TRABAJO FIN DE GRADO

DOBLE GRADO EN INGENIERÍA EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS

Estimación del impacto de la contaminación ambiental en la salud mediante técnicas Machine Learning

Autor
Marta Amor Jurado

Directores
Miguel Molina Solana
Rossella Arcucci

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Facultad de Ciencias

—

Granada, 9 de julio de 2021

Estimación del impacto de la contaminación ambiental en la salud mediante técnicas Machine Learning

Autor
Marta Amor Jurado

Directores
Miguel Molina Solana
Rossella Arcucci

Estimación del impacto de la contaminación ambiental en la salud mediantes técnicas Machine Learning

Marta Amor Jurado

Palabras clave: contaminación, descomposición estacional, alisamiento exponencial, ARIMA, LSTM

Resumen

La contaminación es uno de los principales problemas que existen hoy en día. Las partículas que la conforman pueden causar efectos adversos en la salud de personas y animales. Sabiendo cuales son y su comportamiento a lo largo del tiempo para varias zonas, tratamos de encontrar un modelo que sea capaz de predecir los valores que alcanzarán en un futuro. De esta forma, se podrán establecer planes o pautas futuras para buscar una solución.
Se han utilizado cuatro técnicas de predicción distintas: descomposición estacional, alisamiento exponencial, ARIMA y LSTM. Dentro de cada técnica se han realizado varias pruebas con distintos modelos y valores. Para comprobar que técnicas dan mejores resultados hemos usado el error cuadrático medio. Cada experimiento se ha llevado a cabo teniendo en cuenta la zona en la que se habían realizado las mediciones y para cada una de las partículas que se habían observado. Este estudio sugiere que el método que más se ajusta a los valores reales es uno de los implementados para la descomposición estacional.

Estimation of the impact of environmental pollution on health using Machine Learning techniques.

Marta, Amor Jurado

Keywords: pollution, seasonal descomposition, exponential smoothing, ARIMA, LSTM

Abstract

Pollution is one of the main problems that exist today. The particles that make it up can cause adverse effects on the health of people and animals. Knowing what they are and their behavior over time for various areas, we try to find a model that is able to predict the values they will reach in the future. In this way, it will be possible to establish future plans or guidelines to find a solution. Four different forecasting techniques have been used, seasonal decomposition, exponential smoothing, ARIMA and LSTM. Within each technique several tests have been performed with different models and values. To check which techniques give better results we have used the mean square error. Each experiment was carried out taking into account the area in which the measurements were taken and for each of the particles observed. This study suggests that the method that best fits the real values is one of those implemented for seasonal decomposition.

Yo, **Marta Amor Jurado**, alumna de la titulación Doble Grado en Ingeniería Informática y matemáticas de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77340831L, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Marta Amor Jurado

Granada a 7 de julio de 2021

D. **Miguel Molina Solana (tutor1)**, profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Dña. **Rossella Arcucci (tutor2)**, investigadora del Departamento de Computación del Imperial College London, Reino Unido.

Informan:

Que el presente trabajo, titulado ***Título del proyecto***, ha sido realizado bajo nuestra supervisión por **Marta Amor Jurado (alumna)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 7 de julio de 2021.

El director:

Miguel Molina Solana (tutor1)

Agradecimientos

Agradezco a la universidad de Granada y a todos los profesores que he tenido, tanto dentro de ella como fuera, por haber fomentado el desarrollo de mi curiosidad y la formación que me han procurado durante estos años.
Tambien, agradecer a todos aquellos que me han formado como persona y profesional.
Por último, agradecer a mi familia que me ha dedicado su tiempo, su esfuerzo y todos los recursos necesarios para educarme y formarme.

Capítulo 1

Introducción

Durante millones de años, la humanidad ha vivido teniendo un impacto mínimo sobre el planeta. 15.000 años atrás empezaron las primeras talas de árboles, pero no sería hasta los siglos XVI y XVII que aparecieron las primeras crisis energéticas por la escasez de leña y madera. Sin embargo, fue durante el siglo XVIII cuando apareció un auténtico factor que marcaría el antes y el después en la contaminación ambiental: el descubrimiento de combustibles fósiles.
La Revolución Industrial supuso una gran revolución para el hombre ya que cambió su manera de producir, consumir e incluso viajar. Como consecuencia de todo esto, aumentó la cantidad y la variedad de agentes contaminantes liberados al ambiente. La invención de medios de transporte como la locomotora y los trenes dispararon el uso de carbón y con ello, un enorme aumento en la emisión de dióxido de carbono, vapor de agua, óxidos de azufre y otros productos volátiles. Además, la fundición de metales también contribuyó en gran medida a la generación de gases.
Con la primera expansión industrial se crearon zonas de contaminación, las cuales se pueden encontrar sobre todo en grandes concentraciones urbanas. Durante el último siglo, se ha relacionado la contaminación existente en el aire con problemas en la salud de las personas [22].En los últimos 100 años los altos niveles de partículas contaminantes en el aire se han asociado con episodios de excesos de mortalidad. Hoy en día, pese a la concienciación de la población y la reducción de emisiones de gases, la contaminación sigue siendo cada vez mayor en las ciudades. Debido al impacto tan negativo que tiene en la humanidad, es necesario estudiar la evolución durante los últimos años de estos niveles para poder predecir así niveles futuros con el objetivo de plantear políticas públicas y actuaciones adecuadas.

Este trabajo persigue precisamente el estudio de diversas técnicas matemáticas y su implementación computacional para modelar datos históricos de niveles de contaminación, así como su estimación a futuro.

El estudio nace de la necesidad de controlar y predecir la evolución de las distintas partículas que se encuentran en el ambiente y son perjudiciales para el ser humano.
Es de suma importancia analizar el comportamiento de los distintos elementos, intentando encontrar pautas y tendencias anuales o mensuales. Además, vamos a hacer uso de varios modelos clásicos y avanzados de predicción, analizando los resultados obtenidos y comprobando la efectividad de cada uno de ellos.

Hemos trabajado con datos de la ciudad de Londres, puesto que el director de este trabajo es también investigador del Imperial College de Londres y que esta ciudad poseee un repositorio público de datos en abiertos sobre ella. Los datos recogen mediciones sobre diversas partículas que se han encontrado en el ambiente en varias zonas de Londres durante los últimos años.

Para poder utilizar esta información es necesario tratarla y realizar estudios sobre ella. Una vez que tenemos los datos preparados, haremos un estudio generalizado sobre ellos, viendo con distintas gráficas su comportamiento dependiendo de la zona en la que se encuentran y el momento del tiempo. Además, buscaremos como se relacionan las distintas partículas entre ellas.

Las técnicas de predicción que hemos usado para estimar la contaminación en los próximos años han sido descomposición estacional junto a otros modelos, alisamiento exponencial, ARIMA, SARIMA y modelos LSTM. Los cuáles pasamos a explicar más detalladamente en capítulo 3 de metodología. Con cada técnica hemos probado distintos métodos y con cada método distintos parámetros, para así poder compararlas y encontrar cual nos da un mejor resultado para nuestro estudio.

Tras realizar todas las pruebas descritas en el capítulo 5, sorprendentemente hemos obtenido que para las dos zonas en las que realizamos el estudio y para todas las partículas, obtenemos los mejores resultados con la técnica de descomposición estacional.
Para llegar a esta conclusión, hemos recogido el error cuadrático medio generado en cada caso y lo hemos comparado entre los distintos métodos y técnicas.

Se elige este tema como trabajo de fin de grado del doble grado en Ingeniería Informática y Matemáticas, el cuál tiene 18 créditos.
Como no puede ser de otra manera, este trabajo incluye tanto contenidos informáticos como matemáticos.

Está formado por una introducción en la que se relata el problema exis-

tente y de forma general los pasos a seguir en el trabajo.

Tiene un segundo capítulo de antecedentes en el que se describen los distintos trabajos de investigación que preceden al que se está realizando. Además, se describen los datos con los que trabajamos y qué es el Aprendizaje Automático junto a varias de sus técnicas.

El tercer capítulo explica los distintos métodos que se han usado en la práctica, tanto de limpieza de datos como de predicción.

Todos los experimentos llevados a cabo los mostramos en el capítulo cuarto, incluyendo el código generado para ello.
Finalmente, en el quinto y último capítulo explicamos los resultados y conclusiones a las que hemos llegado tras todo el proceso, asi como planteamos futuras líneas de trabajo que pueden seguir desarrollandose.

Este trabajo ha supuesto un total de 752 horas, las cuáles se ven por tares y temporización en el digrama de Gantt que podemos ver en la figura 1.1. Al haber realizado el trabajo fin de grado compaginandolo con el trabajo, los días que se muestran en el diagrama corresponden a un trabajo de 4 horas en lugar de 8 horas.

WBS	Task Name	Start	Finish	Duration
1	**Lectura y comprensión de trabajos previos**	**ju 15/oct/20**	**vi 20/nov/20**	**27**
1.1	Towards whole-lung simulations of aerosol deposition: A model of deep lung	ju 15/oct/20	lu 26/oct/20	8
1.2	Computational modeling of lung deposition of inhaled particles in chronic obstructive	ma 27/oct/20	mi 28/oct/20	2
1.3	Predicting the clinical effects of a short acting bronchodilator in individual patients	ju 29/oct/20	ma 03/nov/20	4
1.4	Substance deposition assessment in obstructed pulmonary system through	mi 04/nov/20	vi 06/nov/20	3
1.5	No evidence that electric charge increseases inhaled ultrafine particle deposition in	lu 09/nov/20	ma 10/nov/20	2
1.6	Inhaled aerosol distribution between proximal bronchi and lung	mi 11/nov/20	vi 13/nov/20	3
1.7	Inhaled aerosol distribution in Human Airways: A Scintigraphy-Guided Study in a 3D	lu 16/nov/20	ma 17/nov/20	2
1.8	Regional lung deposition and bronchodilator response as a function of β2–Agonist particle	mi 18/nov/20	vi 20/nov/20	3
2	**Datos**	**lu 23/nov/20**	**ju 31/dic/20**	**29**
2.1	Toma de contacto con los datos. Estudio de cada parte y particula	lu 23/nov/20	mi 25/nov/20	3
2.2	Preprocesamiento del conjunto de datos	ju 26/nov/20	ju 03/dic/20	6
2.3	Análisis del conjunto datos con distintas visualizaciones	vi 04/dic/20	ju 31/dic/20	20
3	**Predicción**	**vi 01/ene/21**	**ju 11/mar/21**	**50**
3.1	Descomposición Estacional	vi 01/ene/21	lu 18/ene/21	12
3.2	Alisamiento exponencial	ma 19/ene/21	lu 08/feb/21	15
3.3	ARIMA	ma 09/feb/21	ju 18/feb/21	8
3.4	LSTM	vi 19/feb/21	ju 11/mar/21	15
4	**Memoria**	**vi 12/mar/21**	**lu 05/jul/21**	**82**
4.1	Introducción	vi 12/mar/21	ma 16/mar/21	3
4.2	Antecedentes	mi 17/mar/21	lu 29/mar/21	9
4.3	Metodología	ma 30/mar/21	lu 26/abr/21	20
4.4	Experimentos: Preprocesamiento y modelado	ma 27/abr/21	lu 24/may/21	20
4.5	Experimentos: Predicción	ma 25/may/21	lu 21/jun/21	20
4.6	Conclusiones	ma 22/jun/21	vi 25/jun/21	4
4.7	Otros	lu 28/jun/21	lu 05/jul/21	6

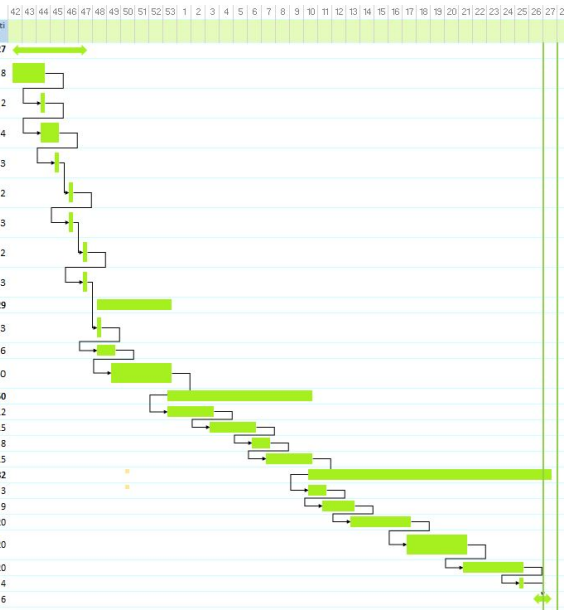

Figura 1.1: Diagrama de Gantt con tareas y temporización.

Capítulo 2

Antecedentes

En este capitulo vamos a hacer un breve resumen de los distintos trabajos de investigación revisados y que guardan relación con el tema a tratar: cómo afectan a la salud de las personas los distintos tipos y tamaños de partículas contaminantes.

También se explican algunos conceptos necesarios relacionados con el machine learning y el tratamiento de datos.

Además de esto, vamos a ver detalladamente el conjunto de datos del que partimos y las características de cada una de las partículas de las que hacemos el estudio y predicción.

2.1. Trabajos previos

En primer lugar vamos a hablar del trabajo ***Regional lung deposition and bronchodilator response as a function of*** $\beta 2 - Agonist$ ***particle size*** [26] fue publicado en septiembre de 2005 y realizado por Omar S.Usmai, Martyn F.Biddiscombe y Peter J. Barnes, tiene como objetivo optimizar la administración del broncodilatador inhalado. Se plantea la hipótesis de dirigir el albuterol a las vías respiratorias regionales modificando el tamaño de las partículas que se inhalan.
Se trata de un estudio aleatorio, doble ciego y controlado con placebo, en el que 12 sujetos con asma inhalan aerosoles monodispersos de albuterol marcados con tecnecio-99m con diámetros de 1,5 μm, 3 μm y 6 μm con distintos flujos de inspiración que son de 30-60 L/min y superiores a 60 L/min. Para cuantificar la deposición de radioaerosoles pulmonares y extratorácivos se usa gammagrafría plana. Se evaluaron tanto la función pulmonar como las mediciones de tolerabilidad. También se compara la eficacia clínica entre el

albuterol monodisperso no marcado (dosis de 15 μmg) y el albuterol de 200 μg en inhaladores de dosis medida (IDM).
Se observo que las partículas más pequeñas tenían una mayor deposición total pulmonar y pulmonar periférica, además de una mayor penetración en las vías respiratorias distales. Sin embargo, las partículas más grandes, las dosis de 30 μg, resultaron ser más eficaces consiguiendo una mayor broncodilatación que la obtenida por el albuterol MDI de 200 μg. Aunque las partículas pequeñas se exhalaron más, la deposición orofaríngea fue mayor con las partículas más grandes. Los flujos de inspiración más rápidos disminuyeron la deposición pulmonar total, pero aumentaron la deposición orofaríngea para las partículas mayores. También se vió un cambio en la distribución de los aerosoles hacía las vías respiratorias en todos los casos.
Se concluyó que la orientación regional de la inhalación de $\beta 2 - Agonist$ hacia las vías respiratorias próximas es más importante que la deposición en los alveolos distales de los broncodilatadores. La alteración de la deposición intrapulmonar mediante el tamaño de las partículas de los aerosoles puede mejorar el tratamiento farmacológico inhalado y puede tener implicaciones para el desarrollo de futuros tratamientos inhalados.

