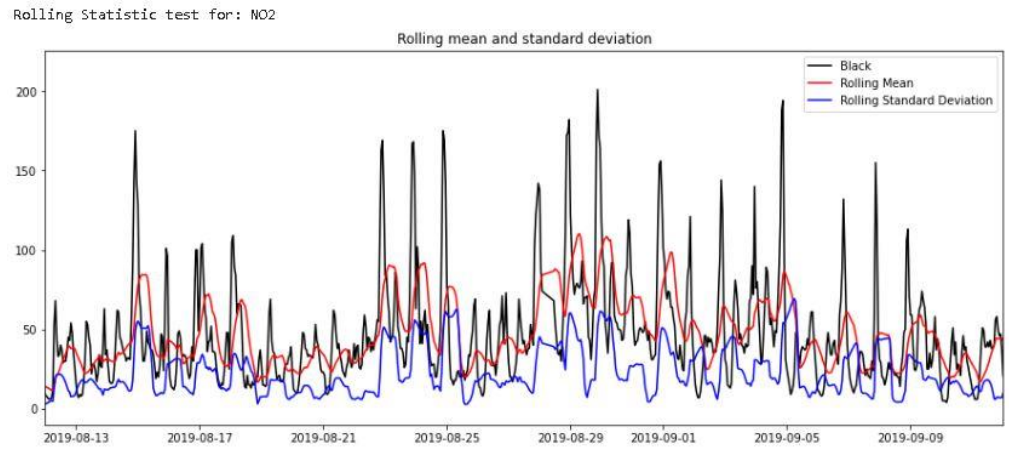
Además, obtenemos una tabla de búsqueda para ayudar a determinar este valor 1%, 5% y 10% de significancia. Podemos observar que nuestra variable ADF Statistic es menor que el -3,431 en 1%. Lo que también nos sugiere que podemos rechazar la hipótesis nula con un nivel de significancia de menos del 1%, esto quiere decir que hay una baja posibilidad de que este resultado haya sido casual.

Por último, como se ha mencionado al principio de este apartado, podemos observar que nuestra variable p-value está por debajo del umbral del 5% lo que corrobora la decisión del rechazo de la hipótesis nula y confirmamos que nuestro conjunto de datos es estacionario.

5.3.2. Rolling Statistic

Esta prueba, resulta interesante para confirmar los resultados obtenidos en el apartado anterior, proporcionando una representación visual para el conjunto de datos.





(C.R

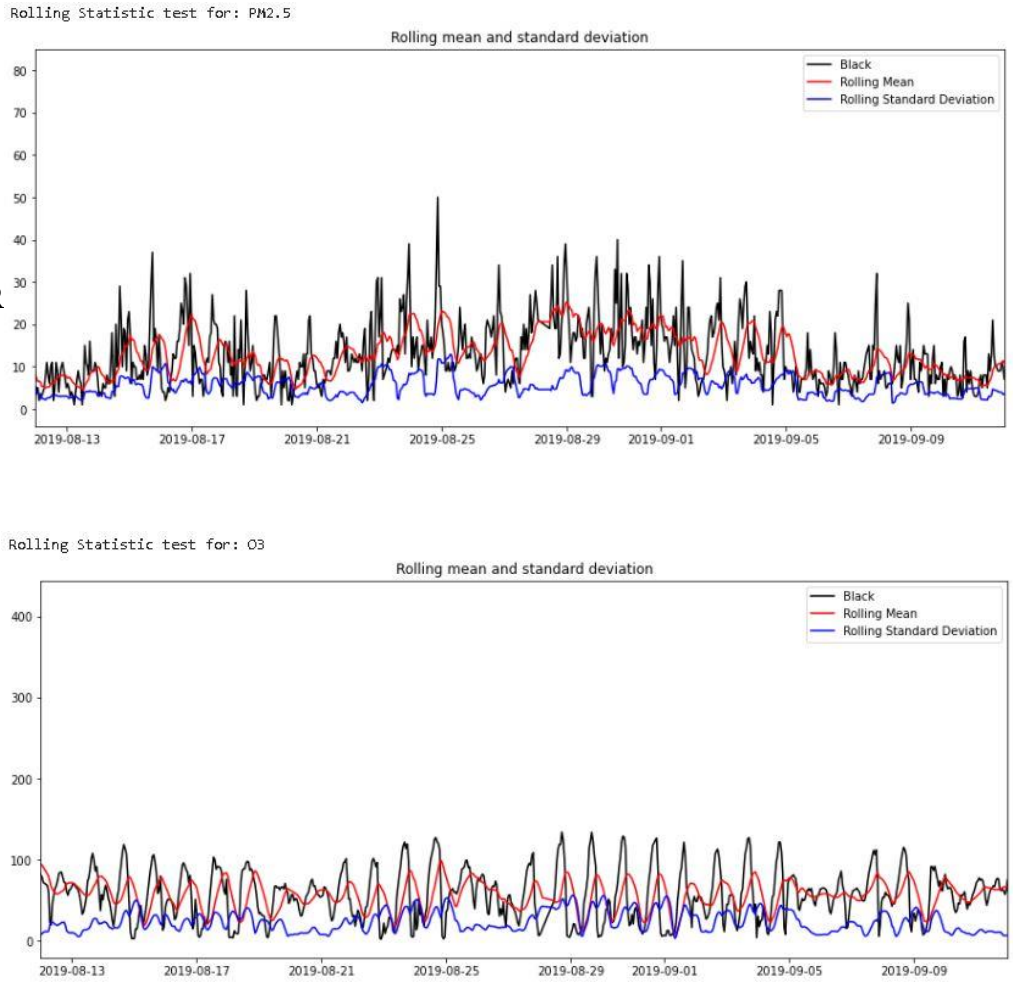


Figura 13. Representación gráfica de la prueba de Rolling statistic, donde Black representa los datos sin tratar, y donde Rolling Mean y Rolling Standar Deviantion corresponden a los valores con la desviación típica y la media.

Fuente: elaboración propia mediante librería Maplotlib de Python.

Como podemos observar «figura 13.». Tras realizar la representación gráfica de las variables relacionadas con la calidad del aire, podemos observar que la *Rolling Mean* y *Rolling Standard Deviation* son constantes, esto confirma lo que ya sabíamos con anterioridad. Podemos afirmar mediante dicha representación, que nuestro conjunto de datos es estacionario.

6 ANÁLISIS DE DATOS Y EXTRACCIÓN DE INFORMACIÓN

A lo largo de este capítulo se estudiarán las relaciones que existen entre todos los elementos que conforman el conjunto de datos. Además, se explicarán los algoritmos utilizados para la creación y selección de modelos, y cuáles de las técnicas disponibles se pueden utilizar para mejorarlos, pudiendo así predecir el comportamiento de los elementos contaminantes y extraer conclusiones acerca de la viabilidad de realizar dichas predicciones. Tras la finalización de este capítulo, se habrán alcanzado los objeticos propuestos. «O4» «O5»

6.1. Correlaciones

Durante este apartado, se pretende averiguar las relaciones existentes para las distintas variables que corresponden a cada conjunto de datos, como entre todo el conjunto de variables disponibles para la realización de este proyecto.

Se utilizará el método de correlación de Pearson mediante la librería de *Python*, *Numpy*. Este coeficiente mide la asociación lineal entre variables.

El conjunto de datos unificado está disponible en el repositorio de **Github**.

6.1.1. Datos sobre calidad del aire

Se ha procedido a eliminar todos los valores nulos que lo componen debido a que estos pueden afectar a los resultados de las correlaciones. Además, se ha filtrado para obtener exclusivamente los elementos químicos contaminantes, quedándonos con las variables, CO, NO, NO2, PM2.5, PM10, NOx y O3.

Se puede observar en la gráfica «figura 14.», hay una correlación muy fuerte entre NO y NOx, puesto que pertenecen a los óxidos de nitrógeno, cuya composición es similar, por lo que cuando cualquiera de estos se incrementa también lo harán sus homólogos.

También se ha observado una alta correlación entre el CO, NO, NO2 y NOx. Una posible explicación a su relación podría ser la generación de estos gases, cuyo origen están en la

quema de combustibles fósiles relacionados a su vez con el tráfico rodado. A su vez, estos contaminantes dependen de la intensidad de las turbulencias y la estabilidad atmosférica de modo que, con condiciones estables, todos los contaminantes tienden a aumentar y por consiguiente las correlaciones entre los ellos.

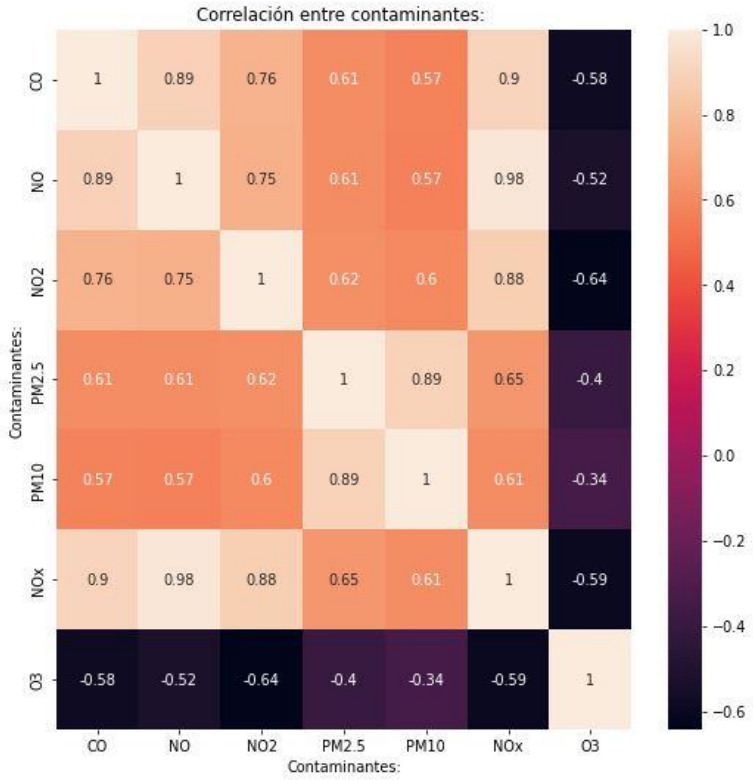


Figura 14. Correlaciones entre contaminantes.

Fuente: elaboración propia mediante Python.

Además, existe una alta correlación entre las partículas de PM2.5, Partículas menores 2.5, y las partículas PM10, partículas menores de 10, la explicación para este fenómeno está en que las PM10 engloban a las PM2.5.

Llama la atención la correlación negativa que existe entre el Ozono y el resto de los contaminantes, especialmente el NO_x. La explicación a esta correlación negativa se realizará durante el siguiente apartado «sección 6.1.1.1.».

Para evitar la redundancia en los datos, se ha tomado la decisión de omitir para el entrenamiento de los modelos las variables con alta correlaciones entre sí, como, por ejemplo, NOx y NO, o las partículas PM10, debido a que estas son las menos perjudiciales para la salud.

6.1.1.1 Correlación negativa entre el NO_x y O_3

Para poder interpretar esta correlación negativa, se han planteado una serie de hipótesis tomando datos de 2017, representados de forma más visual en las siguientes gráficas. «figura 15.» «figura 16.»