Cinco años después, encontramos otro estudio con pacientes para ver la respuesta del broncodilatador e intentar predecir las respuestas de este. El trabajo ***Predicting the clinical effects of a short acting bronchodilator in individual patients using artificial neural networks*** [5] de Marcel de Matas et al. y publicado en septiembre de 2010 tiene como objetivo la investigación es el desarrollo de modelos capaces de predecir respuestas del broncodilatador al sulfato de salbutamol en individuos que siguen una administración de fármacos en un régimen de dosis acumulativo.
Los datos usados para el modelo Artificial Neuronal Network (ANN) son recogidos en publicaciones previas de Usmani et al. (2003) y Usmani (2005). Se realiza una monodispersión de aerosoles de 1,5 μm, 3 μm y 6 μm generadas usando STAG (Spinning Top Aerosol Generator). Cada una de las variantes de tamaño se suministran en dosis de 10 μg, 20 μm, 40 μm y 100 μm dadas cada 0,30,60 y 90 minutos respectivamente, a 18 pacientes con un nivel de asma leve-moderado en un estudio controlado con placebo de forma aleatoria. Se modelan datos tanto in vivo como in vitro, de los tres tipos de tamaños de particulas administradas a los pacientes, usando el software ANN disponible (INFORM v3.4 de Intelligensys, UK).
Se desarrollaron dos modelos de redes neuronales para predecir las respuestas del broncodilatador a los 10 minutos y 20 minutos después de cada dosis. Para que el modelo fuese capaz de predecir se requirieron 14 nodos. Los resultados del estudio sugieren que los parámetros del volumen de los pulmones y las dimensiones de las vías respiratorias son probablemente responsables de la variación de deposiciones de sulfato de salbutamol en los pulmones.
La conclusión del trabajo es que los ANNs son capaces de desarrollar mode-

los que pueden predecir resultados clínicos en sujetos individuales desde el conocimiento de las propiedades de los aerosoles in vitro y las características del paciente. Las técnicas usadas muestran una buena predicción en las respuestas clínicas para la monodispersión de aerosoles de sulfato de salbutamol. La respuesta al broncodilatador muestra relación con factores de los pacientes como la edad, REV (%) y la superficie del área del cuerpo.

En el año 2016 encontramos otros dos trabajos muy relacionados con los anteriores, ya que intentan mejorar la comprensión de las partículas en los pulmones e intentan desarrollar modelos que sigan este flujo.

El primero de ellos es el trabajo ***Inhaled aerosol distribution in Human Airways: A Scintigraphy-Guided Study in a 3D printed model*** [29], de Abril de 2016, tiene como objetivo desarrollar un modelo pulmonar integrado, teórico y pulmonar. De esta forma se pretende estudiar si los aerosoles siguen un flujo de distribución del aire en las personas.
Para ello, se realiza una impresión en 3D de las vías respiratorias humanas. Tras esto, se cuantifica la deposición y distribución de las partículas. También se visualiza la trayectoria que sigue la inhalación. El molde creado se expuso a partículas de aerosol monodispersas y radiomarcadas de 6 μm con distintas velocidades de inhalación. Se obtuvieron imágenes mediante gammagrafía en 2D. También se consiguió una imagen de la distribución de los aerosoles después de las vías respiratorias. Se consiguió imitar los patrones experimentales de deposición de aerosoles con la simulación computacional de dinámica de fluidos (CFD) en la misma geometría de las vías respiratorias en 3D.
Se pudo demostrar que en las partículas con diámetros de 6 μm inhaladas en flujos de hasta 60 L/min, la distribución de los aerosoles tanto en los pulmones como en los cinco lóbulos pulmonares individuales seguían aproximadamente las distribuciones de los flujos de aire. Las deposiciones de los aerosoles fueron mayor en los bronquios del pulmón izquierdo frente al derecho, y en el pulmón derecho frente al izquierdo.
Por tanto, se demostró la combinación de experimentos de imagen junto a simulaciones CFD para estudiar los patrones de deposición de aerosoles en las vías respiratorias hasta la generación 5.

El segundo trabajo que hemos comentado antes es ***Substance deposition assessment in obstructed pulmonary system through numerical characterization of airflow and inhaled particles attributes*** [14], pubilicado en diciembre de 2016 nos conduce a una investigación inicial para mejorar la comprensión de la deposición de las partículas en los pulmones. Se desarrollan modelos para las obstrucciones de las vías respiratorias relacionados con enfermedades pulmonares. Se realiza a una exhaustiva evaluación

del comportamiento del flujo del aire junto con algunas caracteristicas de las partículas inhaladas, como son la densidad y el tamaño. El objetivo es tener un tratamiento efectivo y personalizado para los pulmones, dependiendo de la geometría de estos y la comprensión de los flujos de aire dentro de las vías respiratorias.
Para ello, se ha utilizado una técnica de procesamiento geométrico que incluye algoritmos de contracción, los cuales se utilizan para emplear las diferentes disposiciones respiratorias asociadas a las enfermedades pulmonares que presentan obstrucciones en las vías respiratorias. Se examinan tres casos, el primero de ellos con los pulmones normales, el segundo con modelos de obstrucciones en ambos pulmones y el tercer modelo con estrechamientos solo en el pulmón derecho. A continuación, se elaboran hipótesis precisas sobre el flujo de aire y la fracción de deposición (FD) en varias secciones de los pulmones, simulando estos distintos incidentes mediante el método de volúmenes finitos (MVF), en particular los algoritmos CFD y FPT. Además, se utiliza un análisis paramétrico detallado para aclarar los efectos del tamaño y la densidad de las partículas en términos de deposición regional sobre varias partes del sistema pulmonar. De esta forma, se consigue evaluar el patrón de deposición de varias sustancias.
Los resultados mostraron para el caso del modelo con los pulmones normales, que la mayor parte de las partículas se encontraron en el pulmón derecho. Esto también ocurre para el caso en el que ambos pulmones están obstruidos de manera simétrica. Sin embargo, para el último caso, en el que las obstrucciones solo se encuentran en el pulmón derecho, la mayoría de las partículas se encontraron en el pulmón izquierdo, lo que nos indica que cuando inhalamos la medicación, esta se deposita lejos de las zonas inflamadas. Además, se demuestra que las partículas con diámetros entre $1\mu m$ y 10 μm se suelen depositar en la parte inferior de las vías respiratorias. Sin embargo, las partículas con un diámetro entre 20 μm y 30 μm se suelen depositar en la parte superior. Como resultado, el estudio indica un aumento en los níveles de DF en la parte superior de las vías respiratorias con el aumento del diámetro de las partículas. Cuando la densidad de las partículas aumenta, el DF también aumenta.
Se puede conseguir un avance en los tratamientos inhalatorios para enfermedades repiratorias haciendo uso clínico de simulaciones CFD y FPT, y mediante la evaluación de las desposiciones de las partículas inhaladas.

Tras estos trabajos, vemos como poco a poco se va mejorando y consiguiendo mejorar los tratamientos y la comprensión de como actuan las partículas dentro de nuestras vías respiratorias y nuestros pulmones.

Hemos encontrado en este pasado tres trabajos nuevos y que comple-

mentan a los vistos.

El trabajo de Omar S. Usmani publicado en enero de 2020, ***No evidence that electric charge increseases inhaled ultrafine particle deposition in Human Lungs*** [27], tiene como objetivo comprobar que la hipótesis de las partículas ultrafinas de aerosoles (100 nm en tamaño) que llevan una carga eléctrica podrían incrementar la deposición de particulas en los pulmones adultos, y como los iones corona procedentes de HVPLs podrían influenciar en el riesgo de enfermedades hemato oncológicas en la infancia. Por tanto, se quiere examinar los efectos de la carga eléctrica en partículas más pequeñas de 300 nm en los pulmones humanos con técnicas in vivo.
La investigación se realizó con 8 pacientes saludables, no fumadores con una espirometría normal. Los participantes inhalan (99m Tc-labeled) Technegas partículas cargadas positivamente en dos sesiones, y partículas con carga neutral en otras dos. Se obtienen las médidas de la concentración de las partículas cargadas y descargadas de cada sujeto midiendo la eficiencia de las deposiciones de dichas partículas. El resultado de la medida obtenida es el índice de penetración, para determinar la extención de los inhalados en los pulmones y su asentamiento en las diferentes regiones del pulmón. Se usó ANOVA para analizar las diferencias entre los dos estados de carga de las partículas.
Una comparación de las partículas cargadas y descargadas en términos de PI y DF(p) no muestran evidencias que sostengan la hipótesis de que las cargas eléctricas (de la magnitud aplicada) incrementen la eficiencia de deposiciones de las desposiciones de partículas ultrafinas. PI no muestra diferencia entre ambos tipos de partículas y DF(p)s también son similares para ambas. Por tanto, los resultados no sostienen que los mecanimos de çorona ionesïncrementen el riesgo de leucemia en niños ni en enfermedades hemato oncológicas en adultos que viven cerca de HVPLs. Si una partícula lleva una carga elétrica positiva, el aumento de deposición en los pulmones podría ocurrir por el proceso físico de çarga de imagen".

Un mes después, en febrero de 2020 se publica el trabajo ***Towards whole-lung simulations of aerosol deposition: A model of deep lung*** [13], desarrollado por P.G. Koullapis, F.S. Stylianou, B. Olsson y S.C. Kassinos y que tiene como objetivo conocer la cantidad y el lugar donde se depositan los aerosoles en las distintas parte del pulmón. Esto es importante para poder predecir la eficacia de algunos aerosoles farmacéuticos o el impacto de la contaminación en nuestros pulmones. Para ello, hacen uso de varios modelos parciales del pulmón, con la intención de extrapolar los resultados en ellos a todo el pulmón.
Se pretende evaluar si un pequeño grupo de modelos sobre el pulmón profundo (DLM) se podrían utilizar para predecir la deposición de las distintas

partículas en el pulmón. Como objetivo final, se tiene la integración del DLM con algunos modelos derivados de imágenes de la parte superior de las vías respiratorias y así conseguir predecir las deposiciones en el pulmón completo.
Los métodos implicados en este estudio son varios. El modelo geométrico es el que representa el pulmón profundo. Es un modelo cerrado que incluye un árbol de bifurcación con 10 generaciones en conjunto a múltiples sub-acinus

Por último, el trabajo publicado publicado en mayo de 2020 por la división respiratoria del hospital universitario UZBrussel en Bruselas titulado ***Inhaled aerosol distribution between proximal bronchi and lung*** [28]. El objetivo del estudio es cuantificar la cantidad de aerosoles inhalados en el centro de los puntos calientes de los pacientes con un nivel de asma leve-moderado, para distintos tamaños de partículas y los flujos de respiración que cubre los rangos aerodinámicos de los inhaladores comunes usados en prácticas clínicas. Además, examinan la capacidad que tienen aquellos aerosoles que escapan del efecto de filtro de los bronquios para alcanzar la región distal del pulmón.
Para la realización de la investigación se aplicaron técnicas de análisis de imágenes. De esta forma, se realizan imágenes escintigráficas de los pulmones para 12 pacientes asmáticos, con inhalaciones de 1,5 μm, 3 μm y 6 μm partículas de fármacos monodispersados en flujos respiratorios de 30 y 60 L-min. Para cada paciente se contruyen 7 imágenes: la primera de ellas con exploración de ventilación de Krypton y las restantes exploraciones de las deposiciones de los aerosoles en 6 días distintos.
Los resultados del análisis de imágenes se valido comparando la ventilación pulmonar con la intensidad del pulmón izquierdo obtenidos con Krypton y aerosoles, y en ambos fue similar. El método isocontous ROI también se comportó como se esperaba, y mostró una correlación directa entre el isocontour ROI en el área del pulmón derecho para el Krypton y el FVC. Podemos considerar como el máximo contorno externo que se puede alcanzar con los aerosoles al área ROI del pulmón derecho con el gas Krypton. La intensidad de los puntos calientes incrementa con el incremento del tamaño de las partículas y los flujos de respiración más rápidos. La mitad de la dosis de los pulmones se retiene en el centro de las vías respiratorias de los puntos calientes para las partículas de 6 μm inhaladas en 60 L/min. La porción del pulmón derecho que alcanzan las partículas inhaladas decrementa con el incremento del tamaño de las partículas y con flujos de respiración más grandes.
Los resultados obtenidos tienen una gran relevancia para la práctica clínica diaria. Una porción substancial de la dosis pulmonar de aerosoles de los inhaladores actuales se depositarán cerca de los bronquios. Por tanto, se reduce el área del pulmón alcanzada por los aerosoles. Los pacientes con ASMA y COPD requerirán partículas más pequeñas inhaladas en flujos lentos. Las

deposiciones más próximas a los bronquios son mejores para inhaladores secos (DPI) que para aerosoles nebulizados. La monodispersión de los aerosoles con adecuados para una evaluación de la deposición pulmonar, los puntos calientes son más prominentes con el aumento de tamaño de las partículas y la velocidad de flujo. Además, se intesifican cuanto más grave sea la enfermedad del paciente.
Este trabajo se relaciona con el estudio en curso, ya que hace una investigación sobre como el tamaño de las distintas partículas puede afectar a los pulmones.

Como hemos podido ver en todos estos trabajos, se consigue un progeso notable tanto en el área de estudio como en la práctica.
Se han utilizado diferentes técnicas como el procesamiento geométrico, análisis de imágenes, impresiones 3D y redes neuronales artificiales entre otras.
Cada técnica y modelo aportan información y se ha conseguido una mejor comprensión sobre el flujo del aire y las partículas dentro de nuestro cuerpo.

Gracias a todo esto, podemos ver como afectan los distintos tipos y tamaños de partículas a las personas sanas y a las personas con enfermedades como ASMA o COPD.
Una de las cosas más importantes conseguidas con estos trabajos, es la mejora en los tratamientos inhalatorios.

Todo esto es importante para nuestro trabajo, ya que nos hacemos una idea de como pueden afectar las disintas cantidades y los distintos tamaños de cada materia a nuestro organismo.
Hemos visto las distintas técnicas y modelos que se han utilizado y desarrollado para intentar predecir los resultados de las deposiciones.

2.2. Machine Learning

Esta sección está basada en las ideas de [19]. La Inteligencia Artifical es la disciplina científico-ténica que se ocupa de la comprensión de los mecanismos subyacentes en el pensamiento y la conducta inteligente y su incorporación en las máquinas. Hoy en día podemos encontrar la I.A. en inumerables campos de estudio. Algunos ámbitos en la que está presente es en robótica, computación ubicua, representación del conocimiento y sistemas empresariales, entre otros.

Machine learning o aprendizaje automático es una disciplina de la Inteligencia Artificial cuyo objetivo es que las máquinas o sistemas aprendan automáticamente. Esto significa, que son capaces de identificar patrones

complejos en los datos y conseguir hacer predicciones sobre estos.
Los fundamentos del machine learning se encuentran en las matemáticas, en especial en el campo de la estadística.

Podemos diferenciar tres tipos de técnicas de machine learning: Algoritmos supervisados, algoritmos sin supervisión y algoritmos por refuerzo.

- Aprendizaje supervisado. La máquina aprende mediante el ejemplo. Se le da al algoritmo un conjunto de datos junto con las etiquetas que debería dar como salida. El algoritmo es el encargado de encontrar que funciones o métodos utilizará para conseguir esos datos. El algoritmo es capaz de identificar patrones, aprende de las observaciones y hace predicciones. Una vez que predice, se le corrige y vuelve a realizar una predicción. Este proceso continúa hasta que se decide que el algoritmo ha alcanzado los objetivos de precisión y rendimiento buscados.

- Aprendizaje no supervisado. No hay un supervisor que ayude al algoritmo. Es el propio algoritmo el encargado de estudiar los datos para identificar los patrones que contenga, y es capaz de determinar las correlaciones que existen entre los datos. En este tipo de aprendizaje, el algoritmo interpreta los datos pudiendo organizarlos o agruparlos de alguna forma que le sea más cómoda. Este método se vuelve mejor y más preciso cuantos más datos evalue.

- Aprendizaje por refuerzo. Es un tipo de proceso reglamentado, en el que se da un tipo de algoritmo, unas acciones, unos parámetros y unos valores finales. Este tipo de aprendizaje aprende por ensayo y error. Lo primero que tenemos que hacer es definir unas reglas y el algoritmo será el encargado de buscar distintas alternativas y quedarse con una política óptima.

Los algoritmos más usados en Machine Learning son: algoritmos de regresión, algoritmos bayesianos, algoritmos de agrupación, algoritmos de árboles de decisión, algoritmos de redes neuronales, algoritmos de reducción de dimensión y algoritmos de aprendizaje profundo.

Los algoritmos de regresión modelan la relación existente entre distintas variables. Para conseguir esto, intentan minimizar una medida de error en un proceso iterativo, y así obtener las mejores predicciones posibles.
Este tipo de algoritmos son muy usados en el análisis estadístico.
Los algoritmos de regresión más utilizados son los de regresión lineal y regresión logística.

Los algoritmos bayesianos, como su nombre indica, utilizan el Teorema de probabilidad de Bayes. Utilizando la probabilidad son capaces de predecir una categoria a la que pertenecen, a partir de varias características.
Se usan para problemas de clasificación y de regresión.
Los más utilizados son Naive Bayes, Gaussian Naive Bayes, Multinomial Naive Bayes y Bayesian Network.

Los algoritmos de agrupación pertenecen al grupo del aprendizaje no supervisado. Se usan para poder agrupar los datos de los que no conocemos sus características comunes.
Crean puntos centrales y jerarquías para así poder diferenciar los grupos y descubrir las características comunes. Conseguirán esto usando las medidas de distancia.
Los más utilizados son K-Means, K-Medians y Hierarchical Clustering.

Los algoritmos de árboles de decisión modelan las distintas opciones que hay en los valores de los atributos que tienen nuestros datos. Suelen usarse para clasificar información, bifurcando y modelando los distintos caminos posibles y la probabilidad de ocurrencia de cada uno de ellos, para así mejorar su precisión.
Los Algoritmos de árbol de decisión más usados son Árboles de Clasificación y Regresión (CART), Decisión de Árbol condicional y Random Forest.

Los algoritmos de redes neuronales son algoritmos y estructuras inspirados en las redes neuronales reales. Son útiles en un gran número de probamáticas, aunque suelen usarse para problemas de clasificación y regresión. Son muy buenas para detectar patrones, aunque necesitan una gran capacidad de procesamiento y memoria.
Las redes neuronales básicas y clásicas son compuerta XOR, Perceptron , Back-Propagation, Hopfield Network y MLP: Multi Layered Perceptron.

Los algoritmos de aprendizaje profundo son la evolución de las redes neuronales artificiales. Aprovechan la mejora en la capacidad de ejecución, memoria y disco para así poder analizar una gran cantidad de datos en enormes redes neuronales e interconectarlas en diversas capas, para así ejecutarlas en paralelo.
Los algoritmos más populares de Deep Learning son Convolutional Neural Networks y Long Short Term Memory Neural Networks.

Los algoritmos de reducción de dimensión se tratan de algoritmos no supervisados. Intentan explotar la estructura real para poder simplificar los datos y reducirlos o comprimirlos. Son muy útiles para visualizar datos o simplificar las variables.
Los más utilizados son Principal Component Analysis (PCA) y t-SNE.

2.3. Conjunto de datos

Los datos que usaremos en este trabajo provienen de la página `https://data.london.gov.uk/dataset/london-average-air-quality-levels`. Dicho conjunto de datos muestra lecturas medias de particulas contaminantes en dos zonas, Roadside y Background, de la ciudad de Londres desde enero de 2008 hasta julio de 2019. Las partículas concretas son de Óxido Nítrico, Dióxido de Nitrógeno, Óxidos de Nitrógeno, Ozono, Dióxido de azufre y de partículas de 10 μmg y 2,5 μmg.

El hacer un estudio sobre estas partículas es debido al gran impacto negativo que tienen en la salud de las personas y en el medio ambiente, tal y como hemos visto en capítulos anteriores.

La zona de Roadside [9], son lugares que están entre 1 y 5 metros de una carretera transitada y suelen tener una altura entre 2 y 3 metros. Podemos ver los cambios en la concentración de la contaminación atmosférica procedentes del tráfico y son útiles para poder identificar los puntos críticos de la calidad del aire y que pueden tener un impacto potencialmente negativo sobre la salud de los peatones.

La zona de Background [9] están más alejados de las fuentes de emisiones y están influenciados por varias fuentes de contaminación. Son una buena representaicón de las concentraciones de fondo de la ciudad.

Vamos a ver detalladamente cada una de las partículas.

Ozono: El ozono O_3, como podemos ver en [6], es un gas incoloro que se encuentra en el ambiente. Podemos diferenciar dos tipos de Ozono uno bueno y otro malo.
El beneficioso es aquel que nos protege de los rayos ultravolieta que provienen del sol y se encuentra en la estratosfera.
El ozono dañino es el que se encuentra al nivel del suelo y se forma con los gases contaminantes de coches, fábricas y otros. Suele ser peor en los meses de verano y su inhalación puede traer consecuencias negativas para la salud, ya que provoca tos, irritación de garganta y un empeoramiento de algunas enfermedades como asma o bronquitis. Además, una exposición prolongada incrementa tanto la mortalidad respiratoria como posiblemente la cardiovascular.

Dióxido de Nitrógeno y Óxidos de Nitrógeno: Los óxidos de nitrógeno, como explican en [21] están compuestos por óxido nítrico (NO) y dióxido de nitrógeno (NO_2), y se forman durante la combustión. El monóxido de nitrógeno y el dióxido de carbono son los más tóxicos.

Se liberan al aire mediante el escape de vehículos motorizados, combustión de carbón, petróleo o gas natural, y otros procesos. Estar expuestos a niveles bajos de estas sustancias pueden causar irritación en diversas partes del cuerpo, incluidos los pulmones, además pueden producir falta de aliento, cansancio y naúseas. Si los niveles son altos, estos pueden producirnos quemaduras, espasmos y dilatación de algunos tejidos reduciendo la oxigenación de estos y produciendo una acumulación de líquido en los pulmones, pudiendo llevarnos hasta la muerte.

Dióxido de azufre: El dióxido de azufre SO_2, cuya descripción vemos en [20], es un gas incoloro e irritante. Durante su proceso de oxidación en la atmosfera produce unos sultatos que forman parte de las partículas finas $PM10$. Cuando hay humedad este compuesto forma ácidos en forma de aerosoles, produciendo una parte importante de las particulas finas $PM2.5$. Su producción es debida a la combustión de carbón y petróleo, y por fuentes naturales como los volcanes. Tiene un impacto negativo en el ser humano ya que la exposición a este gas puede producir irritación en los ojos, inflamación en las vías respiratorias, dificultad para respirar , edema pulmonar y paro cardíaco entre otros. También se asocia a problemas de asma y bronquitis crónica, incrementando la mortalidad en personas mayores y niños.

Partículas PM10 y PM2.5: Como explican en [12], las partículas en suspensión, PM, hacen referencia a las distintas sustancias que se encuentran en la atmósfera. Se denominan partículas gruesas si su tamaño es igual o menor a $10\mu m$ y se denominan particulas finas si su tamaño es igual o menor que $2.5\mu m$. Pueden crearse de forma natural por volcanes, arena del desierto, polen, entre otros, o mediante un proceso humano como es la quema de combustibles. Existen números estudios que asocian estas partículas con las principales causas de mortalidad mundial, como son las enfermedades isquémicas de corazón, accidentes cerebrovasculares y enfermedades pulmonares obstructivas crónicas.

En las siguiente tablas que obtenemos de [4] vemos como de peligrosos son los valores de las distintas partículas que estudiamos y como afectan a las personas la exposición de estas.

Band	Index	Ozone Running 8 hourly mean μgm^{-3}	Nitrogen Dioxide hourly mean μgm^{-3}	Sulphur Dioxide 15 minute mean μgm^{-3}	$PM_{2.5}$ Particles (EU Reference Equivalent) 24 hour mean μgm^{-3}	PM_{10} Particles (EU Reference Equivalent) 24 hour mean μgm^{-3}
Low	1	0-33	0-67	0-88	0-11	0-16
	2	34-66	68-134	89-177	12-23	17-33
	3	67-100	135-200	178-266	24-35	34-50
Moderate	4	101-120	201-267	267-354	36-41	51-58
	5	121-140	268-334	355-443	42-47	59-66
	6	141-160	335-400	444-532	48-53	67-75
High	7	161-187	401-467	533-710	54-58	76-83
	8	188-213	468-534	711-887	59-64	84-91
	9	214-240	535-600	888-1064	65-70	92-100
Very High	10	241 or more	601 or more	1065 or more	71 or more	101 or more

Grupo	Índice de riesgo	Individuos con riesgo	Población General
Bajo	1 2 3	Disfrutan de sus actividades usuales al aire libre.	Disfrutan de sus actividades usuales al aire libre.
Moderado	4 5 6	Los adultos y niños con problemas pulmonares y adultos con problemas cardiacos, deberían considerar reducir la actividad física que les agote, en especial en sitios al aire libre.	Disfrutan de sus actividades usuales al aire libre.
Alto	7 8 9	Los adultos y niños con problemas pulmonares y adultos con problemas cardiacos, deberían reducir la actividad física que les agote, y en especialmente si experimentan síntomas. La gente con asma podría necesitar utilizar su inhalador más frecuentemente. Las personas mayores deberían también reducir la ejercitación física.	Cualquiera que experimente incomodidad como dolor de ojos, dolor de garganta y tos debería considerar reducir la actividad, en especial al aire libre.
Muy alto	10	Los adultos y niños con problemas pulmonares y adultos con problemas cardiacos y la gente mayor deberían reducir la actividad física. La gente con asma podría necesitar utilizar su inhalador más frecuentemente.	Reducir la ejercitación física, en particular en sitios al aire libre, especialmente si se experimentan síntomas como tos o dolor de garganta.

Capítulo 3

Metodología

En este tercer capítulo vamos a explicar las técnicas usadas y desarrolladas en los experimentos. Es necesario tener un conocimiento sobre cada una de ellas, estudiando como surgen y sus fundamentos matemáticos. Vamos a dividir el capítulo en tres secciones.

La primera sección trata sobre el tratamiento del dato antes de su uso para experimentos. Explicamos las diferentes anomalías que podemos encontrarnos y como actuar ante ellas.

A continuación, hay una segunda sección de técnicas de predicción. Dentro de esta sección diferenciamos cada una de las técnicas usadas, explicando en que consisten y como se diferencian entre ellas.
Los algoritmos que vamos a explicar los podemos ver en la tabla 3.1.

Técnica	Método
Descomposición Estacional	Persistencia
Descomposición Estacional	Autorregresión
Descomposición Estacional	Descomposicón con HP y Autorregresión
Descomposición Estacional	Descomposición con STL y Autorregresión
Alisamiento Exponencial	Alisamiento exponencial simple
Alisamiento Exponencial	Hölt
Alisamiento Exponencial	Hölt y Winter - Aditivo
Alisamiento Exponencial	Hölt y Winter - Multiplicativo
ARIMA	ARIMA
LSTM	LSTM

Tabla 3.1: Técnicas de predicción junto a los métodos que vamos a estudiar

Finalmente, en la última sección explicaremos como medir el error generado por los distintas técnicas y métodos de predicción. De esta forma, podremos ser capaces de comparar los resultados obtenidos y encontrar aquel que nos proporciona mejores resultados.

3.1. Preprocesamiento

Esta sección está basada en las ideas de [16]. Uno de los principales problemas al procesar datos o trabajar con ellos es la aparición anomalías y otros errores en sus valores. Además de causarnos problemas en el procesamiento, también nos puede llevar a resultados y conclusiones erróneas.

Estas anomalías pueden tener como origen por mediciones erróneas, cambios que se han producido sobre el conjunto de datos y omisiones entre otras. Estas irregularidades se clasifican en:

- Anomalías sintácticas. Son aquellas que contienen errores léxicos, de formato y dominio.
- Anomalías semánticas. Son aquellas que violan las restricciones de integridad en las tuplas, además pueden contener tuplas inválidas o duplicadas.
- Anomalías de contexto. Son aquellas que contienen datos nulos o contiene ausencia de tuplas completas.

En todo trabajo, antes de tratar los datos necesitamos hacer un análisis sobre estos para ver como son y establecer si es necesario realizar una limpieza sobre estos. Vamos a centrarnos en los valores ausentes, ya que es el problema que nos hemos encontrado en nuestros datos
Por tanto vamos a explicar en que consiste esta clase de anomalía y las distintas posibilidades que tenemos de actuar ante ello.

Anomalía de valor ausente
Suele ocurrir con bastante frecuencia en los conjuntos de datos. Puede producrise por varias causas, entre ellas la ausencia de respuesta por parte del agente o cliente, el fallo en las transcripción de algún dato y el error del sistema que obtiene los datos. La definición distingue entre dos formas de ausencia de datos:

- Datos ausente. Es un valor que no se encuentra en el conjunto de datos.
- Datos nulos. Es la ausencia de un dato porque se desconocce el valor o no tiene sentido para ese objeto.

Cuando hay menos de un 1 % de valores ausentes, podemos considerarlos como triviales y si tenemos entre un 1 % y un 5 % los consideramos manejables. Si el valor aumenta entre un 5 % y un 15 % se requieren métodos sotisficados para manejarlos y si tenemos más de un 15 % afectará seriamente a las interpretaciones.

Para detectar que hay valores erróneos en los datos se suele usar los siguientes métodos:

Métodos estadísticos
Podemos determinar valores irregulares con los métodos de la media, la desviación estándar y rango, basados en el teorema de Chebychev.

Recordamos el teorema de Chebychev [25] y el de Markov [24].

Teorema 3.1. *Teorema de Markov: Desigualdad básica*
Si X es una variable aleatoria no negativa tal que $\exists E[X]$ se tiene

$$P(X \geq \epsilon) \leq \frac{E[X]}{\epsilon}, \forall \epsilon \geq 0$$

Demostración. Si X es una variable discreta y toma valores en E_x

$$E[X] = \sum_{x \in E_x} xP(X = x) = \sum_{x \in E_x / x \leq \epsilon} xP(X = x) + \sum_{x \in E_x / x \geq \epsilon} xP(X = x)$$

1. $X \geq 0 \rightarrow \forall x \in Ex, x \geq 0 \rightarrow \sum_{x \in E_x / x \leq \epsilon} xP(X = x) \geq 0$

2. $\sum_{x \in E_x / x \geq \epsilon} xP(X = x) \geq \sum_{x \in E_x / x \leq \epsilon} \varepsilon P(X = x) = \epsilon \sum_{x \in E_x / x \leq \epsilon} P(X = x) = \epsilon P(X \geq \epsilon)$

Usando 1) y 2) tenemos que

$$E[X] \geq \epsilon P(X \geq \epsilon) \rightarrow \frac{E[X]}{\epsilon} \geq P(X \geq \epsilon), \forall \epsilon \geq 0$$

Si X es continua, con función de densidad $f_x, X \geq 0 \rightarrow f_x(x) = 0, \forall x \leq 0$, y razonando de forma análoga al caso discreto,

$$E[X] = \int_0^{+\infty} x f_x(x) \partial x = \int_0^{\epsilon} x f_x(x) \partial x + \int_{\epsilon}^{+\infty} x f_x(x) \partial x \geq \epsilon \int_{\epsilon}^{+\infty} x f_x(x) \partial x = \epsilon P(X \geq \epsilon)$$

$$\frac{E[X]}{\epsilon} \geq P(X \geq \epsilon), \forall \epsilon \geq 0$$

□

Teorema 3.2. *Teorema de Chebychev*
Si X es una variable aleatoria tal que $\exists E[X^2]$, se tiene

$$P(|X - E[X]| \geq K) \leq \frac{Var[X]}{k^2}, \forall k \geq 0$$

Demostración. Aplicando el teorema de Markov de la desigualdad básica, a la variable $(X - E[X])^2 \geq 0$ y tomando $\epsilon = k^2$, tenemos que

$$P(|X - E[X]| \geq k) = P((X - E[X])^2 \geq k^2) \leq \frac{E[(X - E[X])^2]}{k^2} = \frac{Var[X]}{k^2}$$

□

La principal desventaja de estos métodos es que pueden generar falsos positivos. Sin embargo, pueden ser combinados con otros métodos y son rápidos.

Métodos de agrupamientos
Para identificar los datos con valores anómalos, estos métodos utilizan algoritmos de agrupamiento o clusterización usando alguna distancia, como puede ser la distancia euclidea.
Vamos a ver alguno de ellos tal.

Modelos Jerárquicos Aglomerativos - **Estrategia de la distancia mínima o similitud máxima** [8].

Considera la distancia o similitud entre dos cluster como la distancia mínima o la similitud máxima entre sus componentes.