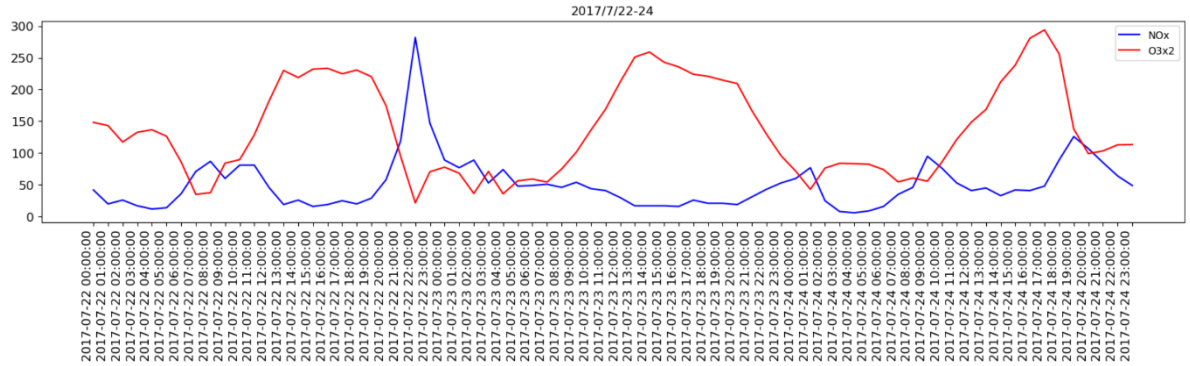


Figura 15. Representación temporal del mes de julio de los contaminantes NOx y O3 multiplicado por dos, para mejorar la observación.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

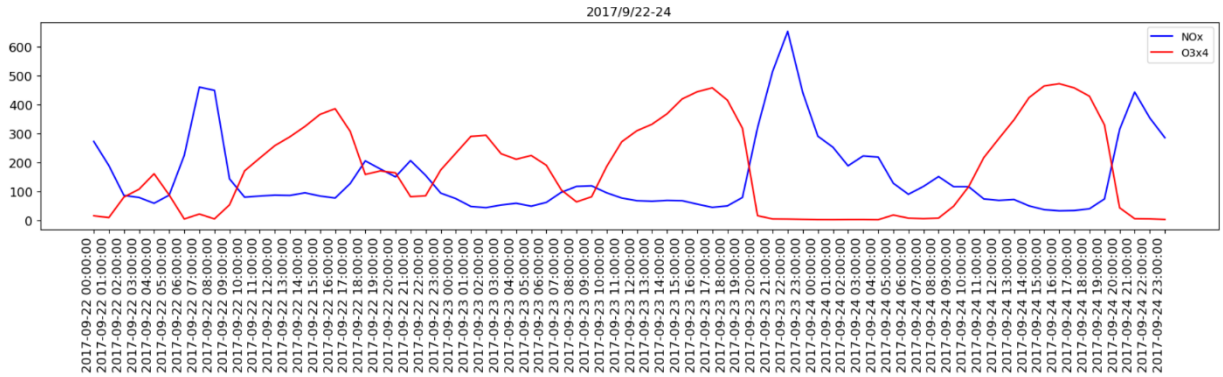


Figura 16. Representación temporal del mes de septiembre de los contaminantes NOx y O3 multiplicado por cuatro, para mejorar la observación.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

Dichas gráficas reflejan el comportamiento espejo que presentan ambos contaminantes, donde se puede observar que cada magnitud reacciona de forma opuesta con cierta asimetría a lo largo de las horas.

Este comportamiento puede deberse a las diferentes reacciones y dependencia que presentan ambos contaminantes con las condiciones ambientales, tales como la radiación solar, que junto al NOx favorece la generación de O_3 troposférico y a su vez coincidiendo con el verano, disminuye la presencia de NOx por a la generación de turbulencias debido a la época del año. (Querol Carceller, 2018)

Asimismo, se estudia esta presencia del O_3, obteniendo los siguientes histogramas de densidad, donde el primer pico en valores bajos en torno a veintitrés corresponde al invierno y el segundo pico en torno a sesenta, corresponden a verano. «figura 17.»

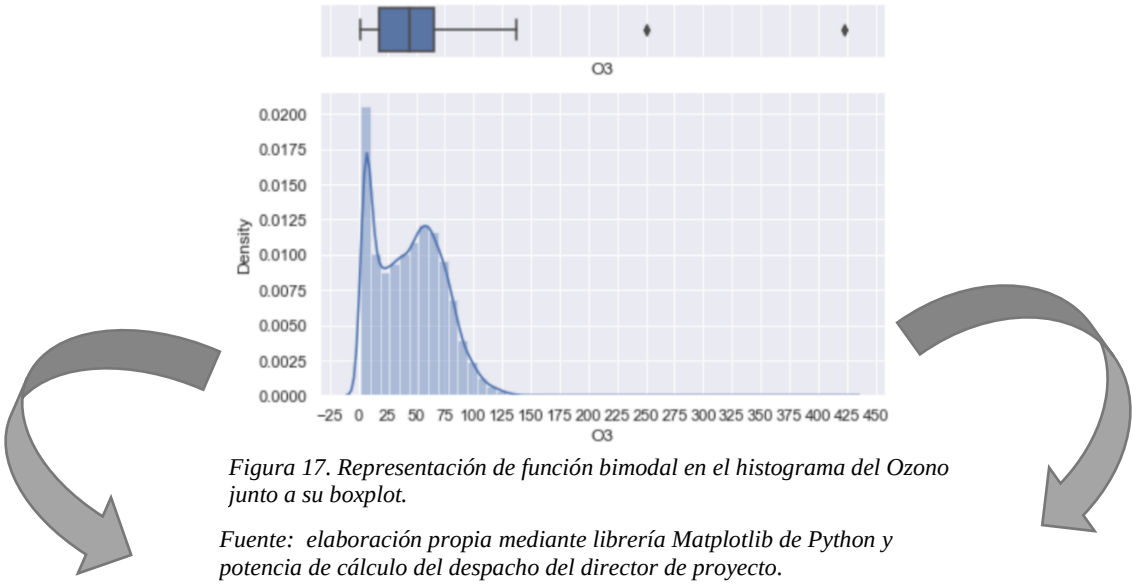


Figura 17. Representación de función bimodal en el histograma del Ozono junto a su boxplot.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

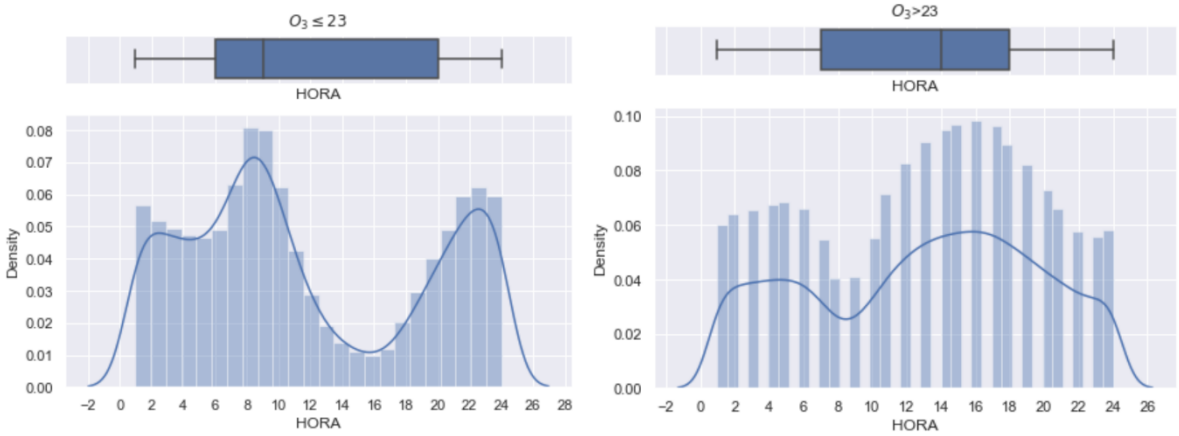


Figura 18. División de la función bimodal del Ozono, para valores menores o iguales y mayores de veintitrés, debido a que corresponde al mínimo del primer histograma.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

Para poder dar una explicación a estos resultados se plantean varias hipótesis. La primera, es que el valor máximo de la gráfica «figura 17» correspondiente a O_3<23, centrado principalmente en invierno, puede deberse a la estabilidad invernal, producida por las inversiones térmicas superficiales.

La segunda hipótesis que se plantea sería que el pico producido en la gráfica «figura 17» correspondiente O_3>23 se produce en la primavera tardía. Esto es debido a la radiación ultravioleta en las horas de sol, produciéndose mayores concentraciones de este compuesto.

Estas dos hipótesis corresponden a dos fenómenos separados, por esta razón se podría explicar el mínimo en la función bimodal, debido a la diferencia de valores.

Posteriormente, se han dividido los datos en dos histogramas diferentes $O_3 \leq 23$ y $O_3 > 23$ «figura 18», pudiéndose apreciar que nuevamente se representan dos funciones bimodales, donde cada una alcanza los máximos en horas distintas. En la gráfica $O_3 \leq 23$ la forma obtenida en la función es fruto de la estabilidad producida por el propio ciclo diurno condicionada por las emisiones de los vehículos. Sin embargo, en la gráfica $O_3 > 23$ correspondería a los meses de verano, generando mayor cantidad de O_3 durante las horas de más radiación ultravioleta

6.1.2. Datos meteorológicos

Al igual que en el apartado anterior, se ha procedido a eliminar todos los valores nulos que lo componen debido a que estos pueden afectar a los resultados de las correlaciones. Además, se ha filtrado para obtener exclusivamente las variables recogidas en el fichero sobre meteorología, quedándonos con las variables, VELOCIDAD DEL VIENTO, DIR. VIENTO, TEMPERATURA, HUMEDAD RELATIVA, PRESIÓN BAROMÉTRICA Y PRECIPITACIONES.

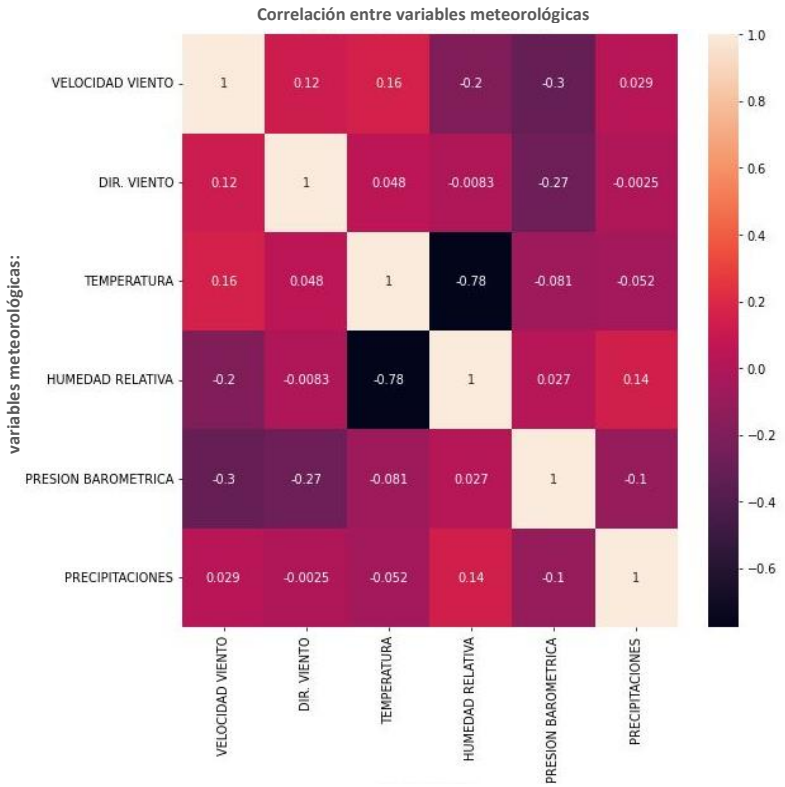


Figura 19. Correlaciones entre variables meteorológicas.
Fuente: elaboración propia mediante Python

Se puede observar en la gráfica «figura 19.», una considerable correlación negativa entre la TEMPERATURA y la HUMEDAD RELATIVA. La temperatura, define la presión de saturación de vapor de agua, cualquier pequeño cambio producido en el valor de la temperatura, principalmente en altas humedades, tiene un efecto significativo en la humedad relativa, ya que la presión de saturación de vapor de agua también cambia.