Si tenemos K cluster ya creados, definimos la distancia entre los clusters C_i y C_j como

$$d(Ci, Cj) = \min_{x_l \in Ci, x_m \in Cj} \{d(x_l, x_m)\} \qquad l = 1, .., n_i \ ; \ m = 1, ..., n_j$$

Para la distancia para el cluster $K + 1$, uniremos los clusters Ci y C_j si

$$d(Ci, Cj) = \min_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \{d(C_{i1}, C_{j1})\} =$$

$$= \min_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \left\{ \min_{x_l \in Ci, x_m \in Cj} \{d(x_l, x_m)\} \right\} \qquad l = 1, .., n_i \ ; \ m = 1, ..., n_j$$

Si tenemos K cluster ya creados, definimos la similitud entre los clusters C_i y C_j como

$$s(Ci, Cj) = \max_{x_l \in Ci, x_m \in Cj} \{s(x_l, x_m)\} \qquad l = 1, .., n_{i1} \ ; \ m = 1, ..., n_{j1}$$

Para la similitud para el cluster $K+1$, uniremos los clusters Ci y C_j si

$$s(Ci, Cj) = \max_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \{s(C_{i1}, C_{j1})\} =$$

$$= \max_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \left\{ \max_{x_l \in Ci, x_m \in Cj} \{s(x_l, x_m)\} \right\} \qquad l = 1, .., n_{i1} \ ; \ m = 1, ..., n_{j1}$$

Modelos Jerárquicos Aglomerativos - **Estrategia de la distancia máxima o similitud mínima**[8].
Considera que las medidas entre dos clusters se tienen que tomar con los elementos más dispares.
Por tanto, la distancia y la similitud entre clusters se considera como la distancia máxima o la simulitud mínima entre las componentes de los clusters.

Si tenemos K cluster ya creados, definimos la distancia entre los clusters C_i y C_j como

$$d(Ci, Cj) = \max_{x_l \in Ci, x_m \in Cj} \{d(x_l, x_m)\} \qquad l = 1, .., n_i \ ; \ m = 1, ..., n_j$$

Para la distancia para el cluster $K+1$, uniremos los clusters Ci y C_j si

$$d(Ci, Cj) = \min_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \{d(C_{i1}, C_{j1})\} =$$

$$= \min_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \left\{ \max_{x_l \in Ci, x_m \in Cj} \{d(x_l, x_m)\} \right\} \qquad l = 1, .., n_i \ ; \ m = 1, ..., n_j$$

Si tenemos K cluster ya creados, definimos la similitud entre los clusters C_i y C_j como

$$s(Ci, Cj) = \min_{x_l \in Ci, x_m \in Cj} \{s(x_l, x_m)\} \qquad l = 1, .., n_{i1} \ ; \ m = 1, ..., n_{j1}$$

Para la similitud para el cluster $K+1$, uniremos los clusters Ci y C_j si

$$s(Ci, Cj) = \max_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \{s(C_{i1}, C_{j1})\} =$$

$$= \max_{i_1, j_1 = 1, ..., n-K, i_1 \neq j_1} \{_{x_l \in Ci, x_m \in Cj} \{s(x_l, x_m)\}\} \qquad l = 1, .., n_{i1} \ ; \ m = 1, ..., n_{j1}$$

La principal desventaja de usar este tipo de métodos es que conllevan una alta complejidad computacional.

Métodos basados en patrones

Tal y como hemos visto en [16] patrón se define por un conjunto de registros que contienen un porcentaje $x\,\%$ de valores con características semejantes. El porcentaje lo puede elegir el usuario, generalmente es un valor bastante alto.

Eliminar las muestras o las variables que tienen datos perdidos. Como su nombre indica, este método elimina aquellas muestras que tienen valores perdidos. El principal inconveniente de este método es que podría reducir demasiado el volumen de filas. La eliminación de variables consiste en eliminar aquellas variables que tienen valores perdidos. El principal inconveniente de este método es que podemos descartar variables que tengan información relevante para nuestro estudio.

Sustituir los valores perdidos por estimaciones. No debemos abusar de este método, ya que como su nombre indica, estamos estimando valores perdidos. Podemos hacer la sustitución por la media, aunque este cambio solo lo podemos realizar con variables numéricas. También podemos sustituir por la mediana. Si tenemos variables de tipo categórico, podemos hacer la sustitución por la moda.

3.2. Técnicas de predicción

En esta sección vamos a describir las distintas técnicas que hemos usado en nuestro estudio. Como ya hemos indicado anteriormente, el objetivo de este trabajado es poder realizar una predicción sobre los futuros valores de contaminación que se encontrarán en el ambiente. Para ello, hemos hecho uso de varias técnicas clásicas de predicción.

A continuación, vamos a ver de forma detallada cada método utilizado.

3.2.1. Descomposición estacional

Como hemos podido ver en [23], una serie temporal es una sucesión de observaciones sobre una variable a lo largo del tiempo. El objetivo es predecir los futuros valores de esa variable, estudiando los cambios que se producen.

Las series temporales se dividen en estacionarias y no estacionarias. Una serie es estacionaria si su media y variabilidad son constantes a lo largo del tiempo. Una serie no es estacionaria si su media y/o variabilidad no se mantienen constantes a lo largo del tiempo. Pueden mostrar una tendencia y pueden presentar efectos estacionales.

Podemos ver las series temporales como la combinación de una tendencia T, un ciclo C, una parte estacional S y una parte residual o error R. La parte estacional muestra la oscilación de los movimientos, es decir, un patrón que se repite con una periocidad que conocemos. La tendencia muestra el comportamiento de la serie en un futuro, si este es creciente o decreciente. El ciclo es un patrón que se repite con cierta regularidad pero no conocemos la periocidad. Por último, los residuos muestran las variaciones aleatorias de los componentes de la serie.

Para seguir, vamos a ver cualquier elemento cíclico como parte de la tendencia. Por tanto, vamos a hacer combinaciones de los otros tres elementos.

Vemos el modelo aditivo puro como la combinación lineal de la tendencia, la estacionalidad y el error. Para un momento t, se tiene que

$$Y(t) = T(t) + S(t) + R(t)$$

Un modelo multiplicativo puro es el producto de las tres componentes. Para un momento t, se tiene que

$$Y(t) = T(t) \times S(t) \times R(t)$$

Existen otros modelos realizando combinaciones de los elementos. Por ejemplo, para un momento t, se tiene que

$$Y(t) = (T(t) + S(t)) \times R(t)$$

Existen muchas técnicas para estimar las distintas componentes de la descomposición. Vamos a ver algunas de ellas.

Denotamos a s como el periodo de la serie, por tanto tenemos que $St = St - s$. Para conocer la componente estacional $S(t), (t = 1, ..., T)$ basta con conocer los valores consecutivos de s.
Se tiene que

$$S(t+1) + ... + S(t+s) = S1 + ... + Ss = cte$$

El método paramétrico consiste en usar modelos paramétricos para expresar la relación que existe entre la tendencia y la estacionalidad. Estos modelos se ajustan a la serie de tiempo, por ejemplo con el método de los mínimos cuadrados.

Recordemos que el método de mínimos cuadrados [15] consiste en hacer que la suma de los cuadrados de las longitudes (distancia euclídea) L_i sea

mínima. Dados los datos $(x_i, y_i) i = 1, 2, ..., N$, obtener una función Φ_x tal que:

$$\sum_{i=1}^{N} (y_i - \Phi(x_i))^2 = Min$$

Φ_x puede ser una función que se ajuste a un modelo lineal o a un modelo no lineal. Los modelos lineales son aquellos que usan funciones del tipo:

$$\Phi(x) = a_1 q_1(x) + a_2 q_2(x) + ... + a_n q_n(x)$$

Siendo $a_i, i = 1, ..., n \leq N$ = número de datos y $q_i(x), i = 1, ..., n \leq N$ funciones linealmente independientes.
Los modelos no lineales son aquellos que usan funciones $\Phi(x)$ no lineales con respecto a los parámetros de ajustes como $\Phi(x) = ae^{bx}$ ó $\Phi(x) = ax^b$

El método no paramétrico consiste en asumir que existe una suavidad en la relación existente entre la tendencia y la estacionalidad. Se aisla la tendencia y la estacionalidad a través de la suaviación del gráfico de secuencia, usando por ejemplo filtros de media móviles.

Recordamos que la media móvil [18] tiene como objetivo reducir la variabiliad de la serie, eliminando o reduciendo lo máximo posible las fluctuaciones periódicas.
Se trata de ir agrupando k valores de la serie e ir determinando para cada grupo su media.

Para obtener las medias móviles de orden k, tomamos los k primeros elementos y calculamos su media para hacerla corresponder al periodo mediano de los periodos $1, 2, ..k$. Sean los k primeros valores de la serie $y_1, y_2, ..., y_k$

$$y'_1 = \frac{\sum_{i=1}^{k} y_i}{k}$$

Sean los k valores de la serie para la segunda media móvil $y_2, y_3, ..., y_{k+1}$

$$y'_2 = \frac{\sum_{i=2}^{k+1} y_i}{k}$$

Si continuamos el proceso, hasta la última observación tendremos la n-k+1 media móviles que representarán la serie suavizada.

En nuestro caso, hemos utilizado el filtro de Hodrick-Prescott [2] para extraer la tendencia de la serie. Este método descompone la serie en dos componentes, uno tendencial y otro cíclico. El ajuste de sensibilidad de la

tendencia a las fluctuaciones a corto plazo se obtiene modificando un multiplicador λ.

Este método toma la serie Y_t para $t = 1, ...T$. La tendencia se representa por τ y la estacionalidad por c, y tenemos $Y_t = \tau_t + c_t$. Sea $\lambda > 0$.

Este filtro estima la tendencia, minimizando la desviación entre la serie original y restringiendo la volatilidad a un cierto límite superior. Concretamente, el método consiste en minimizar las desviaciones entre la serie actual y la tendencia.

$$C_t = Y_t - T_t$$

Toma en consideración que las variaciones del producto de tendencia no superen cierto porcentaje en dos períodos sucesivos. La expresión a minimizar es:

$$min \sum_{t=1}^{T} (y_t - \tau_t) + \lambda \sum_{t=2}^{T-1} [(\tau_{t+1} - \tau_t) - ((\tau_t - \tau_{t-1})]^2$$

Aunque hayamos visto la tendencia con el método de Hodrick-Prescott, usamos el algoritmo STL, cuya descripción interna y en la que nos hemos basado se encuentra en [3], para descomponer la serie temporal de nuevo. Este método se basa en el suavizado de dispersión localmente ponderado (loess). $\hat{g}_{(x)}$ es una curva de regresión loess que se usa para suavizar. Para calcularla hacemos lo siguiente:
Sean x_i, y_i con $i = 1, ..., N$ medidas de una variable dependiente.
Sea $q \in \mathbb{Z}$ el número de valores más cercano a x. Establecemos el peso para x_i de acuerdo con la distancia entre x_i y x cuando $q \leq n$. W denota una función de peso.

$$W(u) = \begin{cases} (1 - u^3)^3 & \text{si} \quad 0 \leq u < 1 \\ \\ 0 & \text{si} \quad u \geq 1 \end{cases}$$

$v_i(x)$ es la función de los pesos vecinos de x_i. Sea $\lambda_q(x)$ la distancia entre x_i y x:

$$v_i(x) = W(|x_i - x|) / \lambda_q(x)$$

Cuanto mayor sea la distancia $\lambda_q(x)$, menor será el valor del peso $v_i(x)$. Por último, buscamos un valor ajustado $\hat{g_x}$ en el punto x_i con el peso $v_i(x)$.

STL se compone de dos procedimientos, que incluyen un bucle interno y un bucle externo. El bucle interno está anidado dentro del bucle externo. Los principales pasos del bucle interno son el suavizado estacional y el suavizado de la tendencia. Los pasos para k son los siguientes:

1. Se obtiene una nueva serie restando los valores de tendencia T_k^v de los valores originales Y_v.
2. Cada una de las subseries de los ciclos obtenidas en el paso 1 se somete a una regresión de loess, y el resultado se registra como C_v^{k+1}.
3. El filtro para C_v^{k+1} incluye tres pasos. El primer paso es una media móvil de longitud n, siendo n el número de muestras. El siguiente paso es también una media móvil de longitud n. El último paso es una media móvil de longitud 3. Por último, se aplica el loess a los resultados del filtrado de paso bajo. El resultado se registra como L_v^{k+1}.
4. Obtenemos la serie estacional S_v^{k+1} como C menos L
$$S_v^{k+1} = C_v^{k+1} - L_v^{k+1} \text{ para } v = 1, ..., n$$
.
5. Obtenemos las series desestacionalizada de Y menos S.
6. La tendencia T_v^{k+1} se obtiene después de aplicar el loess a la serie desestacionalizada.

Los pasos del bucle externo son los siguientes:

1. Los valores de T_v y S_v se obtienen después del bucle interno. A continuación, el residuo R_v se calcula como:
$$R_v = Y_V - T_V - S_v$$
2. El peso de robustez ρ se define para evaluar la robustez de R_v. ρ_v es el peso de robustez en el momento v.
$$h = 6median(|R_v|)$$
$$\rho_v = B(|R_v|)/h$$
3. La fórmula de la función de peso bicuadrado B es la siguiente:
$$W(u) = \begin{cases} (1-u^2)^2 & si \quad 0 \leq u < 1 \\ \\ 0 & si \quad u \geq 1 \end{cases}$$

3.2.2. Exponential Smoothing

Se trata de uno de los métodos clásicos de predicción más importantes y usados. Vamos a ver algunas de las distintas variantes de este método basandonos en las ideas de [10].

Simple Exponential Smoothing

El alisamiento exponencial simple (SES), como su nombre indica, es el más sencillo de todos los métodos de alisamiento exponencial. Elegiremos este método cuando queramos realizar una previsión de datos donde no tengamos una tendencia clara o un patrón estacional.

Vamos a denotar al valor que predecimos por $\hat{y}$. Si usamos el método naïve, las predicciones que hagamos serán iguales al úlimo valor que tengamos en la serie. Es decir, para cierto instante $t \in T$, la predicción para los momentos $t + h$ siendo $t \in T$ y $h = 1, 2, ...$, serán el último valor disponible para la serie y_t.

$$\hat{y}_{T+h|T} = y_T \text{ para } h = 1, 2, ...$$

Con el método naïve solo importa la observación más reciente ya que todas las anteriores no van a aportarnos nada para realizar una predicción futura. Por tanto, podemos considerar a este método como una media ponderada, que tiene todo su peso a la última observación realizada.

Si usamos el método de la media, como bien indica su nombre, todos los valores futuros son iguales a la una media simple de todos los datos.

$$\hat{y}_{T+h|T} = 1/T \sum_{t=1}^{T} y_t \text{ para } h = 1, 2, ...$$

Al contrario que con el método anterior, con el método de la media le damos la misma importancia y el mismo peso a todos los valores de la serie para obtener una predicción.

Lo que tenemos con el método de alisamiento exponencial simple es una mezcla de los dos anteriores. Vamos a ver esto paso a paso. Para un momento t, tenemos que el valor es y_t y la predicción $\hat{y}_t$. Por tanto, el error de predicción es $y_t - \hat{y}_t$. El método SES predice el siguiente valor tomando el valor calculado para el momento anterior y lo ajusta usando el error de predicción. Sea c una constante tal que $0 \leq \alpha \leq 1$

$$\hat{y}_{t+1} = \hat{y} + \alpha(y_t - \hat{y}_t)$$

Podemos observar que cuando α es un valor cercano a 1, el valor que pretendemos predecir tendrá un importante ajuste por el error anterior. Sin embargo, si α es un valor cercano a 0, apenas tendrá en cuenta el anterior error de predicción. Vamos a ver de otra forma la ecuación anterior.

$$\hat{y}_{t+1} = \hat{y} + \alpha(y_t) - \alpha(\hat{y}_t)$$

$$\hat{y}_{t+1} = \alpha(y_t) + (1-\alpha)\hat{y}_t$$

Para obtener el valor en el instante $t+1$, hacemos una suma ponderada entre el valor más reciente y_t por α y la predicción más reciente $\hat{y}_t$ por $1-\alpha$. Vamos a desarrollar la última ecuación obtenida, reemplazando $\hat{y}_t$ por sus componentes.

$$\hat{y}_{t+1} = \alpha(y_t) + (1-\alpha)[\alpha(y_{t-1}) + (1-\alpha)\hat{y}_{t-1}]$$

$$\hat{y}_{t+1} = \alpha(y_t) + \alpha(1-\alpha)y_{t-1} + (1-\alpha)^2\hat{y}_{t-1}$$

Si continuamos haciendo estas sustituciones, tenemos que

$$\hat{y}_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \alpha(1-\alpha)^3 y_{t-3} + ... + \alpha(1-\alpha)^(t-1)y_1 + (1-\alpha)^t y_1$$

Como deciamos al principio, vemos que es una combinación de los métodos anteriores vistos. Calculamos un valor futuro como una media ponderada, dandole más importancia a los valores más recientes que a los antiguos. Las ponderaciones van disminuyendo de una forma exponencial, de esta forma conseguimos que las observaciones más antiguas tengan un peso menor. Es sencillo ver que cuanto más cerca esté α de 0, más peso tendrán los valores pasados. Por el contrario, cuanto más cerca esté α de 1, se le dará un mayor peso a los valores más recientes.