De este modo, se puede explicar la relación inversa entre estas dos variables meteorológicas.

6.1.3. Datos sobre Tráfico

Se ha procedido a eliminar todas las filas nulas, para que no afecten a los resultados de las correlaciones, posteriormente se ha filtrado el conjunto de datos por las tres estaciones seleccionadas.

Originalmente existe una alta correlación entre todas las testaciones de tráfico, especialmente entre las estaciones ES53 y ES54 las cuales se encuentran muy próximas entre sí al este de Plaza Elíptica, como se puede comprobar en el apartado *outliers* «sección 5.2.3.» corresponden a las calles de Marcelo Usera y Avenida de Rafaela Ibarra, por lo que se optó por eliminar la estación ES54 debido a que una de las calles recibía la mayor parte del tráfico de la otra, y sumar ambas estaciones generando una nueva variable ESSUMA, dando como resultado la siguiente gráfica. «figura 20.»

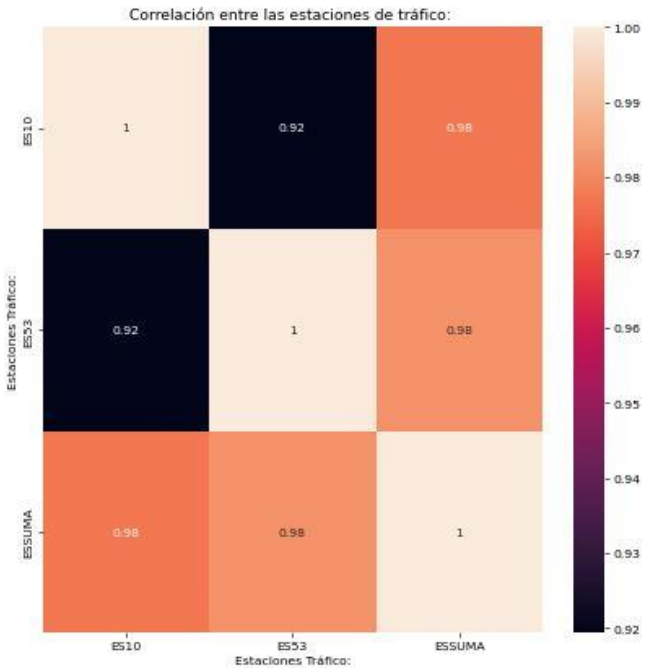


Figura 20. Correlaciones entre estaciones de tráfico.

Fuente: elaboración propia mediante Python.

6.1.4. Correlación con todas las variables

Tras la realización del estudio de las correlaciones entre las distintas variables, no se pudieron sacar conclusiones directas entre la meteorología con el tráfico, debido a que este mide la cantidad de coches que pasan en una hora y no la densidad, congestiones, accidentes, etc.

Asimismo, al observar la relación sobre la meteorología y la calidad del aire, destaca la correlación negativa que existe entre el viento y el resto de los contaminantes, excluyendo el Ozono, esto se puede explicar por el efecto difusivo que ejerce sobre los mismos. Sin embargo, el Ozono, guarda una relación mucho más importante con la radiación ultravioleta, con una marcada componente horaria y estacional, por lo que el efecto del viendo queda enmascarado. Por consiguiente, si se tiene en cuenta la cantidad de radiación ultravioleta producida en invierno y las inversiones térmicas durante esta estación, haciendo que el viento sea más débil a causa del bloqueo que estas generan, se explica la relación directa que existe entre el O_3 y la velocidad del viento. «figura 21.»

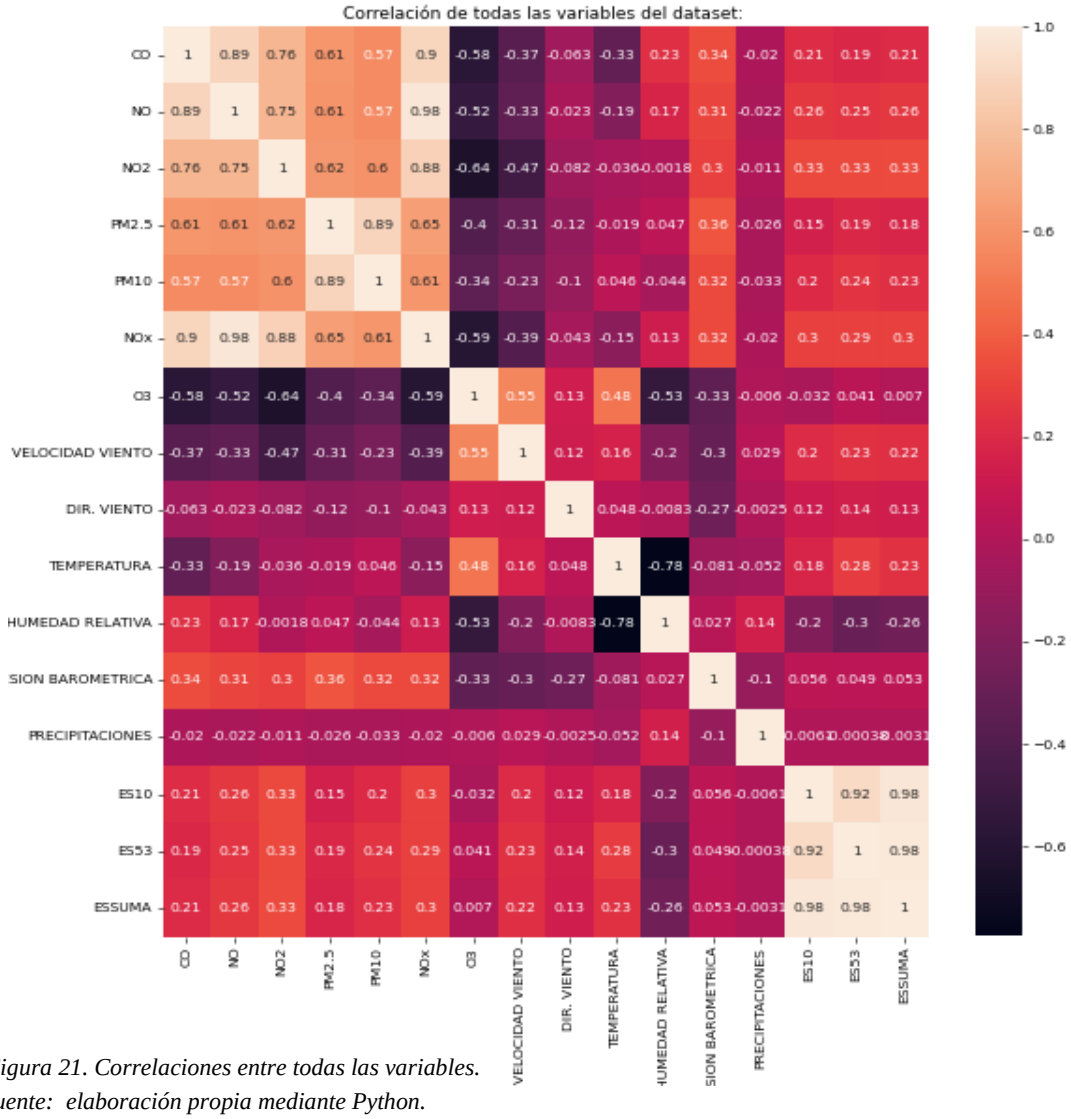


Figura 21. Correlaciones entre todas las variables.
Fuente: elaboración propia mediante Python.

Como conclusión podemos observar que existe una ligera relación entre los distintos componentes como el tráfico o la meteorología con la contaminación, este comportamiento es debido a que las correlaciones miden la relación directa entre cada una de ellas independientemente del resto de variables por lo que el resultado es que de forma independiente la mayoría de variables no ejercen una gran influencia en la contaminación ambiental, a excepción del viento, que resulta clave para determinar la concentración de los mismos.

Como veremos más adelante, se espera que un modelo de *Machine Learning* consiga ver las relaciones entre las diferentes variables que componen el problema.

6.1.5. Correlación con desplazamiento horario

Observando la baja correlación obtenida entre el tráfico rodado y los contaminantes, se llegó a la conclusión de que una de las posibles causas fuera que el impacto de las emisiones producidas por el tráfico rodado no tuviera un efecto inmediato en los datos recogidos por la estación de calidad del aire, o que en invierno las turbulencias se dan un poco de pues de los picos de tráfico. Por lo que se tomó la decisión de correlacionar los contaminantes con los valores anteriores del tráfico. En la práctica esto supuso hacer un desplazamiento de las columnas de tráfico hacia arriba, haciendo coincidir el tráfico de una hora con el contaminante de t horas después, para t=1, 2, 3.

Como se puede observar se alcanza el máximo de correlación una hora más tarde para el caso de contaminante NO2.

- t (0) ES10/ES53/NO2: 0.325573 / 0.325315
- t (1) ES10/ES53/NO2: 0.363645 / 0.354687
- t (2) ES10/ES53/NO2: 0.350988 / 0.340106
- t (3) ES10/ES53/NO2: 0.306725 / 0.297716

No obstante, no sucede lo mismo con los contaminantes PM2.5 y PM10 que tienen una repercusión dos horas más tarde respecto al registro tomado en el sensor.

- t (0) ES10/ES53/PM2.5: 0.151163 / 0.191389
- t (1) ES10/ES53/PM2.5: 0.184814 / 0.219141
- t (2) ES10/ES53/PM2.5: 0.194180 / 0.225952
- t (3) ES10/ES53/PM2.5: 0.182534 / 0.211592

- t (0) ES10/ES53/PM10: 0.201407 / 0.240458
- t (1) ES10/ES53/PM10: 0.239198 / 0.271417
- t (2) ES10/ES53/PM10: 0.251706 / 0.278297
- t (3) ES10/ES53/PM10: 0.239569 / 0.260299

Estos resultados, nos dan una idea de que la influencia del tráfico en la contaminación es acumulativa y que debemos tener en cuenta no solo el valor del tráfico en el momento, s ino en las horas anteriores.

6.2. Modelos

En este proyecto se han utilizado técnicas de *Machine Learning* para la creación de un modelo predictivo para la calidad del aire con el objetivo de maximizar la precisión de las predicciones. Se han realizado pruebas con distintos algoritmos, pero el que mejor resultado ha dado ha sido el modelo de *Random Forest Regressor*.

Random Forest Regressor consiste en un conjunto de árboles de decisión combinados con *bagging*. Esto hace que cada árbol se entrene con distintas muestras del conjunto de datos para un mismo problema por lo que al combinar los resultados los errores cometidos por algunos árboles se compensan con otros, como resultado obtenemos predicciones que generalizan mejor. Además, es idóneo para el objetivo del proyecto.

El valor utilizado como referencia para medir la precisión de los diferentes modelos que utilizaremos será el de la fórmula *mean absolute error* «ecuación 1», MAE, esta mide la magnitud promedio de los errores en un conjunto de predicciones.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Ecuación 1. Fórmula correspondiente al mean absolute error

6.2.1. Modelo Naïve

El modelo *Naïve*, también conocido como modelo ingenuo, asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica. Por tanto, el modelo Naïve predice valores futuros como la repetición del último valor conocido. Esta predicción a priori tan simple, resulta no ser tan mala en caso de series temporales con una marcada estacionalidad, donde los valores se repiten periódicamente, por esta razón, se ha considerado como una referencia base para comprobar si nuestras predicciones superan o no este método y en qué medida.