La elección del primer valor es muy importante y se conoce como la inicialización del problema.

Para predicciones de mayor alcance, se toma la función de predicción como una función plana. Esto es debido a que el método SES trabaja mejor con datos que no tienen tendencias ni estacionalidades. Sea $h \in \mathbb{Z}$

$\hat{y}_{t+h|t} = \hat{y}_{t+1}$

Otra forma de expresar el método para ver de forma generalizada el alisamiento exponencial para permitir la tendencia y la estacionalidad.

$$l_t = \hat{y}_{t+1}$$

$$\hat{y}_{t+h|t} = l_t \quad \wedge \quad l_t = \alpha y_t + (1-\alpha)l_{t-1}$$

Holt's Linear Trend Method

En 1957 Holt extendio el método de alisamiento exponencial simple para poder realizar la predicción cuando tenemos datos con una tendencia. Esta compuesto de tres ecuaciones, una de previsión y dos de alisamiento.

$$\text{Ecuación de previsión} \qquad \hat{y}_{t+h|t} = l_t + hb_t$$

$$\text{Ecuación de valores} \qquad l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \quad \text{con } 0 \leq \alpha \leq 1$$

$$\text{Ecuación de tendencia} \qquad b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \quad \text{con } 0 \leq \alpha \leq 1 \quad \text{y } 0 \leq \beta^* \leq 1$$

Siendo l_t una estimación del valor de la serie para un momento t. Una estimación de la tendencia para el momento t es b_t. α el parámetro de alisamiento de los valores y β^* el parámetro de alisamiento de la tendencia, ambos con valores comprendidos entre 0 y 1.

La función de predicción está formada por la función de valores más la función de tendencia. Concretamente, para el instante $t+h$ tenemos que el valor predicho va a ser el valor de la ecuación de valores en el instante t más h veces el valor de la tendencia en el instante t.

El valor de la ecuación de valores en el momento t, l_t, será una media ponderada de y_t y el valor de la predicción para el instante anterior. Por otro lado, la ecuación de tendencia nos dice que para el instante t, b_t es la media ponderada de la estimación de la tendencia y el valor de la tendencia para $t-1$. Es sencillo ver que este método tiene una tendencia constante, creciente o decreciente.

Un caso particular es cuando tomamos $\beta^* = 0$, denominado como "SES con derivada", el cuál está muy relacionado con el método de previsión visto en [1].

En 1985, Gardner y McKenzie añadieron un nuevo parámetro que hace que en un futuro esta tendencia se aplane, y denominaron al método como método de la tendencia amortiguada. Este tipo de métodos han tenido mucho éxito. Vamos a denominar al parámetro de amortiguación como ϕ. Tendríamos las siguientes ecuaciones:

$$\text{Ecuación de predicción} \qquad \hat{y}_{t+h|t} = l_t + (\phi + \phi^2 + \ldots + \phi^h)b_t \quad \text{con } 0 \leq \phi \leq 1$$

$$\text{Ecuación de valores} \qquad l_t = \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1})$$

$$\text{con } 0 \leq \alpha \leq 1 \text{y } 0 \leq \phi \leq 1$$

$$\text{Ecuación de tendencia} \qquad b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1}$$

$$\text{con } 0 \leq \alpha \leq 1,\ 0 \leq \beta^* \leq 1 \text{y } 0 \leq \phi \leq 1$$

Si tomamos ϕ como 1, obtendremos las mismas predicciones que con el método lineal de Holt.

Si $0 \leq \phi \leq 1$, a medida que crece h, los valores predichos tenderán a una asíntota dada por $l_t + (\phi b_t)/(1-\phi)$

Holt´s-Winter´s Seasonal Method

Los métodos vistos anteriormente no recogían la estacionalidad y por tanto no podían manejar por sí solos problemas con datos estacionales. Entre 1957 y 1960, el método de Hölt visto anteriormente fue extendido primero por el propio Hölt y más tarde por Winters.

Esta nueva versión se compone de una función de predicción y tres más de alisamiento. Las ecuaciones para el alisamiento se componen de una para el nivel l_t, otra para la tendencia b_t y una última para la estacionalidad s_t. Para esta extensión se usa una variable m, la cual indica la frecuencia de la estacionalidad. Existen dos variaciones distintas para este método, un modelo aditivo y otro multiplicativo. Ambas difieren en la variación de la estacionalidad, ya que con el modelo aditivo tomamos estas variaciones como constantes y con el multiplicativo toma el cambio proporcional al nivel de la serie. Vamos a ver cada uno de estos métodos en profundidad.

Holt-Winters' Seasonal Method Additive

Las ecuaciones del método aditivo son las siguientes:

$$\begin{aligned}
\text{Ecuación de predicción} \quad & \hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)} \\
\text{Ecuación de nivel} \quad & l_t = \alpha(y_t + s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\
\text{Ecuación de tendencia} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\
\text{Ecuación de estacionalidad} \quad & s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned}$$

$$\text{Siendo } 0 \leq \alpha \leq 1, \quad 0 \leq \beta^* \leq 1 \text{ y} \quad k = (h-1)/m \text{ con } k \in \mathbb{Z}$$

También podemos definir la función para la estacionalidad como la siguiente función:

$$s_t = \gamma^*(y_t - l_t + (1-\gamma^*)s_{t-m}$$

$$\text{Siendo } \gamma = \gamma^*(1-\alpha) \quad 0 \leq \gamma^* \leq 1$$

Esta forma de definir k es para asegurar que la estacionalidad que usamos es del último año de la muestra. La ecuación de valores se define para un instante t como una media ponderada entre los datos ajustados a la estacionalidad $y_t - s_{t-m}$ y la predicción sin la estacionalidad $l_{t-1} + b_{t-1}$. Como podemos observar, la ecuación para la tendencia es la misma que usamos

en el método de Holt Lineal. Por último, la ecuación para la estacionalidad se realiza también como una media ponderada entre la estacionalidad más reciente, $y_t - l_{t-1} - b_{t-1}$ y la estacionalidad en el mismo periodo hace m periodo de tiempo.

Holt-Winters' Seasonal Method Multiplicative

Las ecuaciones del método multiplicativo son las siguientes:

$$\text{Ecuación de predicción} \qquad \hat{y}_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)}$$

$$\text{Ecuación de nivel} \qquad l_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$\text{Ecuación de tendencia} \qquad b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$

$$\text{Ecuación de estacionalidad} \qquad s_t = \gamma y_t/(l_{t-1} + b_{t-1}) + (1-\gamma)s_{t-m}$$

$$\text{Siendo } 0 \leq \alpha \leq 1, \quad 0 \leq \beta^* \leq 1 \text{ y} \quad k = (h-1)/m \text{ con } k \in \mathbb{Z}$$

La longitud de la estacionalidad se denomina m.
Tanto para el modelo aditivo como para el multiplicativo, la ecuación de tendencia es la misma.

3.2.3. ARIMA

En esta sección nos vamos a apoyar en [11]Otro de los métodos clásicos de predicción, muy conocido y utilizado, es ARIMA (."AutoRegressive Integrated Moving Average"). El objetivo de estos modelos es describir las correlaciones existentes entre los datos. Existen dos tipos de modelos distintos dependiendo de si son estacionales o no. Vamos a definir algunos conceptos antes de explicar cada modelo en detalle.

Conceptos previos

Vamos a explicar y definir los conceptos de estacionalidad y diferenciabilidad, además de algunos modelos que usaremos en ARIMA.

Definimos ruido blanco somo las series temporales que no muestran autocorrelación. Una de las propiedades de las series temporales es que pueden ser estacionarias. Esta propiedad no depende del momento en el que veamos la serie, por tanto, si una serie tiene tendencia o tiene estacionalidad, no será estacionaria. Una serie temporal que es estacionaria, en general no tiene patrones que podamos predecir a largo plazo. Las series con ruido blanco son estacionarias. Las series temporales con comportamientos cíclicos, pero sin tendencia ni estacionalidad, son también estacionarias.

La diferencial nos muestra la diferencia entre dos valores consecutivos. De esta forma, es posible hacer que una serie temporal no estacionaria sea estacionaria. Por otro lado, la diferenciabilidad puede ayudar a estabilizar la media de una serie temporal ya que podemos tener cambios de nivel, y por consiguiente reducir, incluso eliminar, la tendencia y estacionalidad.

Modelo de recorrido aleatorio

Este tipo de modelo es muy utilizado con datos no estacionarios, en especial con datos financieros y económicos. Los recorridos aleatorios normalmente tienen largos periodos con aparente crecimiento o decrecimiento y repentinos cambios en la dirección.

Como los futuros valores de una serie son imprevisibles, la predicción de estos modelos es igual al último valor que tienen.

Primero, describimos como serie diferenciada al cambio que existe entre valores consecutivos en una serie, es decir:

$$y'_t = y_t - y_{t-1}$$

Cuando estamos ante una serie con ruido blanco la función cambia. Sea ϵ_t el ruido, tenemos:

$$y_t - y_{t-1} = \epsilon_t$$

$$y_t = y_{t-1} + \epsilon_t$$

Un modelo muy relacionado, permite que estas diferencias tengan una media distinta a cero. Luego, tenemos que:

$$y_t - y_{t-1} = c + \epsilon_t$$

$$y_t = c + y_{t-1} + \epsilon_t$$

Siendo c la media entre los cambios de las valores consecutivos. Por tanto, tenemos que si $c > 0$ entonces y_t tenderá a crecer. Si $c < 0$ entonces y_t tenderá a decrecer.

Algunas veces, aunque hayamos hecho una primera derivada, podemos observar que los datos no son estacionarios, por lo que será necesario que volvamos a derivar.

$$y''_t = y'_t - y'_{t-1}$$

$$y''_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$$

$$y''_t = y_t - 2y_{t-1} - y_{t-2}$$

Generalmente, no necesitaremos ir más a allá de la segunda derivada.

Una diferencia estacional es una diferencia entre un valor con respecto al valor del periodo anterior. La predicción en este modelo es igual al último valor del periodo correspondiente.
Sea m el número de periodos, tenemos que:

$$y'_t = y_t - y_{t-m}$$

Cuando estamos ante una serie con ruido blanco la función cambia. Sea ϵ_t el ruido, tenemos:

$$y_t - y_{t-m} = \epsilon_t$$
$$y_t = y_{t-m} + \epsilon_t$$

Una segunda derivada para las series estacionales sería:

$$y''_t = y'_t - y'_{t-1}$$
$$y''_t = (y_t - y_{t-m}) - (y_{t-1} - y_{t-m-1})$$
$$y''_t = y_t - y_{t-1} - y_{t-m} + y_{t-m-1}$$

Algunas veces, es necesario tomar tanto diferencias ordinarias como estacionales para poder conseguir que los datos que tenemos sean estacionarios. Aunque no importa el orden en el que usar ambas diferencias, se recomienda que si los datos tienen un patrón estacional fuerte, se utilice primero la derivada estacional.

Para poder saber de una forma más segura si es necesario usar la diferenciación para obtener una serie estacionaria se usa la prueba de raíz unitaria. Existen numerosas pruebas de raíz unitaria, que se basan en distintos supuestos, por lo que se puede llegar a respuestas contradictorias según la que utilicemos.

Modelos autoregresivos

Este tipo de modelo es muy flexible manejando un amplio rango de patrones de series temporales. Un modelo autoregresivo predice el valor de una variables haciendo una combinación lineal de los valores anteriores de esa variable. Sea p el valor del orden del modelo, ϵ_t el ruido blanco y c la media,

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t$$

Definimos un modelo $AR(p)$ como un modelo autoregresivo de orden p.

Modelos de media variable

Estos modelos se usan para predecir valores futuros, haciendo una media ponderada de los últimos errores de previsión. Sea p el valor del orden del modelo, ϵ_t el ruido blanco,

$$y_t = c + \epsilon_t + \epsilon_{t-1}\theta_1 + \epsilon_{t-2}\theta_2 + ... + \epsilon_{t-p}\theta_p$$

Definimos un modelo $MA(p)$ como un modelo de media variable de orden p.

Modelo ARIMA no estacional

Un modelo ARIMA no estacional se obtiene combinando diferenciabilidad junto al modelo de autoregresión y al modelo de media variable.

$$y'_t = c + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \epsilon_t + \epsilon_{t-1}\theta_1 + ... + \epsilon_{t-q}\theta_q$$

Definimos este modelo como $ARIMA(p, d, q)$ donde p es el orden de la parte autoregresiva, d es el grado de diferenciabilidad y q es el orden de la parte del modelo de media variable.

Es complicado elegir los valores más apropiados para las variables del modelo. Dependiendo de los valores que tomemos de p, d, q, podemos obtener modelos ya conocidos:

- Si $p = d = q = 0$ entonces tenemos ruido blanco.
- Si $p = q = 0$ y $d = 1$ sin constantes entonces tenemos un modelo de recorrido aleatorio.
- Si $p = q = 0$ y $d = 1$ con una constante entonces tenemos un modelo de autoregresión.
- Si $p = d = 0$ y $q \neq 0$ entonces tenemos un modelo de media variable.

En algunos lenguajes de programación, existen funciones que nos dan los valores óptimos para p, d y q.

La constante c tiene también un impacto importante en las predicccciones a largo plazo. El grado que elegimos para la diferenciabiliad d tiene efecto en los intervalos de predicción, ya que cuanto más alto sea el valor más rápido aumentan los intervalos de predicción. Además el valor que se le asigna a p es importante si tenemos ciclos dentro de nuestros datos. Si queremos obtener predicciones cíclicas necesitaremos que $p \geq 2$

- Si $c = 0$ y $d = 0$ entonces a largo plazo las predicciones tenderán a 0.
- Si $c = 0$ y $d = 1$ entonces a largo plazo las predicciones tenderán a una constante distinta de 0.

- Si $c = 0$ y $d = 2$ entonces a largo plazo las predicciones seguirán una línea recta.
- Si $c \neq 0$ y $d = 0$ entonces a largo plazo las predicciones tenderán a la media.
- Si $c \neq 0$ e y $d = 1$ entonces a largo plazo las predicciones seguirán una línea recta.
- si $c \neq 0$ y $d = 2$ entonces a largo plazo las predicciones seguirán una tendencia cuadrática.

Modelo ARIMA estacional

Este tipo de modelo es como el anterior pero añadiendo términos estacionales. Sea m el número de observaciones por año, el modelo se ve como,

$$ARIMA(p, d, q)(P, D, Q)_m$$

Siendo p, d, q los valores de la parte no estacional y P, D, Q los valores para la parte estacional del modelo.

3.2.4. LSTM

Desde hace varios años, se ha dado un nuevo enfoque al análisis de datos intentando replicar la capacidad de aprendizaje de los seres vivos, concretamente imitando el sistema neuronal.
Vamos a basarnos en [7] para explicar está sección. Este sistema consiste en una red compleja de células, denominadas neuronas, que son capaces de trabajar en paralelo y reorganizarse en la fase de aprendizaje.

Cada neurona está formada por dendritas, un cuerpo celular y un axón. El proceso de aprendizaje lo conforman los siguientes pasos:

1. Las dendritas reciben las señales de entrada y ponderan la información recibida según su importancia.
2. El cuerpo celular se encarga de sumar la información ponderada.
3. Si se llega a un umbral, el axón manda un mensaje de salida a su neurona vecina.

Este comportamiento es el que se intenta simular en una red neuronal. Tenemos como entrada una información que se pondera antes de mandarla a un cuerpo celular, que es representado por una función de activación. La salida que se genera es proporcional al valor devuelto por esa función y se

manda como información de entrada a otras neuronas.
Las neuronas están ordenadas de acuerdo a la finalidad que tengan.