Teniendo en cuenta este concepto, se ha calculado cual es el *mean absolute error* para las predicciones de la hora siguiente, con los datos de la hora anterior, obteniendo los siguientes resultados:

- El MAE con el modelo naïve para el CO: 0.0576
- El MAE con el modelo naïve para el NO2: 8.552
- El MAE con el modelo naïve para el PM2.5: 3.054
- El MAE con el modelo naïve para el PM10: 3.783
- El MAE con el modelo naïve para el O3: 6.858

El objetivo a partir de la realización de este modelo es intentar generar otro modelo de *Machine Learning*, con el cual mejorar la precisión en las predicciones obtenidas con el modelo *Naïve*.

Se ha probado a realizar este proceso con distintos algoritmos, pero se ha llegado a la conclusión que el algoritmo que realizaba mejores predicciones y tenía mejor comportamiento es el *Random Forest Regressor*, el cual utilizaremos a partir de ahora.

También se han intentado realizar algunas técnicas como el *Clustering* para ver si se conseguían mejorar las predicciones. Como veremos más adelante no se ha conseguido una mejora significativa por lo que se ha terminado descartando.

6.2.2. Modelo Random Forest Regressor

Para este modelo se ha tenido en cuenta una característica que en el modelo anterior se había pasado por alto, esta es la utilización del resto de las variables que pueden afectar en la predicción de la contaminación, véase las variables referentes a la meteorología como velocidad viento, dirección del viento, temperatura, humedad relativa, presión barométrica, precipitaciones, la información sobre el tráfico en las estaciones seleccionadas ES10, ES53, además de la hora y el tipo de día: laborable, sábado, domingo, festivo. Además, se añadió la predicción de la hora anterior.

Para la realización del modelo se ha utilizado una herramienta perteneciente a la librería de *Scikit-Learn*, la función *train_test_split*. Esta nos permite dividir el conjunto de datos en por lo menos dos bloques, uno destinado al entrenamiento y otro destinado a la validación del modelo. Esta división puede ser realizada de forma lineal, útil para conjuntos de datos con características de series temporales, como de forma aleatoria, configurable con el parámetro *shuffle*, el cual viene establecido de forma predeterminada a true. Además, también permite seleccionar el porcentaje de los datos que se destinarán al conjunto de entrenamiento y al conjunto de validación configurable con el parámetro test_size, el cual típicamente se establece a 0.30 aproximadamente para establecer 30% del conjunto de datos a de validación y el 70% a entrenamiento.

En el caso de las series temporales es importante entrenar con datos hasta cierta hora y realizar las predicciones con las horas siguientes, por lo que no se debería de elegir los conjuntos de validación y entrenamiento de forma aleatoria, como se hace habitualmente, porque esto haría que nuestro modelo usase datos del futuro para la predicción de valores del pasado.

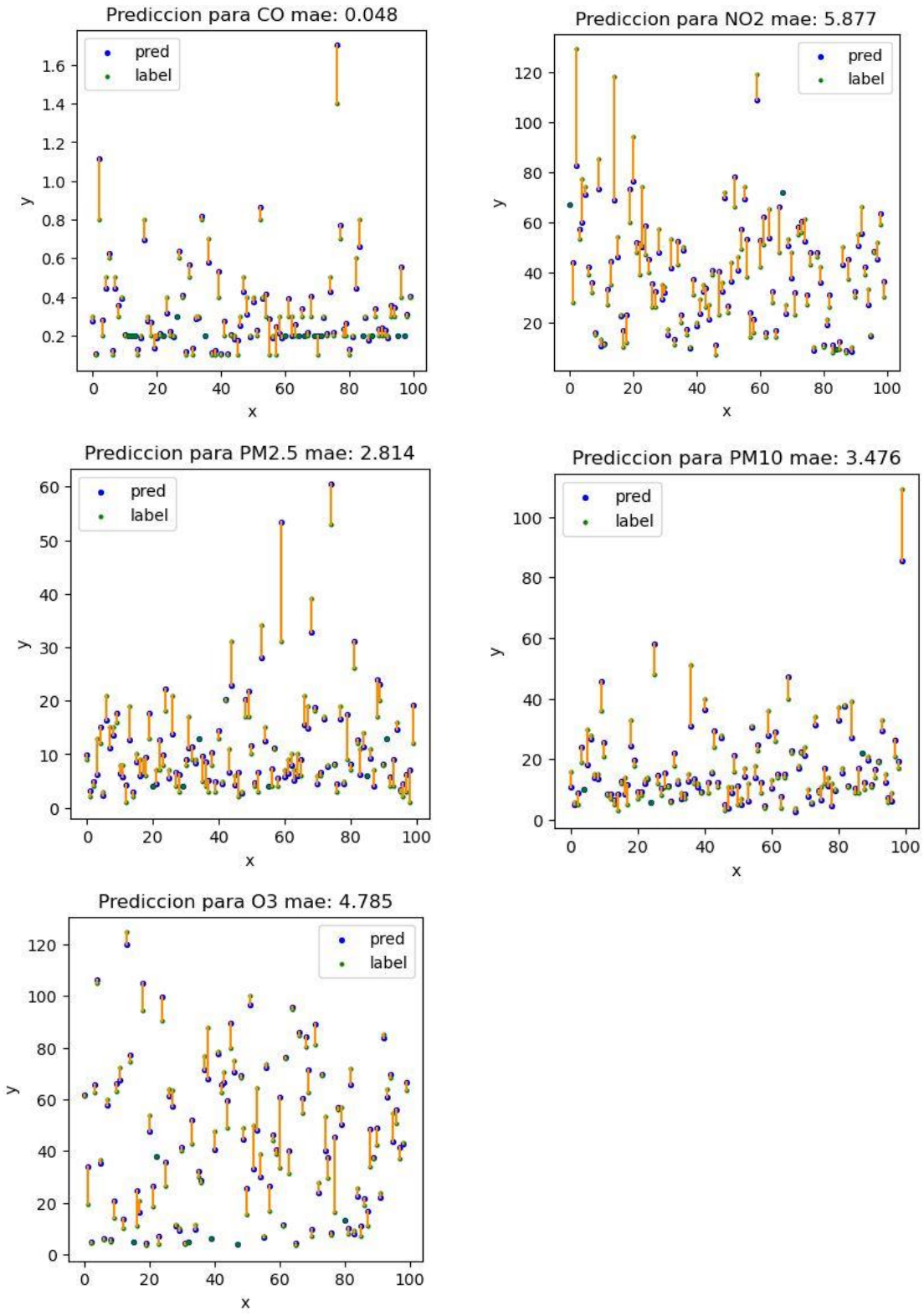


Figura 22. Predicciones para el modelo Random Forest Regressor de todos los contaminantes.

Fuente: elaboración propia mediante librería Matplotlib de Python.

Como podemos observar en la figura de arriba «figura 22.», las predicciones realizadas basándonos en la combinación de todas las variables que interfieren en la contaminación, conlleva una mejora significativa para todos los elementos correspondientes a las predicciones de calidad del aire. Aunque en general en los valores altos de contaminantes, se observa que el modelo tiende a subestimar los valores, pero debe apreciarse que en este caso se consideran errores absolutos, por lo que la diferencia no sería tan elevada. No obstante, teniendo en cuenta los resultados obtenidos en el apartado anterior «sección 6.2.1.», para el contaminante CO podemos encontrar una mejora del 15,7% respecto al modelo *Naïve*, de forma más notoria lo podemos apreciar para el NO_2 y para el O_3 que conlleva una mejora del 31,27% para el primero y de 30,21% para el segundo. De igual forma ocurre para las partículas en suspensión tanto las PM2.5 como las PM10 que su mejora respecto al modelo *Naïve* es de 7,85% y de 8,11% respectivamente.

6.3. Clustering

El *clustering*, es una técnica de *Machine Learning* de clasificación no supervisada que consiste en agrupar los datos en distintos conjuntos homogéneos basándose en la similitud de sus características o propiedades.

Para intentar llevar a cabo estas agrupaciones, se han utilizado dos algoritmos, el método de *k-Means* y el método de *Silhouette*.

6.3.1. K-Means.

Para conseguir los grupos, este algoritmo utiliza en proceso iterativo en el que se van ajustando para obtener el número de agrupaciones idóneas. Para poder llevar a cabo la implementación, se ha utilizado el método *kMeans* de la librería de Scikit-Learn, para ello se debe especificar el número de grupos y el conjunto de datos sobre el que queremos realizar *clustering*.

Posteriormente, el algoritmo inicializa los centroides de forma aleatoria y se generan los grupos calculándose la distancia cuadrada Euclidiana más próxima al centroide. Después los centroides de cada grupo son recalculados. Este algoritmo realiza estas operaciones

de forma iterativa hasta que: no hay cambios en los puntos asignados a los grupos, si la suma de las distancias se minimiza o si se alcanza un número máximo de iteraciones.

La ventaja de este algoritmo es que es rápido y sencillo, pero sensible a *outliers* y al inicializar los centroides de forma aleatoria puede llegar a reproducir diferentes valores en diferentes ejecuciones. (Bagnato, 2018)

Para llevarlo a cabo en el proyecto se han seleccionado las variables que estamos utilizando hasta ahora, estas son las referentes a las variables de calidad del aire, meteorología y tráfico que seleccionamos anteriormente «ver sección 6.1.». Se ha procedido a realizar el desplazamiento horario para las variables de tráfico que como pudimos se obtenía una mejora «ver sección 6.1.5.». Después se ha procedido a realizar una normalización de los datos debido a la gran diferencia de valor entre algunas de las variables y el tipo de valor recogido, si no realizáramos esta normalización los atributos de mayor valor dominarán las distancias, por lo que los atributos con valores más bajos perderían importancia.

Tras esta preparación procedemos a definir nuestro valor «k», el número de *clusters*, en nuestro caso vamos a intentar realizar de dos a quince *clusters*, obteniendo el siguiente resultado «figura 23.».

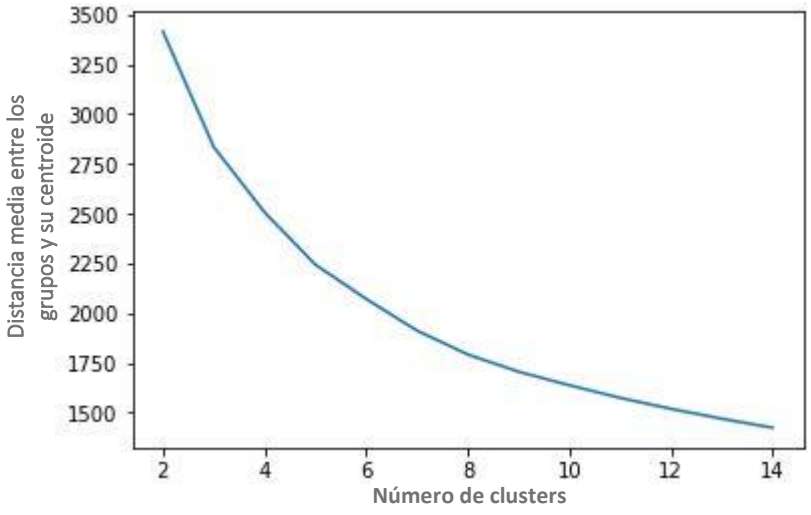


Figura 23. Representación gráfica del algoritmo k-means.
Fuente: elaboración propia mediante librería Matplotlib de Python.

Como se puede apreciar en la figura anterior, no se puede definir un claro número de grupos, por lo que se ha decidido implementar análisis de *Silhouette* o análisis de la silueta.

6.3.2. Silhouette

Este mide la calidad del agrupamiento, midiendo la separación entre los distintos *clusters* cuya medida se encuentra entre [-1,1]. La interpretación del valor arrojado por el criterio de la silueta es que cuanto más cercano al uno sea dicho valor, mejor será el agrupamiento indicando que la observación se encuentra lejos de los *clusters* vecinos. Por el contrario, un calor cercano a cero, indica que la observación está cerca o en la frontera entre dos *cluster* y por último valores negativos indican que esas muestras quizás estén asignadas de forma errónea.

Este método calcula la media de los coeficientes de silueta de todas las observaciones para distintas agrupaciones, distintos valores de k. El número óptimo de *clusters* es el que maximiza la media de los coeficientes.

Para este caso, hemos utilizado la función *silhouett_score* perteneciente a la librería Scikit-Learn.

Se han utilizado los mismos valores de *k-Means* utilizando la distancia euclídea obteniendo los siguientes resultados.

- Para 2 *clusters* el promedio es: 0.28
- Para 3 *clusters* el promedio es: 0.25
- Para 4 *clusters* el promedio es: 0.23
- Para 5 *clusters* el promedio es: 0.23
- Para 6 *clusters* el promedio es: 0.22
- Para 7 *clusters* el promedio es: 0.21
- Para 8 *clusters* el promedio es: 0.21
- Para 9 *clusters* el promedio es: 0.21
- Para 10 *clusters* el promedio es: 0.20

Utilizando este método, podemos concluir que no hay un número claro de *clusters* óptimo, por lo que se ha procedido a entrenar el modelo *Random Forest Regressor* añadiendo la columna del *cluster* asignado, para cada conjunto, obteniendo unos resultados para el NO_2 que no mejoran e incluso en algunas ocasiones empeoran los resultados obtenidos por el modelo *Naïve*, por lo que se ha procedido a descartar este el método para intentar realizar mejores predicciones.

6.5. Degradación del error

Hasta el momento, se ha estado tratando el conjunto de datos como una serie de datos independientes, pero como se ha mencionado anteriormente «sección 5.3.», nuestro conjunto se comporta como una serie temporal estacionaria.

Durante el entrenamiento de nuestro modelo, se ha utilizado la función *train_test_split* «ver la sección 6.2.», esta herramienta la habíamos utilizado de forma que se realizará una partición del conjunto de datos de forma aleatoria, lo que conlleva la ruptura de la serie temporal y por tanto la pérdida de parte de la información de la que disponemos. Además, corremos el riesgo de entrenar con valores del futuro y validar con valores del pasado. Para solucionar este problema se ha utilizado el concepto que utiliza la función *timeSeriesSplit* perteneciente a la librería de *Scikit-Learn*. Su objetivo es proporcionar índices de entrenamiento y prueba para dividir conjuntos de datos de series temporales. Este método, crea divisiones incrementales donde el conjunto de entrenamiento aumenta en cada iteración y donde los valores de prueba son los siguientes. «figura 24.»

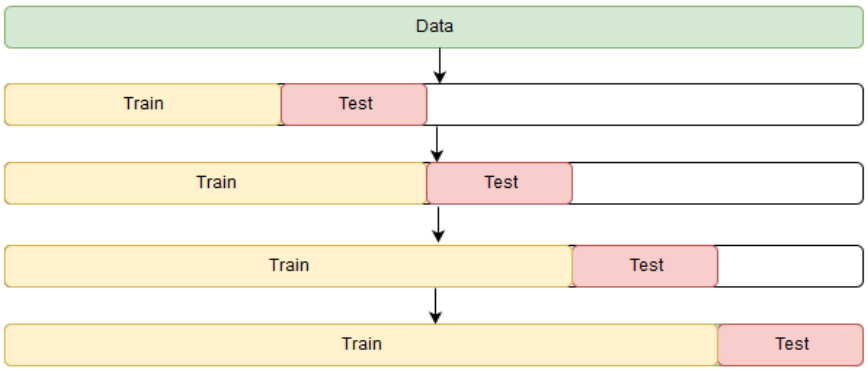


Figura 24. Representación gráfica de cómo trabaja TimeSeriesSplit.

Fuente: Parra-Royón, M. (2017). Series Temporales. https://github.com/manuparra/seriestemporales/blob/master/README.md

Además, se ha generado una función que incorpora al conjunto de datos tantas columnas como datos del pasado se especifique, para todas las variables de entrada del modelo, forzando una entrada con un mayor número de variables, para intentar compensar las pérdidas de no utilizar un algoritmo que trabaje con series temporales, estas nuevas columnas generadas serán las variables desplazadas (t + p) veces, siendo la variable p el tiempo hacia atrás especificado.

Durante este apartado, veremos cómo se degrada el error a lo largo del tiempo para los modelos que disponemos. Además, se estudiará de igual forma la degradación omitiendo subconjuntos de datos, entendiendo como subconjunto los tipos de datos que disponíamos al comienzo de este proyecto, contaminación o calidad del aire, meteorología, tráfico y festividades, representado en las siguientes gráficas como calendario. Con el objetivo de observar su comportamiento, el ruido y la dependencia de dichos subconjuntos.

6.5.1 Comparación modelos

A continuación, se muestra la degradación del error para una f=48 horas, comparando el modelo *Naïve*, el modelo *Random Forest Regressor* con datos de entrada con p= 0 horas anteriores y el mismo modelo para datos de entrada p=10 horas anteriores, este último se consigue utilizando la función nombrada durante el apartado anterior, la cual aplica una transformación que combinando p filas consecutivas, logra filas con más atributos que contienen los datos de las p horas anteriores, para todos los contaminantes. Esta última transformación, da lugar a un proceso mucho más lento, pero se observará más adelante, en general se obtendrán mejores resultados.

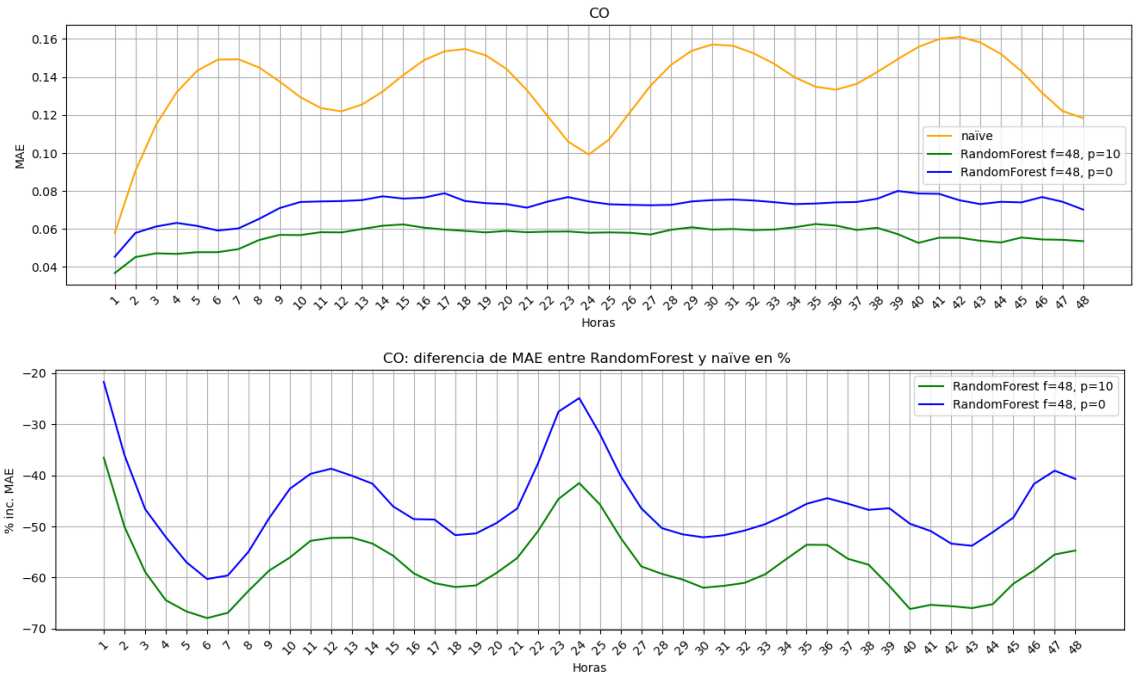


Figura 25. Gráfica con los valores absolutos del error para los métodos Naïve, para Randomforest sin pasados anteriores y para Randomforest con 10 pasados anteriores para el CO (parte superior) y esta misma información como % de error sobre el método Naïve (parte inferior).

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

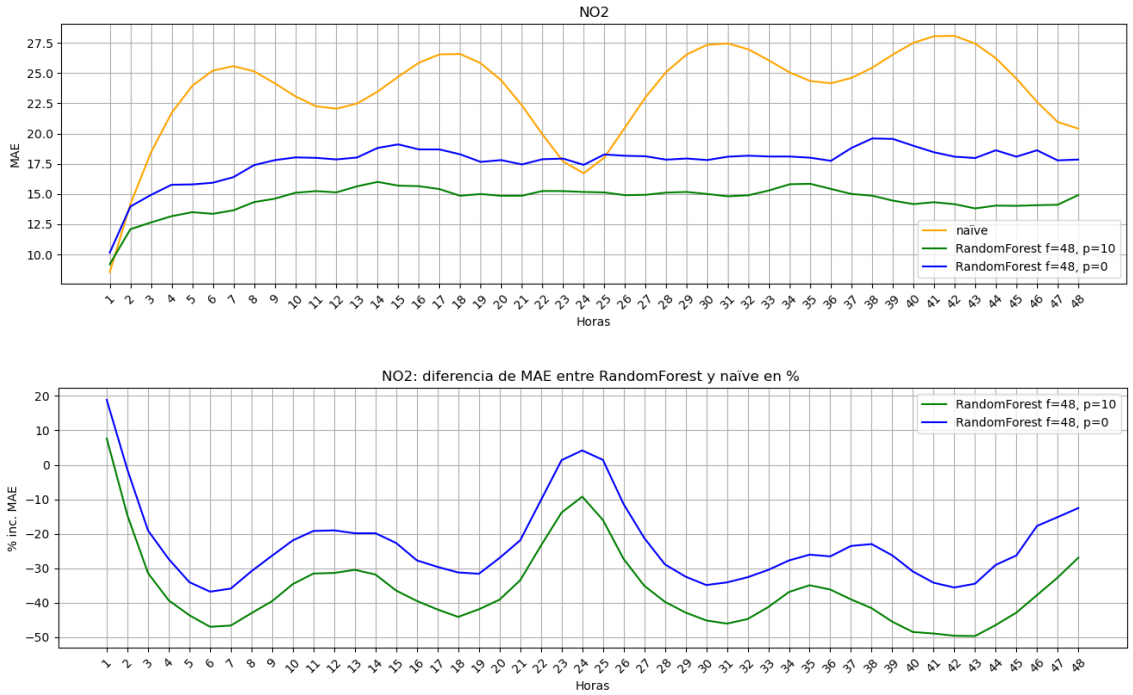


Figura 26. Gráfica con los valores absolutos del error para los métodos Naïve, para Randomforest sin pasados anteriores y para Randomforest con 10 pasados anteriores para el NO_2 (parte superior) y esta misma información como % de error sobre el método Naïve (parte inferior).