LSTM es un acrónimo de "Long-Short Term Memories", es decir, memorias a corto plazo, y es un tipo de red neuronal. En general, este tipo de modelos son complejos y se necesita una gran cantidad de datos para poder predecir.

Una capa de memoria LSTM es una variante de un capa simple recurrente, en la cual el vector de información $(c_t)_{t\in\mathbb{Z}}$ siendo $c_t \in \mathbb{R}^q$ lleva la información a través de varios pasos.

Conceptos previos

Neuronas y función de activación

En un modelo de red neuronal la información se almacena y transmite en n vectores de dimensión p, $x_i = (x_{i1}, ..., x_{ip})^\top$, siendo $x_{iq} \in \mathbb{R}$ con $q = 1, ..., p$ e $i = 1, ..., n$.

La información llega a las dendritas, las cuales asignan un peso $w_j \in \mathbb{R}$ a cada x_{ij} y suman estos valores. Normalmente también se añade un desplazamiento w_0 a esta suma.
Denotamos a $w = (w_0, ..., w_p^\top)$ como el vector de peso o vector de ponderación.
Luego las dendritas realizan la siguiente operación:

$$w_0 + \sum_{j=1}^{p} w_j x_{ij}$$

Una vez en el cuerpo celular, la suma ponderada es usada una función de activación ϕ.
Estas funciones son continuas y estrictamente crecientes, además suelen estar acotadas superior e inferiormente.

La elección de la función de activación es muy importante ya que recoge el comportamiento entre las entradas y las salidas. Para poder capturar relaciones más complejas, se suelen usar funciones no lineales. Algunas de las funciones más comunes son:

- Función sigmoide o logística.

$$\phi(z) = \frac{1}{1 + e^{-x}}$$

$$\phi(z) : \mathbb{R} \to [0, 1]$$

- Función tangente hiperbólica.

$$\phi(z) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
$$\phi(z) : \mathbb{R} \to [-1, 1]$$

- Función de distribución de probabilidad, por ejemplo función de distribución Gaussiana

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{1}{2}u^2} \partial u$$
$$\phi(z) : \mathbb{R} \to [0, 1]$$

El axon es el encargado de enviar la salida generada por $\hat{y}_i$.

$$\hat{y}_i = w_0 + \sum_{j=1}^{p} w_j x_{ij}) = \phi(w^\top \begin{pmatrix} 1 \\ x_i \end{pmatrix})$$

Una red neuronal es un conjunto interconectado de neuronas. Denominamos neuronas ocultas a aquellas que se encuentran entre las neuronas de información y las de salida.
Se denominan redes recurrentes a aquellas que tienen conexiones circulares, redes superficiales a aquellas que tienen una única capa oculta y profundas a aquellas que tienen más de una capa.

Procesos estocásticos

Las variables estocásticas son variables que toman un determinado valor con cierta probabilidad, por lo que se define como un conjunto de estados (posibles valores que puede tener) y una distribución de probabilidad sobre ese conjunto.
El conjunto de estados puede ser discreto, como cara o cruz en una moneda, o continuo en un intervalo, como la energía de una partícula. La denotaremos de forma vectoral X si tiene un carácter multidimensional, como las componentes de velocidad de una partícula Browniana.

Modelo LSTM

Denotamos como LSTM(p,q) a un modelo neuronal autoregresivo LSTM con un orden p y con q neuronas escondidas. La función que corresponde a este modelo es

$$x_{t+1} = f_\omega(x_t, y_{t-1}, c_t) + w_t$$

Vamos a ver detalladamente la función.

w_t es un proceso estocástico con media nula, es decir $E(w_t) = 0$ y una desviación estándar σ_w^2.

Sea el producto de Hadamard $\otimes$, el flujo de información $(c_t)_{t\in\mathbb{Z}} \in \mathbb{R}^q$ se calcula como la suma de la tasa de olvido por el flujo de información anterior y la tasa de entrada por la información marginal.

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \bar{c}_t$$

Definimos las funciones usadas arriba como:

Puertas de olvido (forget rate) $$f_t = \phi_{sig}(\alpha_{f,k}^\top y_{t-1} + \beta_{f,k}^\top \begin{pmatrix} 1 \\ x_t \end{pmatrix})_{k=1,\dots,q}$$

Puertas de entrada (input rate) $$i_t = \phi_{sig}(\alpha_{i,k}^\top y_{t-1} + \beta_{i,k}^\top \begin{pmatrix} 1 \\ x_t \end{pmatrix})_{k=1,\dots,q}$$

Puertas de información $$\bar{c}_t = \tanh(\alpha_{c,k}^\top y_{t-1} + \beta_{c,k}^\top \begin{pmatrix} 1 \\ x_t \end{pmatrix})_{k=1,\dots,q}$$

Siendo la tasa de olvido $(f_t)_{t\in\mathbb{Z}} \in \mathbb{R}^q$, la tasa de entrada $(i_t)_{t\in\mathbb{Z}} \in \mathbb{R}^q$ y la información marginada en el momento t, $(\bar{c}_t)_{t\in\mathbb{Z}}$.
La funcion ϕ_{sig} es una función de activación sigmoidal. Además los valores $\alpha_{f,k}, \alpha_{i,k}, \alpha_{c,k} \in \mathbb{R}^q$ y $\beta_{f,k}, \beta_{i,k}, \beta_{c,k} \in \mathbb{R}^{p+1}$.

Se denota por $(y_t)_{t\in\mathbb{Z}} \in \mathbb{R}^q$, a la salida que se produce en la capa intermedia de las neuronas, y se calcula como el producto de la puerta de salida $(o_t)_{t\in\mathbb{Z}} \in \mathbb{R}^q$ y la tangente hiperbólica de la información c_t.

$$y_t = o_t \otimes \tanh(c_t)$$

$$o_t = \phi_{sig}(\alpha_{o,k}^\top y_{t-1} + \beta_{o,k}^\top \begin{pmatrix} 1 \\ x_t \end{pmatrix})_{k=1,\dots,q}$$

Siendo $\alpha_{o,k}, \beta_{o,k} \in \mathbb{R}^{p+1}$ para $k = 1, ..., q$.

Por último, sea $\delta \in \mathbb{R}^{q+1}$, la función f_ω se define como:

$$f_\omega(x_t, y_{t-1}, c_t) = \delta^\top \begin{pmatrix} 1 \\ y_t \end{pmatrix}$$

Tenemos como entrada la salida producida en la neurona anterior denotada por y_{t-1} y los valores actuales de nuestra serie temporal x_t.

La salida que se genera es ponederada por la tangente hiperbólica del flujo de información c_t.

3.3. Métricas de error

En esta sección vamos a explicar que son las métricas de error y como las utilizamos para ver que método es mejor para realizar una predicción. Podemos usar distintas medidas de error para comparar un valor real con uno pronosticado.

Lo que explicamos está basado en las ideas de [17]. Se define el error como la diferencia que existe entre el valor real y el valor que hemos predicho. Sea y_t el valor real para el momento $t \in T$ y sea $\hat{y}_t$ el valor pronosticado para el momento $t \in T$.

$$e_T = y_T - \hat{y}_T$$

Lo primero que vamos a hacer es ver la diferencia que existe entre las métricas dependientes de la escala y las métricas agnósticas de escala.

Las medidas de error dependientes de la escala son aquellas donde la escala en la que se encuentran los datos es la misma que se usa en las mediciones de los errores.
Con este tipo de medidas no podemos hacer comparaciones entre series que tienen distintas unidades de medida o si tienen la misma, que esta difiera demasiado entre los datos.

Las medidas de error agnósticas de escala se suelen expresar como porcentajes, ya que de esta forma no están ligando a las unidades. Se suelen usar para comporar los rendimientos de pronóstico.
Sin embargo, si el valor real es 0 para algún momento t las medidas de error serán indefinidad o infinitas.
Siendo e_t el error de pronóstico para un instante t e y_t el valor real.

$$p_t = 100\frac{e_t}{y_t}$$

Una vez vista la diferencia entre ambas medidas, vamos a ver las métricas utilizadas.

- Error medio
  El ME se calcula como la media de los errores que se han producido.

$$ME = med(e_t)$$

- Error absoluto medio
  El MAE se calcula como la media de los errores absolutos que se han producido.

$$MAE = med(|e_t|)$$

- Error cuadrático medio
  EL RMSE se calcula como la raíz cuadrada de la media de los errores al cuadrado que se han producido.

$$RMSE = \sqrt{med(e_t^2)}$$

- Error porcentual medio
  EL MPE se calcula como la media de los errores porcentuales que se han producido.

$$MPE = med(p_t)$$

- Error porcentual absoluto medio
  EL MAPE se calcula como la media de los errores porcentuales absolutos que se han producido.

$$MAPE = med(|p_t|)$$

Capítulo 4

Experimentos: preprocesamiento y modelado

En este cuarto capítulo vamos a explicar detalladamente el preprocesamiento y el análisis del conjunto de los datos. Los datos utilizados están explicados en la sección 2.3. El proceso que vamos a seguir lo podemos ver en 4.1

Como ya hemos comentado en el capítulo 3 de metodología, lo primero que realizamos es encontrar y solucionar las anomalías sobre el conjunto de datos, para así poder trabajar con ellos.

Una vez hecha la limpieza de datos, consideramos tres dataframes distintos. El primero de ellos contendrá los datos para ambas zonas, el segundo contendrá los valores para la zona de Roadside y el tercero los valores para la zona de Background.
De esta forma, podemos comparar los resultados que se tienen entre ambas zonas y hacer un estudio más detallado dependiendo del sitio en el que nos encontremos.

Una vez preparados los datos y dataframes para trabajar con ellos, vemos las estadísticas básicas de cada uno. Estas incluyen la media, la mediana, el máximo, el mínimo, el número de datos y la desviación estandar.

Tras esto, intentamos buscar tendencias en nuestros datos para ver si hay un comportamiento iterativo. Hacemos varias visualizaciones gráficas de las distintas métricas, comparando ambas zonas.

El siguiente objetivo es estudiar las posibles relaciones existentes entre las variables que tenemos. Mostramos la matriz de correlación entre las distintas métricas de cada dataframe, sus tendencias y distribuciones de probabilidad entre otras.

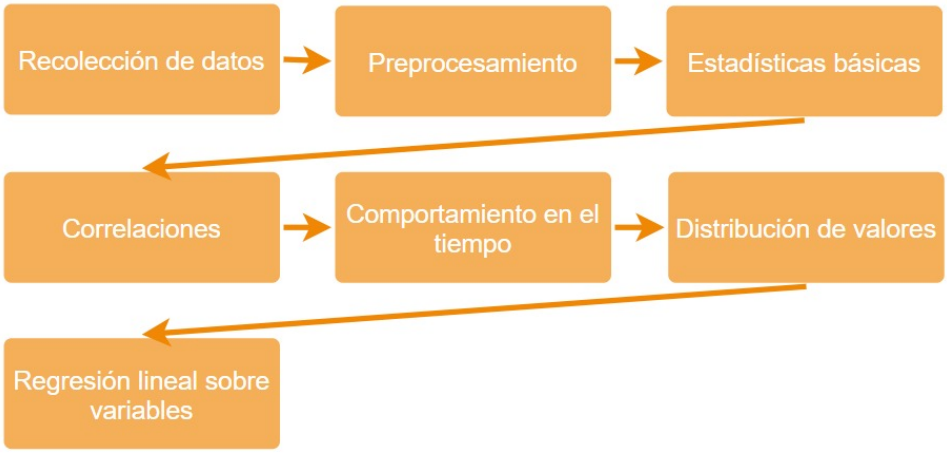

Figura 4.1: Proceso a seguir en el capítulo 4. Experimentos-Procesamiento y Modelo

4.1. Preprocesamiento

Empezamos nuestro trabajo con la carga de datos desde el fichero excel. Como podemos ver se trata de un archivo que contiene los valores de varias partículas para las zonas de Roadside y Background de Londres a un nivel mensual. Ambas zonas se encuentran en la parte superior del excel y cada métrica se repite para cada una de ellas.

Haciendo un análisis general sobre los valores y ya que nuestro conjunto de datos tenía 141 x 15 celdas, no ha hecho falta ningún algoritmo para encontrar defectos en él.
Hemos podido hacer la detección de anomalías manualmente y hemos encontrado lo siguiente:

- Valores anómalos. Existen valores no numéricos para variables numéricas.
- Valores nulos. Existen valores nulos para variables numéricas
- Distinto formato. Hay varios formatos distintos para expresar el mes.

Una vez encontradas las irregularidades, procedemos a subsanarlas. Para ello, estuvimos pensando en varios procesos:

1. Eliminar las muestras o las variables que tienen datos perdidos.

   a) Eliminación de muestras. Como su nombre indica, este método elimina aquellas muestras que tienen valores perdidos. El principal inconveniente de este método es que podría reducir demasiado el volumen de filas.

   b) Eliminación de variables. Como su nombre indica, este método consiste en eliminar aquellas variables que tienen valores perdidos. El principal inconveniente de este método es que podemos descartar variables que tengan información relevante para nuestro estudio.

2. Sustituir los valores perdidos por estimaciones. No debemos abusar de este método, porque como su nombre indica, estamos estimando valores perdidos. Podemos hacer la sustitución por la media, aunque este cambio solo lo podemos realizar con variables numéricas. También podemos sustituir por la mediana. Si tenemos variables de tipo categórico, podemos hacer la sustitución por la moda.

Finalmente, para nuestro estudio, hemos decidido hacer sustitución de los datos por la media, ya que consideramos que es la mejor forma para no descartar ninguna variable, ni descartar fechas en las que tenemos medidas de otras variables. Adicionalmente, tiene sentido aprovecharse de la continuidad que se le supone a las medidas.

```
1
2 filePath = "/content/drive/MyDrive/TFG/Datos/air-quality-london.xlsx"
3 xls = pd.ExcelFile(filePath)
4
5 pr = pd.read_excel(xls, 'Monthly averages', header=[0,1],
6             index_col=[0],skiprows=0)
7
8 pr
```

	London Mean Roadside							London Mean Background						
Month	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)
Jan-2008	NaN	55.502688	NaN	29.512097	24.969086	14.678763	4.217742	NaN	42.338710	NaN	36.942204	18.817204	.	3.572581
Feb-2008	NaN	75.922414	NaN	20.317529	39.477011	28.772989	7.553161	NaN	60.237069	NaN	26.425287	31.896552	.	6.734195
Mar-2008	NaN	55.610215	NaN	40.103495	21.569892	12.300135	3.868280	NaN	39.801075	NaN	50.227151	15.477151	.	2.286290
Apr-2008	NaN	61.756944	NaN	37.884722	28.740278	20.461111	4.475000	NaN	44.009722	NaN	50.133333	21.729167	.	3.236111
May-2008	NaN	62.903226	NaN	46.266129	34.611559	27.508065	4.634409	NaN	44.141129	NaN	60.512097	29.545699	16.5768	4.250000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2019-03-01 00:00:00	30.952285	42.575403	90.038978	34.875134	20.914651	11.297043	4.895270	7.719355	25.699328	36.038306	52.689382	16.628360	10.8745	2.271774
2019-04-01 00:00:00	25.231944	46.223750	84.910278	38.126528	32.253750	22.804444	12.417270	6.927083	31.455278	39.896806	54.743472	29.204861	23.1193	2.762639
2019-05-01 00:00:00	24.680914	39.665726	77.507527	35.689516	19.479973	10.655914	11.511081	5.481048	22.446371	29.282258	53.678360	15.331989	10.8839	1.535484

Figura 4.2: Datos cargados desde excel

Para poder continuar y trabajar con este conjunto, vamos a volver a cargar los datos, pero esta vez modificando la disposición y el formato.

```
filePath = "/content/drive/MyDrive/TFG/Datos/air-quality-london.xlsx"
xls= pd.ExcelFile(filePath)

pr = pd.read_excel(xls, 'Monthly averages', header=[0,1], index_col
    =[0], skiprows=0, na_values='.')
pr = pr.unstack().rename_axis(('Zone','Metric','Month')).reset_index(
    name='Value')
pr = pr[['Month','Zone','Metric','Value']]

pr = pr.pivot_table(index=["Month", "Zone"],
                columns='Metric',
                values='Value')

pr = pr.reset_index(level=[0,1])

pr['Month'] =  pd.to_datetime(pr['Month'])
pr.set_index('Month', inplace=True)
pr.sort_index(inplace=True)

pr_LMR = pr.loc[pr["Zone"]=="London Mean Roadside"]
pr_LMB = pr.loc[pr["Zone"]=="London Mean Background"]

del(pr_LMR['Zone'])
del(pr_LMB['Zone'])

metrics=['Nitric Oxide (ug/m3)', 'Nitrogen Dioxide (ug/m3)', 'Oxides
    of Nitrogen (ug/m3)', 'Ozone (ug/m3)', 'PM10 Particulate (ug/m3)',
     'PM2.5 Particulate (ug/m3)', 'Sulphur Dioxide (ug/m3)']
names= ['Nitric Oxide (ug/m3)', 'Nitrogen Dioxide (ug/m3)', 'Oxides of
     Nitrogen (ug/m3)', 'Ozone (ug/m3)', 'PM10 Particulate (ug/m3)', '
    PM2.5 Particulate (ug/m3)', 'Sulphur Dioxide (ug/m3)']

for metric in metrics:
    mean= pr[metric].mean()
    pr[metric].fillna(mean, inplace=True)

for metric in metrics:
```

```
32      mean= pr_LMR[metric].mean()
33      pr_LMR[metric].fillna(mean, inplace=True)
34
35  for metric in metrics:
36      mean= pr[metric].mean()
37      pr_LMB[metric].fillna(mean, inplace=True)
```

En la línea 4 cargamos el archivo excel al igual que hemos hecho antes pero añadiendo `na_values = '.'`, para que el valor '.' sea considerado como NaN.