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

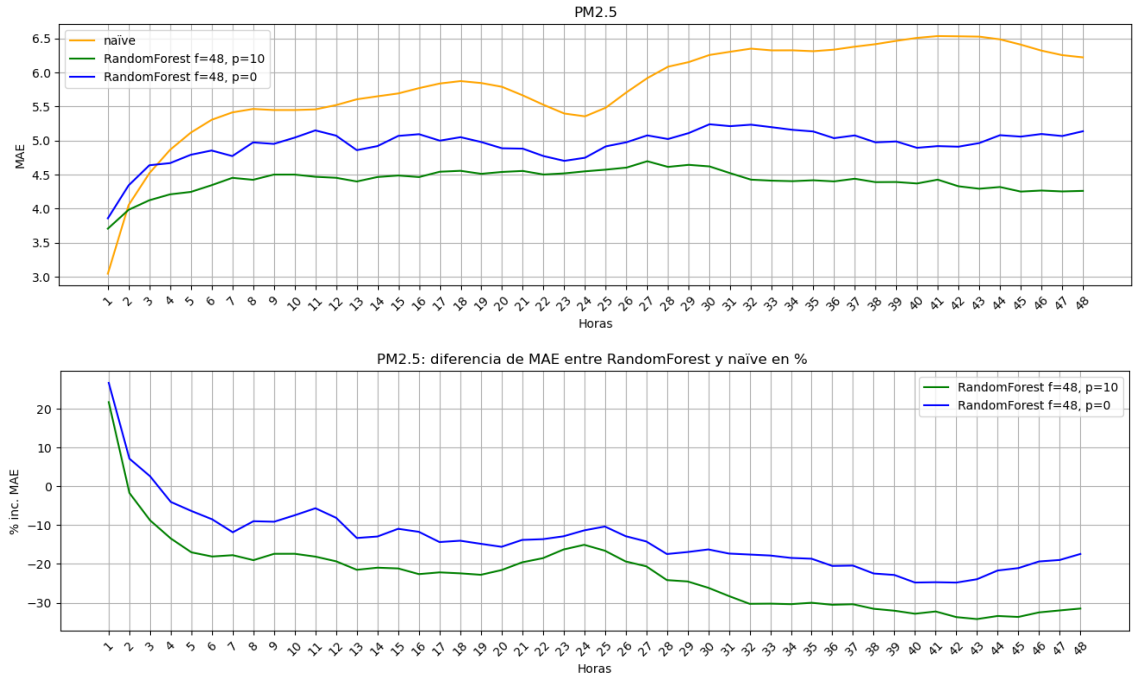


Figura 27. Gráfica con los valores absolutos del error para los métodos Naïve, para Randomforest sin pasados anteriores y para Randomforest con 10 pasados anteriores para el PM2.5 (parte superior) y esta misma información como % de error sobre el método Naïve (parte inferior).

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

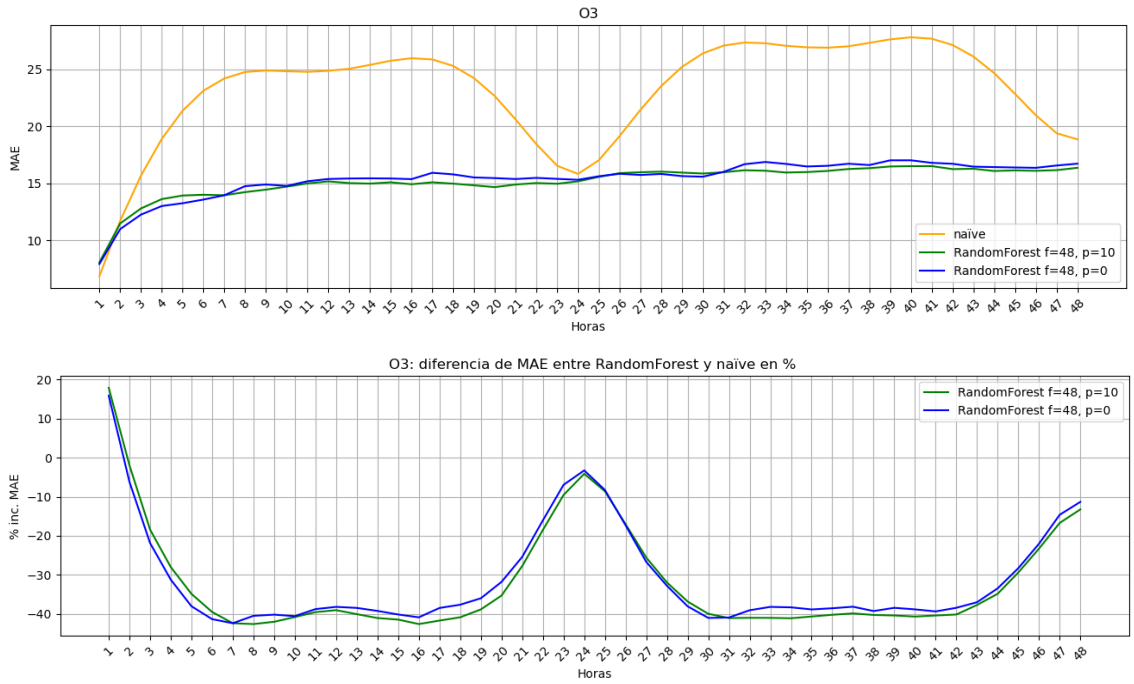


Figura 28. Gráfica con los valores absolutos del error para los métodos Naïve, para Randomforest sin pasados anteriores y para Randomforest con 10 pasados anteriores para el O_3 (parte superior) y esta misma información como % de error sobre el método Naïve (parte inferior).

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

Como se puede observar se ha obtenido dos gráficas por cada contaminante, la gráfica superior muestra el mae en términos absolutos, comparando los modelos mencionados al principio de este apartado y donde un valor más elevado, indica mayor error en las predicciones. En cambio, en la gráfica inferior se muestra el porcentaje de mejora con respecto al modelo Naïve, donde un valor positivo en el eje y, indica por tanto que ha obtenido peores resultados que el modelo Naïve y un valor negativo en el eje de ordenadas, indica unos mejores resultados.

Como se observa a lo largo de las cuatro figuras anteriores, el modelo *Naïve* tienen un componente cíclico cada 24 horas, donde alcanza un mínimo para posteriormente repetir el ciclo, pero como también se puede observar, tiene un incremento en la degradación de las predicciones a lo largo del tiempo. Para las «figuras 25.» y «figura 26.», a las 7 horas y a las 18 horas obtenemos unos máximos cuyo fenómeno está relacionado con los horarios de desplazamiento de la población, correspondientes a máximos y horas punta.

Para el caso del NO_2 «figura 26.», podemos observar que a las 24 horas el modelo *Naïve* mejora al modelo *Random Forest Regressor* que no tiene en cuenta las horas anteriores, esto es debido a que el ciclo diurno este contaminante suele repetirse, especialmente en

situaciones meteorológicas invariantes, lo que nos haría plantearnos si merece la pena utilizar dicho modelo para intentar predecir la contaminación a 24 horas futuras, confirmando el dicho «un reloj estropeado da dos veces bien la hora al día.»

No obstante, para el contaminante CO «figura 25.», al igual que el NO_2, a las 24 horas alcanzar el mínimo, aunque no llega a mejorar las predicciones obtenidas por ambos modelos de *Random Forest Regressor*.

Sin embargo, para las partículas menores de 2.5μm «figura 27.» podemos observar que no hay una relación tan evidente con los desplazamientos poblacionales, si no que posee una dependencia con la velocidad del viento, favoreciendo la concentración o la dispersión de este, haciendo más evidente la degradación en la predicción respecto a otros contaminantes. Al igual ocurre con el Ozono el cual está relacionado con la radiación ultravioleta cuando el mismo en las predicciones.

Por último, hay que destacar que tanto para el Ozono como para las partículas menores de 2.5μm, no merece la pena invertir tiempo en el entrenamiento del modelo para obtener la degradación con las p=10 horas anteriores, porque el beneficio obtenido es insignificante en relación con el tiempo invertido.

En la siguiente página, se muestra una gráfica con f=100 horas «figura 29.», en la cual se puede ver que para predicciones que se extienden en el tiempo, en este caso cien horas, el modelo *Random Forest Regressor* con las p=10 horas anteriores, disminuye considerablemente su porcentaje de acierto, siendo más efectivo el modelo de *Random Forest Regressor* sin tener en cuenta horas anteriores p=0.

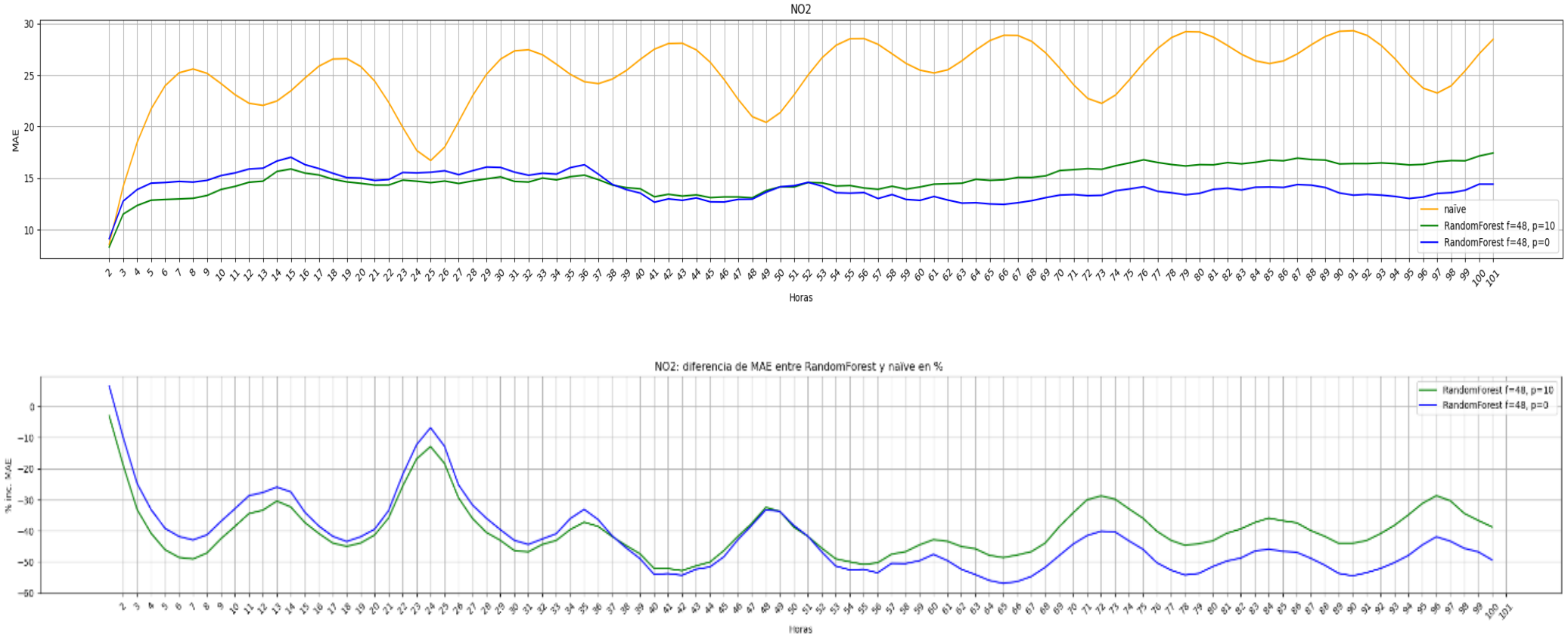


Figura 29. Degradación del Error para NO2 para 100 horas.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

6.6. Influencia de los grupos de datos en la predicción

Durante este apartado, al igual que en la sección anterior, ese pretende estudiar la degradación del error para los tres tipos de modelos explicados anteriormente «sección 6.5.1.», pero con la diferencia de omitir del conjunto de datos cada uno de los subconjuntos de forma independiente, con el objetivo de poder determinar la importancia de cada uno de ellos a la hora de realizar predicciones. En este caso se ha realizado para el contaminante CO.

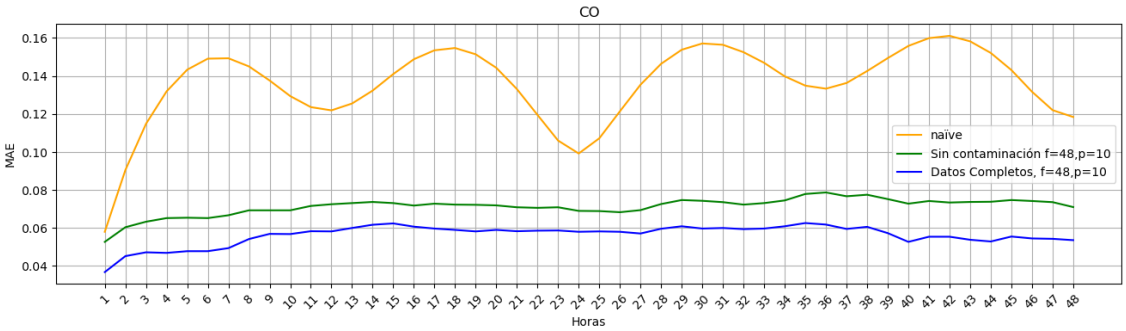


Figura 30. Degradación del CO sin el subconjunto de datos de calidad del aire.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

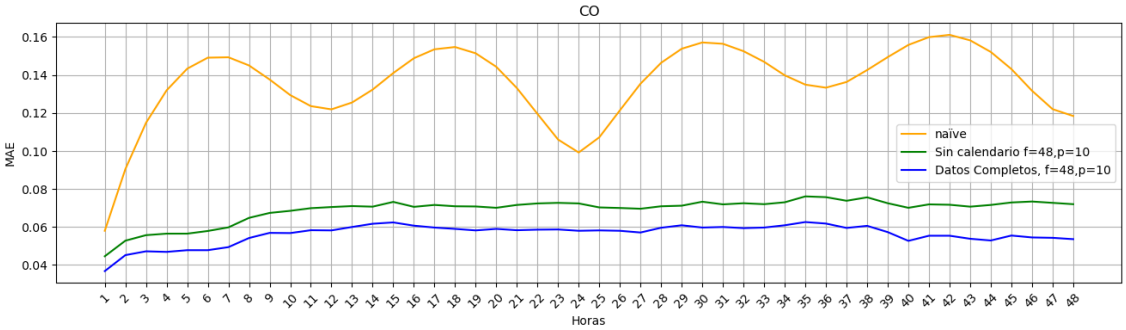


Figura 31. Degradación del CO sin el subconjunto de datos correspondientes a las festividades.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

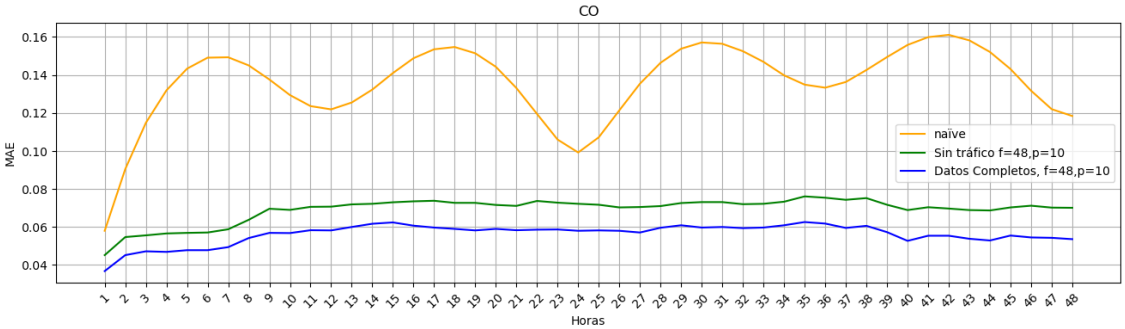


Figura 32. Degradación del CO sin el subconjunto de tráfico.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

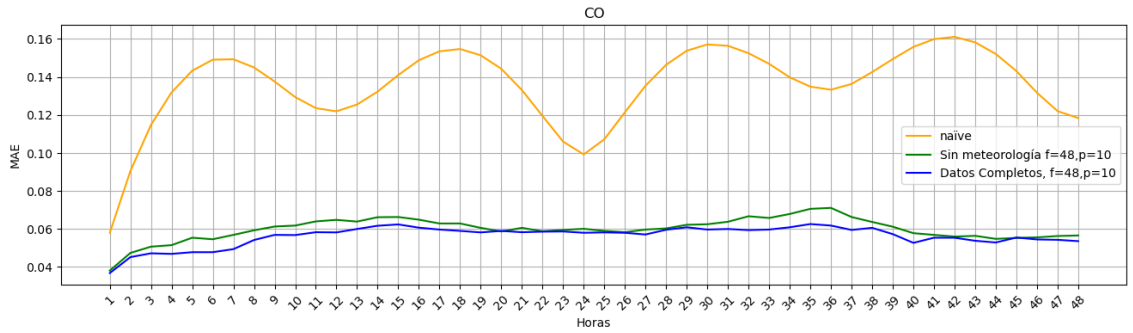


Figura 33. Degradación del CO sin el subconjunto de datos de meteorología.

Fuente: elaboración propia mediante librería Matplotlib de Python y potencia de cálculo del despacho del director de proyecto.

El conjunto de datos de calidad del aire está conformado por las variables seleccionadas durante el apartado que trata las correlaciones entre los distintos contaminantes «sección 6.1.1.». Como se puede observar en la «figura 30.» si omitimos los datos pertenecientes a la calidad del aire, conseguimos un empeoramiento de las predicciones.

De igual manera ocurre con la omisión del subconjunto de tráfico «sección 6.1.3.» y con la omisión de festividades, cuyos valores tienen el objetivo de identificar las filas como laborables, sábados, domingos y festivos en nuestros datos. Como era de esperar, la ausencia de estas causa una disminución en la precisión de las predicciones. «figura 31.» «figura 32.»

No obstante, como se puede ver en la «figura 33.» la omisión del subconjunto de datos de meteorología, cuyos elementos que lo conforman son los datos descritos durante la sección «sección 6.1.2.», no tienen ningún tipo de implicación considerable, por lo menos para los modelos que tienen en cuenta valores de 10 horas pasadas. Esto significa que tanto los datos de calidad del aire anterior, como los de tráfico como los de festividades son importantes para una mejor predicción, pero no así los datos de meteorología que no parecen mejorar los resultados cuya consecuencia puede residir en el propio ciclo diurno de las variables meteorológicas.

7. CONCLUSIONES Y TRABAJO FUTURO

En vista de los resultados obtenidos durante la realización de este trabajo y partiendo de los objetivos definidos al comienzo, se han obtenido las siguientes conclusiones:

O1. Tras el desarrollo de este trabajo, se puede corroborar que los datos de calidad del aire, meteorológicos y de tráfico necesarios, están disponibles para cualquier persona, ya que no se necesitan requisitos previos para poder acceder a ellos, estando disponibles y abiertos al público y pudiendo ser descargados de forma sencilla accediendo directamente a la web del Ayuntamiento de Madrid o mediante automatización de esta tarea mediante *Python*.

O2. También se han tenido que realizar las transformaciones necesarias para cada conjunto de datos y así poder unificarlos todos en un único fichero. No obstante, a diferencia del resto, los datos sobre tráfico suponen un incremento de dificultad, debido a la cantidad de transformaciones previas que se debían realizar, por presentar una distribución más compleja y con mayores diferencias respecto al resto de los conjuntos de datos. A pesar de estas dificultades, se ha conseguido unificarlos en un único fichero gracias a la herramienta y librerías de *Python,* todas las transformaciones se han realizado mediane la librería *Pandas,* la cual facilita la realización de estas tareas.

O3. Se ha realizado un análisis y visualización de cada conjunto de datos, con el objetivo de quitar variables que estuvieran altamente relacionadas y así poder tener un único fichero con el menor número de datos redundantes. Asimismo, se ha intentado omitir la mayor cantidad *outliers* posible, para así reducir el ruido que estos infieren en nuestros modelos.

Se ha podido corroborar durante el desarrollo del trabajo, que nuestros datos al ser recopilados secuencialmente a lo largo del tiempo presentan un formato de serie temporal, los cuales, mediante varias pruebas, se corrobora que estos poseen un comportamiento estacionario, por lo que no se ha tenido que realizar la transformación de la tendencia de la serie.

O4. Se ha conseguido predecir el comportamiento de los contaminantes mediante el uso del algoritmo *Random Forest Regressor*, comparándolo con el modelo Naïve, obteniendo una mejora considerable respecto al modelo de partida.

Asimismo, se han utilizado diferentes métodos para mejorar dichos resultados. Los cuales, algunos, como el *clustering*, no han logrado mejorar las predicciones.

O5. Se ha comprobado que en general, para los modelos utilizados durante este trabajo, todos los contaminantes se pueden predecir mejor que con el modelo Naïve.

Además, se ha podido determinar que tener en cuenta horas pasadas ayuda a realizar mejores predicciones, teniendo en cuenta una serie de factores. Por ejemplo, deberíamos tener en cuenta cual sería el enfoque para nuestro problema, puesto que, con los resultados obtenidos, no es lo mismo predecir para veinticuatro horas, doce horas, treinta y ocho horas, o hasta cien horas futuras. Como se ha visto en algunos casos, el tiempo necesario para realizar la predicción no se compensa con una mejora significativa en las predicciones para determinadas horas. Por lo que, para cada tipo de predicción, en función de las necesidades del problema, convendría usar un algoritmo distinto.

Se puede determinar que la creación de un modelo de *Machine Learning*, que no tenga en cuenta series temporales pero que sí tengan en cuenta valores de un determinado número de horas anteriores. obtiene mejoras significativas en el rango de las 5-22h y 29-45h, para la mayoría de los contaminantes. No obstante, para los contaminantes O_3 y PM2.5 no hay una diferencia considerable que compense el tiempo invertido para la obtención de esos resultados.

Como trabajos futuros, se podría estudiar el comportamiento del modelo si además de utilizar un número de horas del pasado también se le añaden predicciones para un determinado número de horas a futuro.