En la línea 5 cambiamos la estructura de los datos, ya que al tener un multiindice es un poco más complicado trabajar con los valores. Dejamos cuatro columnas que contienen la zona, el nombre de la métrica, el mes y los valores de las métricas. A continuación en la línea 6 ordenamos el orden en el que queremos que se muestre, quedandose como se muestra en la figura 4.3.

	Month	Zone	Metric	Value
0	Jan-2008	London Mean Roadside	Nitric Oxide (ug/m3)	NaN
1	Feb-2008	London Mean Roadside	Nitric Oxide (ug/m3)	NaN
2	Mar-2008	London Mean Roadside	Nitric Oxide (ug/m3)	NaN
3	Apr-2008	London Mean Roadside	Nitric Oxide (ug/m3)	NaN
4	May-2008	London Mean Roadside	Nitric Oxide (ug/m3)	NaN
...	...	...	...	...
1941	2019-03-01 00:00:00	London Mean Background	Sulphur Dioxide (ug/m3)	2.271774
1942	2019-04-01 00:00:00	London Mean Background	Sulphur Dioxide (ug/m3)	2.762639
1943	2019-05-01 00:00:00	London Mean Background	Sulphur Dioxide (ug/m3)	1.535484
1944	2019-06-01 00:00:00	London Mean Background	Sulphur Dioxide (ug/m3)	1.824861
1945	2019-07-01 00:00:00	London Mean Background	Sulphur Dioxide (ug/m3)	2.576075

Figura 4.3: Cambio de estructura de los datos

Desde las líneas 8 a la 12, volvemos a cambiar la estructura para que se parezca un poco más a como la queremos finalmente. Ponemos como columnas los meses, las zonas y las distintas métricas.

En la línea 14, cambiamos el formato para todos los valores del mes estableciendolo a un tipo fecha. En las líneas 15 y 16 establezco el mes como índice y ordenamos el conjunto de datos por este índice, ya que no teníamos una ordenación por fecha.

Decidimos tener tres conjuntos distintos, ya que de esta forma, podemos estudiar las zonas por separado y tener otro conjunto de datos en el que ver el comportamiento de las partículas independientemente de donde se hayan tomando los valores. Esto lo realizamos desde la línea 18 a la 22.

Por tanto vamos a tener un conjunto denominado como pr que contendrá todo, otro conjunto denominado pr_LMR que solo tendrá los valores de London Mean Roadside y un tercero conjunto pr_LMB con los valores para London Mean Background.

Finalmente, desde las líneas 27 a la 37, decidimos usar la media de cada variable para sustituir a los valores NaN que tenemos. Para poder hacer un buen estudio, añadimos la media de cada conjunto por separado.

Con todos estos cambios, el conjunto total de datos se nos queda tal y como se muestra en la figura 4.7. Los conjuntos pr_LMR y pr_LMB tendrán el formato que podemos ver en las imágenes 4.8 y 4.9 respectivamente.

Metric Month	Zone	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)
2008-01-01	London Mean Roadside	48.509456	55.502688	96.214623	29.512097	24.969086	14.678763	4.217742
2008-01-01	London Mean Background	48.509456	42.338710	96.214623	36.942204	18.817204	14.465143	3.572581
2008-02-01	London Mean Background	48.509456	60.237069	96.214623	26.425287	31.896552	14.465143	6.734195
2008-02-01	London Mean Roadside	48.509456	75.922414	96.214623	20.317529	39.477011	28.772989	7.553161
2008-03-01	London Mean Background	48.509456	39.801075	96.214623	50.227151	15.477151	14.465143	2.286290
...	...	...	...	...	...	...	...	...
2019-05-01	London Mean Roadside	24.680914	39.665726	77.507527	35.689516	19.479973	10.655914	11.511081
2019-06-01	London Mean Background	4.969306	19.414583	24.890417	51.814722	14.112083	9.464583	1.824861

Figura 4.4: Conjunto de datos modificado

Metric Month	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)
2008-01-01	75.618072	55.502688	136.867098	29.512097	24.969086	14.678763	4.217742
2008-02-01	75.618072	75.922414	136.867098	20.317529	39.477011	28.772989	7.553161
2008-03-01	75.618072	55.610215	136.867098	40.103495	21.569892	12.300135	3.868280
2008-04-01	75.618072	61.756944	136.867098	37.884722	28.740278	20.461111	4.475000
2008-05-01	75.618072	62.903226	136.867098	46.266129	34.611559	27.508065	4.634409
...	...	...	...	...	...	...	...
2019-03-01	30.952285	42.575403	90.038978	34.875134	20.914651	11.297043	4.895270
2019-04-01	25.231944	46.223750	84.910278	38.126528	32.253750	22.804444	12.417270
2019-05-01	24.680914	39.665726	77.507527	35.689516	19.479973	10.655914	11.511081
2019-06-01	22.002361	34.860139	68.594583	31.384444	17.953333	9.202222	4.485190
2019-07-01	23.389785	36.190054	72.054570	28.249328	18.469086	9.086156	5.738166

139 rows × 7 columns

Figura 4.5: Conjunto de datos para la zona de Roadside

Metric Month	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)
2008-01-01	21.400840	42.338710	55.562149	36.942204	18.817204	13.289275	3.572581
2008-02-01	21.400840	60.237069	55.562149	26.425287	31.896552	13.289275	6.734195
2008-03-01	21.400840	39.801075	55.562149	50.227151	15.477151	13.289275	2.286290
2008-04-01	21.400840	44.009722	55.562149	50.133333	21.729167	13.289275	3.236111
2008-05-01	21.400840	44.141129	55.562149	60.512097	29.545699	16.576826	4.250000
...	...	...	...	...	...	...	...
2019-03-01	7.719355	25.699328	36.038306	52.689382	16.628360	10.874462	2.271774
2019-04-01	6.927083	31.455278	39.896806	54.743472	29.204861	23.119306	2.762639
2019-05-01	5.481048	22.446371	29.282258	53.678360	15.331989	10.883871	1.535484
2019-06-01	4.969306	19.414583	24.890417	51.814722	14.112083	9.464583	1.824861
2019-07-01	4.828629	18.378629	24.388710	50.670833	13.937231	8.935349	2.576075

139 rows × 7 columns

Figura 4.6: Conjunto de datos para la zona de Background

4.2. Análisis del conjunto

Una vez que ya tenemos los datos listos para trabajar con ellos, vemos algunas estadísticas de cada conjunto usando la función `describe()`.
Esta función nos muestra el número total de datos que tiene cada conjunto, los valores máximo y mínimo que alcanzan, sus percentiles y la media.

```
pr.describe()
pr_LMR.describe()
pr_LMB.describe()
```

Metric	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)
count	278.000000	278.000000	278.000000	278.000000	278.000000	278.000000	278.000000
mean	48.509456	44.871616	96.214623	32.347221	22.127952	14.465143	3.354776
std	32.541667	13.348235	44.976091	11.227874	5.736647	4.827799	1.468350
min	4.172043	18.378629	24.388710	10.658199	11.926882	6.394624	-1.686945
25%	21.181215	33.740499	58.117118	23.580712	17.732359	11.316717	2.522080
50%	48.509456	45.541673	96.214623	31.261280	21.512097	13.136111	3.246013
75%	65.804859	55.558686	123.723750	40.156564	25.193407	16.819231	4.032829
max	180.933333	75.922414	250.743414	62.561828	43.314919	32.580780	12.417270

Figura 4.7: Estadísticas para el conjunto de datos completo

Metric	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)
count	139.000000	139.000000	139.000000	139.000000	139.000000	139.000000	139.000000
mean	75.618072	55.206747	136.867098	27.314557	25.009654	15.607172	3.408252
std	27.071310	8.267615	30.079302	8.334305	5.211375	4.912187	1.818686
min	22.002361	34.860139	68.594583	10.658199	16.284946	7.897849	-1.686945
25%	60.718616	48.700911	118.495551	21.167986	21.474097	12.340067	2.419892
50%	75.618072	55.540995	136.867098	26.432796	23.825278	14.240591	3.266129
75%	92.146976	60.256463	152.221647	34.130159	27.867357	18.185906	4.130997
max	180.933333	75.922414	250.743414	46.266129	43.314919	32.580780	12.417270

Figura 4.8: Estadísticas para la zona de Roadside

Metric	Nitric Oxide (ug/m3)	Nitrogen Dioxide (ug/m3)	Oxides of Nitrogen (ug/m3)	Ozone (ug/m3)	PM10 Particulate (ug/m3)	PM2.5 Particulate (ug/m3)	Sulphur Dioxide (ug/m3)
count	139.000000	139.000000	139.000000	139.000000	139.000000	139.000000	139.000000
mean	26.081464	34.536486	62.581281	37.379685	19.246250	13.323113	3.301301
std	16.543293	8.608747	25.278574	11.513500	4.708451	4.475180	1.007147
min	4.172043	18.378629	24.388710	13.869489	11.926882	6.394624	1.079167
25%	12.161806	27.780334	39.890943	29.098320	16.080000	9.962231	2.647872
50%	20.907500	33.682222	57.697083	36.647177	18.100000	12.050806	3.230952
75%	41.860618	40.860484	85.455491	46.541828	21.512097	15.121983	4.008533
max	79.245296	60.237069	129.152285	62.561828	36.932661	29.912366	6.734195

Figura 4.9: Estadísticas para la zona de Background

A continuación, usamos una gráfica de líneas. Estos gráficos se usan para mostrar valores cuantitivos a lo largo de un periodo de tiempo.

```
start_date = datetime(2008, 1, 1)
end_date = datetime(2019, 12, 31)

date_filter = (pr_LMR.index >= start_date) & (pr_LMR.index <= end_date
    )

fig, ejes = plt.subplots(7, 1, figsize=(11, 15), sharex=True)
for name, metric, eje in zip(names, metrics, ejes):
    sns.lineplot(data=pr_LMR[date_filter], x="Month", y=metric, ax=eje
        , color=u'#1f77b4')
    sns.lineplot(data=pr_LMB[date_filter], x="Month", y=metric, ax=eje
        , color=u'#ff7f0e')
    eje.set_title(name)
    if eje != ejes[-1]:
        eje.set_xlabel('')
```

Vemos en la figura 4.10 como se comporta cada una de las métricas a lo largo del tiempo. Hemos mostrado los valores del conjunto de datos de la zona Roadside en azul y los de la zona de Background en naranja.

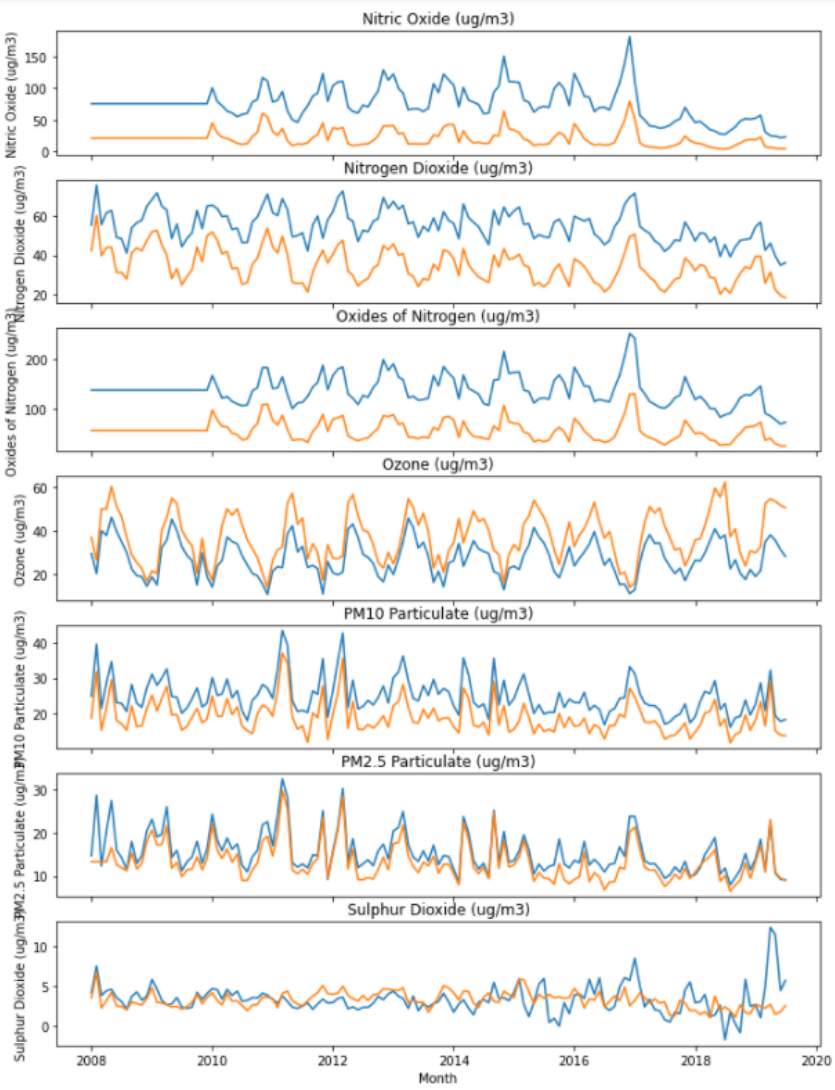

Figura 4.10: Concentración de las distintas partículas a lo largo del tiempo. La zona de Roadside se ve en azul y la de Background en naranja

La visualización que hacemos es una gráfica de líneas simple, en el que ponemos en el eje x los meses y en el eje y los valores de las distintas partículas.

Podemos observar que ninguna de las partículas muestra una tendencia, aunque parece que si existen ciertos ciclos de comportamiento en algunas partículas como es el caso del ozono. Además, vemos a simple vista como todas las partículas tienen un comportamiento casi idéntico para ambas zonas.

Los valores difieren mucho entre las distintas zonas para el óxido nítri-

co, el dióxido de nitrógeno y los óxidos de nitrógeno. Para el ozono, y las partículas PM10 existe menos variación entre los valores. Para las partículas PM2.5 los valores son muy parecidos entre las dos zonas, incluso para ciertos momentos son idénticos. Sin embargo, la única partícula que varía un poco más su comportamiento es el dióxido de azufre, ya que al principio de la serie tiene valores muy similiares en ambas zonas, pero conforme se va desarrollando la serie, los valores van variando dependiendo del lugar.

Una vez visto esto, mostramos la matriz de correlación entre las distintas métricas de cada dataframe. Una matriz de correlación nos muestra la dependencia/relación que hay entre las distintas variables y sus valores serán entre -1 y 1.