BIBLIOGRAFÍA

B., A. (2014). *MAD formula for outlier detection*. Obtenido de stats.stackexchange: https://stats.stackexchange.com/questions/123895/mad-formula-for-outlier-detection

Bagnato, J. I. (12 de 3 de 2018). *K-Means en Python paso a paso*. Obtenido de aprendemachinelearning.com: https://www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/

Brownlee, J. (30 de 12 de 2016). *How to Check if Time Series Data is Stationary with Python*. Obtenido de machinelearningmastery.com : https://machinelearningmastery.com/time-series-data-stationary-python/

C.Román-Cascón, C. J.-R. (s.f.). *Comparing mountain breezes and their impacts on CO2 mixing ratios at three contrasting areas*. Obtenido de sciencedirect.com: https://www.sciencedirect.com/science/article/pii/S0169809519300596?via%3Dihub

ecologistasenaccion. (s.f.). *¿Qué son las PM2,5 y cómo afectan a nuestra salud?* Obtenido de ecologistasenaccion.org : https://www.ecologistasenaccion.org/17842/que-son-las-pm25-y-como-afectan-a-nuestra-salud/

Enviraiot. (10 de 7 de 2020). *¿Cuáles son los gases más contaminantes que hay en la atmósfera?* Obtenido de enviraiot.es : https://enviraiot.es/cuales-son-gases-contaminantes-de-la-atmosfera/

Fernandez, L. G. (2018-2019). *UNIVERSIDAD NACIONAL DE EDUCACION A DISTANCIA*. Obtenido de e-spacio.uned.es: http://e-spacio.uned.es/fez/eserv/bibliuned:master-ETSInformatica-IAA-Lgarcia/Garcia_Fernandez_Lorena_TFM.pdf

Madrid, A. d. (s.f.). *Aforos de tráfico en la ciudad de Madrid permanentes*. Obtenido de datos.madrid.es: https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=fabbf3e1de124610VgnVCM2000001f4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default

Madrid, A. d. (s.f.). *Calidad del aire. Datos horarios desde 2001*. Obtenido de datos.madrid.es : https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=f3c0f7d512273410VgnVCM2000000c205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default

Madrid, A. d. (s.f.). *Calidad del aire. Estaciones de control*. Obtenido de datos.madrid.es: https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=9e42c176313eb410VgnVCM1000000b205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default

Madrid, A. d. (s.f.). *Datos meteorológicos. Datos horarios desde 2019*. Obtenido de datos.madrid.es : https://datos.madrid.es/sites/v/index.jsp?vgnextoid=fa8357cec5efa610VgnVCM1000001d4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD

Madrid, A. d. (s.f.). *Datos meteorológicos. Estaciones de control*. Obtenido de datos.madrid.es: https://datos.madrid.es/sites/v/index.jsp?vgnextoid=2ac5be53b4d2b610VgnVCM2000001f4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD

Navas, J. L. (30 de 4 de 2020). *academica-e.unavarra*. Obtenido de academica-e.unavarra.es: https://academica-e.unavarra.es/xmlui/bitstream/handle/2454/37601/Detecci%C3%B3n%20de%20outliers%20en%20series%20temporales%20de%20contaminantes.pdf?sequence=1&isAllowed=y

Nishtha. (26 de 4 de 2021). *How to check Stationarity of Data in Python.* Obtenido de analyticsvidhya.com : https://www.analyticsvidhya.com/blog/2021/04/how-to-check-stationarity-of-data-in-python/

Pintado, G. V. (9 de 6 de 2019). *Universidad Oberta de Catalunya*. Obtenido de uoc.edu: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/99446/7/gabvilpiTFM0619memoria.pdf

Querol Carceller, X. (2018). *La calidad del aire. Un reto mundial.* Madrid: Fundación Naturgy.

Ripley, B. D. (2004). *Robust Statistics*. Obtenido de web.archive.org: http://web.archive.org/web/20120410072907/http://www.stats.ox.ac.uk/pub/StatMeth/Robust.pdf

Sousa, D. G. (2019). *uc3m.* Obtenido de e-archivo.uc3m: https://e-archivo.uc3m.es/bitstream/handle/10016/32687/TFG_Daniel_Garcia_Sousa.pdf?sequence=2&isAllowed=y

WHO. (s.f.). WHO global. Obtenido de apps.who.int: https://apps.who.int/iris/bitstream/handle/10665/345329/9789240034228-eng.pdf?sequence=1&isAllowed=y

INTRODUCTION

1.1. Problem statement

The following section explains how the problem has been approached, from the starting point to the different techniques used.

Due to the problems caused to people's health by living with high levels of atmospheric pollution, it is important to know in advance when these levels are going to be reached and to take appropriate measures. In this case, Data Science techniques will be used to find out the relationship between the different pollutants together with meteorology and road traffic, in order to predict the behaviour of the most harmful elements and act accordingly. In this way, we avoid exceeding the limits established by the WHO, which suppose a high risk to people's health.

For this work we have obtained real data provided by the Madrid City Council, categorised into three types: air quality data, meteorological data, traffic data and festivities. The recording of these data is carried out in hourly intervals. These have had to be unified in a single file in order to be able to analyse them correctly. From this point, the study and application of different Data Science techniques is carried out to extract relevant information, based on the use of Machine learning algorithms presenting a transformation in the data input, in such way that it takes better advantage of the time series nature of these data.

We will be able to see the behaviour of different pollutants for the same model and determine if it is possible to make satisfactory predictions with this method, as well as the degradation of the predictions when they are made for far-off futures.

The interpretation of the results has been supported by the micrometeorology and climate variability group of the Complutense University of Madrid, in particular with the advice of its director, Carlos Yagüe, and the co-direction of Carlos Román Cascón.

1.2. Objectives

The aim of this project is to illustrate how technology and data analysis can be used for the common good, such as predicting and reducing pollution levels.

The summary of the objectives would be:

O1. Display that air quality, meteorological and traffic data are available for public use and show where they are available, how to download and process them. In this way they can be made available to anyone who needs them for other studies.

O2. Use the most popular and current methods found in the world of data analysis in order to understand and pre-process the data, generating a single file with all the information we will need for the prediction objective.

O3. Illustrate through visualisation tools the relationship between different datasets.

O4. Predict the behaviour of pollutants by using machine learning methods.

O5. Draw conclusions about if such predictions are possible and to observe which pollutants are most affected by specific factors.

Furthermore, as a personal objective, knowledge of Python programming and the use of the most relevant libraries to process and analyse data has been obtained.

These objectives correspond to the classic phases of Data Science: downloading, pre-processing and visualisation.

1.3. Memory content

This report consists of the following sections:

1. Introduction: this chapter sets out the problems, objectives and mechanisms for achieving them.

2. State of the art: during this chapter, the study of the main gases that are most harmful to health will be exposed. Also, what are the maximum pollution levels recommended by the World Health Organisation. It will also present the main differences between this project and other related projects.

3. Origin and collection of the data: in this section, the source from which the data have been extracted and the methodology used to obtain them will be named. In addition, it also explains the description of the data in its original form, the space it occupies on disk, number of files, etc. This chapter corresponds to objective O1, which will be covered after its end.

4. Data pre-processing and cleaning: during this section, the methods used for data treatment and cleaning will be explained, in order to achieve a suitable format for its study, and thus facilitate the extraction of information. The result obtained after the transformation is also shown.

5. Null values and outliers: the different techniques and methods for the treatment of null values and unexpected values or outliers are shown, with the aim of not losing too much relevant information and eliminating information that may introduce noise in the results, respectively. During the development of this chapter, objectives O2 and O3 will be achieved, which will be covered after the conclusion of this chapter.

6. Data analysis and information extraction: the aim of this section is to present the algorithms used for the creation and selection of models, as well as the techniques that can be used to improve their predictions.

This section also wants to visualise the comparison between the different models. In addition, the objectives O4 and O5 will be achieved in the course of this chapter.

7. Conclusions and future work: this section lists the conclusions reached following the results obtained during the previous section. It also describes possible future work that could be carried out as a result of this project.

Attached below is a link to the GitHub repository containing the code created during this work:

- https://github.com/albercol/TFG

1.4. Tools used

This section details the most important tools and libraries that have been used for the realisation of the project.

I. Python

High-level programming language, which aim is to focus on code readability. It is a multi-paradigm programming language, because it supports object-oriented programming, imperative programming and functional programming.

In turn, it is the most widely used programming language in analysis and Data Science because of its large community, the large number of statistical and mathematical tools, and its ease of learning.

II. Jupyter Notebook

Jupyter Notebook is an open source web application that allows data scientists to create and share documents that integrate live code, equations, computational results, graphical

visualisations and other resources, along with explanatory text. Being able to re-execute code snippets without the need to run the entire script.

The main attraction for analytics and Data Science is the possibility to generate reports, graphs and results in one interface.

It was launched in 2015 by the non-profit organisation Project Jupyter:

III. Libraries

- *os* library: is a module that provides an easy and versatile way to use operating system-dependent functionalities.

  This module has been used to access the folder where the data previously obtained on atmospheric pollution is stored.

- *Glob* library: is a module that finds all path names that resemble a specified pattern, according to the rules used on a Unix terminal.

  With this module, the name of the files that have been downloaded has been identified and an identifying name has been assigned to them in accordance with their content.

- *Pandas* Library: is a library specialised in the handling and analysis of data structures.

  With this library it has been possible to open and read the files, as well as save them in a dataframe variable, and store them in a list in order to be able to concatenate all the contents of this list in a single file that stores all the information.

- *Beautifulsoup* library: is a Python library that allows extracting information from HTML or XML content. To use it, it is necessary to specify a parser, responsible for transforming an HTML or XML document into a complex tree of *Python* objects. This allows us to interact

with the elements of a web page as if we were using the developer tools of a browser.

This tool has been used to examine the elements of the City Council's website and to automate the process of downloading all the files necessary to carry out this work.

- *Request* library: This library allows HTTP and FTP requests to be made and has been used to download part of the files necessary for this project.

- *Zipfile* library: this library provides tools to create, read, write, add and display a ZIP file.

  This library has been used because one of the downloaded files that we will use during this work was in this format. This tool will allow us to extract the individual components.

- *Scikit-Learn* library: this library has classification, regression, clustering and dimensionality reduction algorithms. It is also compatible with other libraries such as *NumPy*, *SciPy* and *Matplotlib*, which will be used during the prediction stage.

- *NumPy* library: a library specialised in numerical computation and data analysis, especially for large volume data, this tool supports Pandas to represent data vectors.

  It incorporates objects called arrays, which allow the representation of collections of data of the same type in several dimensions and efficient functions for their manipulation.

- *Matplotlib* library: it is a library specialised in the creation of two-dimensional graphs. This tool will be used during this work to represent the graphs and to visualise the results obtained.

- *Seaborn* library: like Matplotlib, it is a tool that provides a high-level interface for the representation of statistical graphs. This library will be used for the visual representation of the values obtained.

CONCLUSIONS

In view of the results obtained during the course of this work and based on the objectives defined at the beginning, the following conclusions have been drawn:

O1. After the development of this work, it can be corroborated that the necessary air quality, meteorological and traffic data are available to anyone, as no prerequisites are needed to access them, as they are available and open to the public and can be downloaded easily by directly accessing the City Council's website or by automating this task using *Python*.

O2. The necessary transformations also had to be carried out for each dataset in order to unify them all in a single file. However, unlike the rest, the data on traffic is more difficult due to the number of previous transformations that had to be carried out, as it has a more complex distribution and greater differences with respect to the rest of the datasets. Despite these difficulties, it has been possible to unify them in a single file thanks to the Python tool and libraries; all the transformations have been carried out using the Pandas library, which facilitates the performance of these tasks.

O3. An analysis and visualisation of each dataset has been carried out, with the aim of removing variables that are highly related in order to have a single file with as few redundant data as possible. We also tried to omit as many outliers as possible, in order to reduce the noise they infer in our models.

It has been possible to corroborate during the development of the work that our data, when collected sequentially over time, present a time series format, which, through several tests, it is corroborated that they have a stationary behaviour, so it has not been necessary to perform the transformation of the trend of the series.

O4. The behaviour of pollutants has been predicted using the Random Forest Regressor algorithm and compared with the Naïve model, obtaining a considerable improvement over the initial model.

Furthermore, different methods have been used to improve these results. Some of them, such as clustering, have failed to improve predictions.

O5. It has been found that in general, for the models used during this work, all pollutants can be predicted better than with the Naïve model.

In addition, it has been found that taking into account past hours helps to make better predictions, considering a number of factors. For example, we should bear in mind what would be the approach to our problem, since, with the results obtained, it is not the same to predict for twenty-four hours, twelve hours, thirty-eight hours, or even one hundred hours in the future. As has been seen in some cases, the time needed to make the prediction is not compensated by a significant improvement in the predictions for certain hours. Therefore, for each type of prediction, depending on the needs of the problem, a different algorithm should be used.

It can be determined that the creation of a Machine Learning model, which does not consider time series but does take into account values from a certain number of previous hours. It obtains significant improvements in the range 5-22h and 29-45h, for most pollutants. However, for the pollutants O_3 and PM2.5 there is no considerable difference that compensates for the time invested in obtaining these results.

As future work, the behaviour of the model could be studied if, in addition to using a number of hours from the past, predictions for a certain number of hours in the future are also added.