Un valor positivo muestra una correlación positiva, es decir, ambas tienden a aumentar o disminuir de igual forma. Cuanto más cercano a uno, mayor correlación positiva tendrán.
Un valor negativo muestra una correlación negativa, es decir, cuando una aumenta la otra disminuye. Cuanto más cercano a menos uno, mayor correlación negativa tendrán.

Si el valor es 0, entonces no existe una correlación entre las variables.

```
pr_simpl = pr.iloc[:,:]
pr_simpl_LMR = pr_LMR.iloc[:,:]
pr_simpl_LMB = pr_LMB.iloc[:,:]

correlation = pr_simpl.corr()
correlation_LMR = pr_simpl_LMR.corr()
correlation_LMB = pr_simpl_LMB.corr()

datas=[correlation, correlation_LMR, correlation_LMB]

fig, ejes = plt.subplots(3, 1, figsize=(11, 10), sharex=True)
for name, dat, eje in zip(['Ambas zonas', 'London Mean Roadside', '
    London Mean Background'],datas, ejes):
    sns.heatmap(data=dat, annot = True, ax=eje)
    eje.set_title(name)
    if eje != ejes[-1]:
        eje.set_xlabel('')
```

Para ello lo primero que hacemos es seleccionar todas las líneas de los distintos conjuntos y a continuación usamos la función `corr()` para obtener las correlaciones por pares entre las columnas de cada dataset.

Finalmente mostramos los resultados con un mapa de calor, en el que se ven facilmente la relación que tienen.

Podemos observar en la figura 4.11 las relaciones que existen entre las distintas variables, dependiendo del conjunto de datos en el que nos encontremos. Vamos a prestar atención aquellas cuyo valor absoluto sea mayor que $0,5$.

En la primera matriz, se muestran las relaciones entre las distintas variables en ambas zonas. Vemos que existe una gran correlación positiva entre el óxido nítrico, el dióxido de nitrógeno y los óxidos de nitrógeno. También existe otra gran correlación positiva entre las partículas PM10 y PM2.5. Hay también una correlación positiva bastante fuerte entre las partículas PM10 con el óxido nítrico, con los óxidos de nitrógeno y con el dióxido de nitrógeno. Además, vemos otra relación entre el dióxido de nitrógeno y las partículas PM2.5.
Las correlaciones negativas más importantes que encontramos son las del ozono con el óxido nítrico, con los óxidos de nitrógeno y con el dióxido de nitrógeno.

En la segunda matriz, se muestran las relaciones entre las distintas variables en la zona Roadside. Existe una gran correlación positiva entre las partículas PM10 y PM2.5. Hay también una correlación positiva bastante fuerte entre el dióxido de nitrógeno y las partículas PM10 y PM2.5. Vemos que entre el óxido nítrico, el dióxido de nitrógeno y los óxidos de nitrógeno también existe una correlación mayor a 0,5.
Las correlaciones negativas más importantes que encontramos son las del ozono con el óxido nítrico y con los óxidos de nitrógeno.

En la tercera matriz, se muestran las relaciones entre las distintas variables en la zona Background. Existe una gran correlación positiva entre las partículas PM10 y PM2.5. Hay también una correlación positiva bastante fuerte entre el dióxido de nitrógeno y las partículas PM10 y PM2.5. Vemos que entre el óxido nítrico, el dióxido de nitrógeno y los óxidos de nitrógeno también existe una correlación mayor a 0,5.
Las correlaciones negativas más importantes que encontramos son las del ozono con el óxido nítrico, con los óxidos de nitrógeno y con el dióxido de nitrógeno.

Podemos apreciar que en ninguno de los conjuntos existe una gran relación ni positiva ni negativa del dióxido de azufre con las demás partículas.

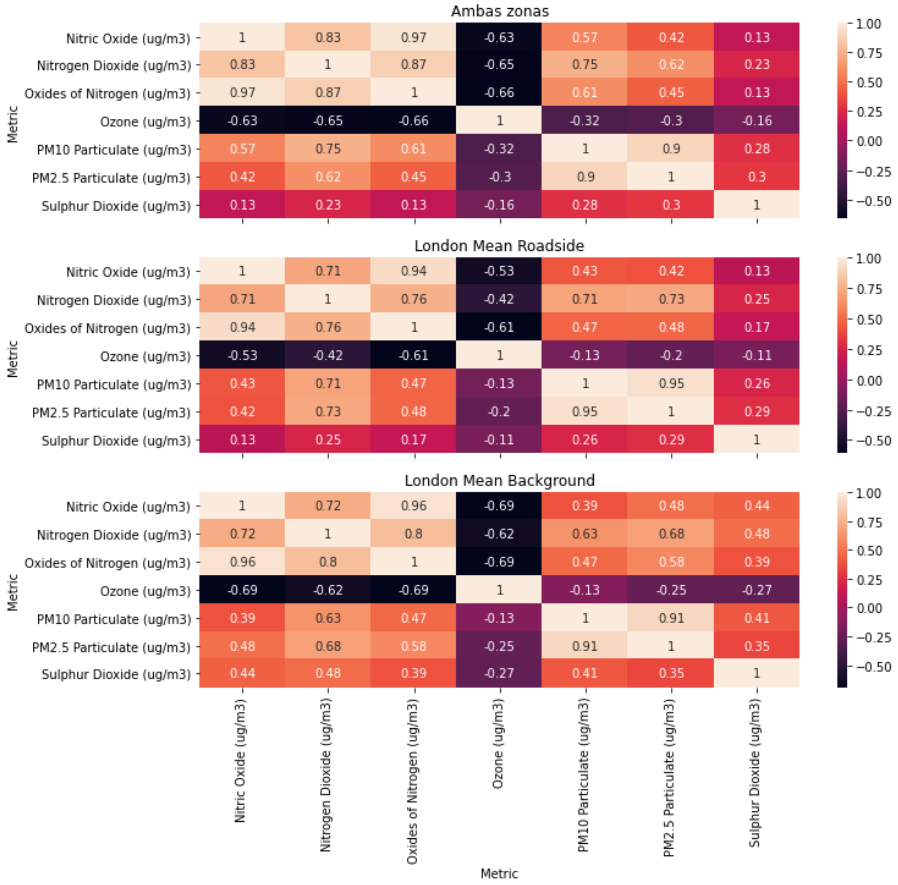

Figura 4.11: Correlación entre las distintas variables. Primer mapa de calor para el conjunto de datos completo. Segundo mapa de calor para la zona de Roadside. Tercer mapa de calor para la zona de Background. El ozono es la única partícula que tiene una correlación negativa con las demás partículas. Las partículas con óxidos tienen un alta correlación positiva entre ellas, al igual que tienen las partículas PM10 y PM2.5.

Una vez vistas las matrices de correlación, decidimos ver las relaciones entre varias variables con la matriz de dispersión. Este tipo de gráfico nos permite ver tanto la distribución de las variables de forma independiente como la relación entre dos variables. Estas matrices son muy útiles para identificar tendencias.

El resultado que obtengamos con este tipo de diagrama es muy diverso, vemos las posibilidades que podemos encontrar y su significado en la figura 4.12. Si observamos que los puntos están puestos al azar, sin seguir un patrón, entonces significa que no hay relación entre las variables. Si formasen algún patrón, podemos decir que existe algún tipo de relación.

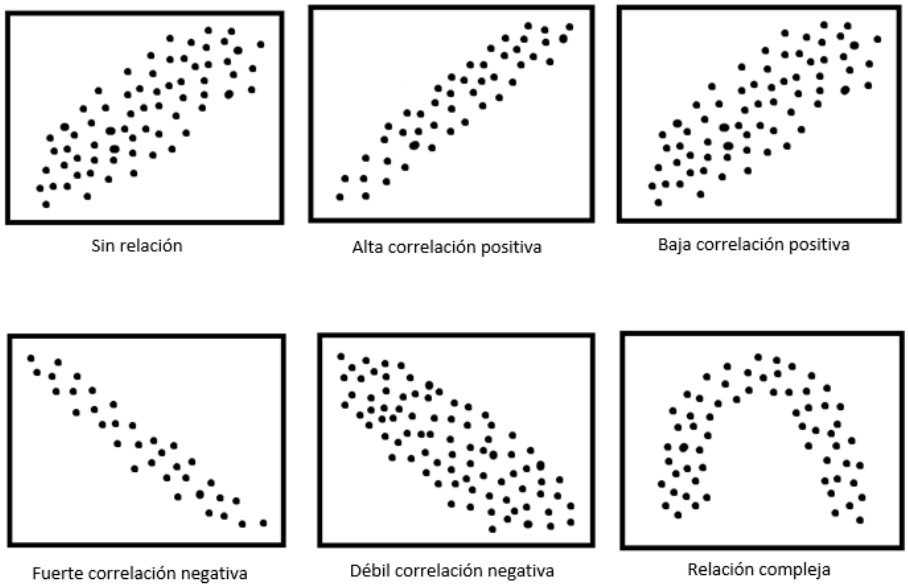

Figura 4.12: Posibles relaciones existentes que nos podemos encontrar en un diagrama de dispersión y su significado.

Para obtener estas gráficas solo tenemos que usar la función `pairplot` de la libreria `seaborn`. Si queremos distinguir por colores alguna variable categórica como es la zona para el conjunto de datos que contiene ambas zonas, lo indicamos con la opción `hue`.

```
sns.pairplot(pr.dropna(), hue='Zone',height=4,vars=metrics,kind='
    scatter')

sns.pairplot(pr_LMR.dropna(),height=4,vars=metrics,kind='scatter')

sns.pairplot(pr_LMB.dropna(),height=4,vars=metrics,kind='scatter')
```

Podemos ver los resultados en la figura 4.13. Esta visualización, como hemos comentado antes, tiene dos tipos de gráficos distintos y muestra el gráfico de dispersión para ambas zonas con el conjunto de datos PR.

El primero de ellos se encuentra en la diagonal de la matriz de gráficos. Este histograma muestra la distribución de cada variable por separado. los gráficos de dispersión nos muestran la relacion existente entre las partículas.

Si observamos las visualizaciones en las figuras 4.14 y 4.15, vemos que concuerda con los resultados obtenidos en la matriz de correlación. Además, podemos observar una pequeña línea recta de puntos para el óxido nítrico y

los óxidos de nitrógeno. Esto es debido a los valores nulos que teníamos al principio y rellenamos con la media de cada partícula.

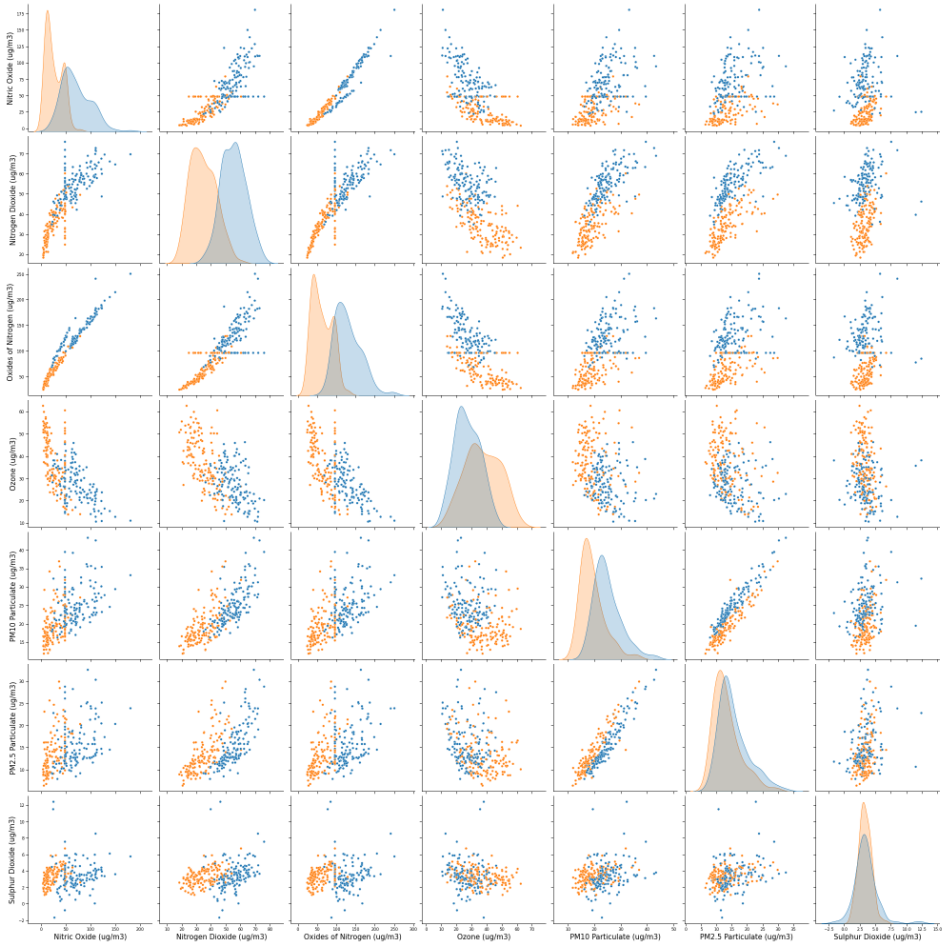

Figura 4.13: Matriz de dispersión para el conjunto de datos completo. El ozono es la única partícula que tiene una correlación negativa con las demás partículas. Las partículas con óxidos tienen un alta correlación positiva entre ellas, al igual que tienen las partículas PM10 y PM2.5.

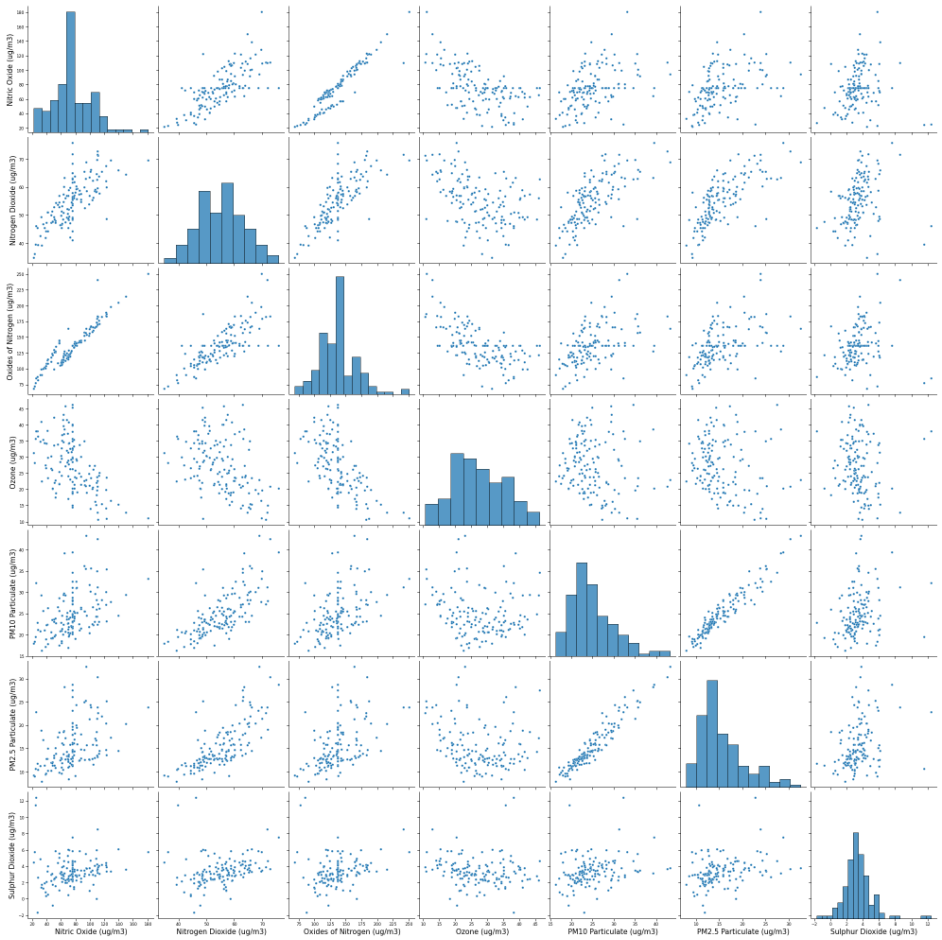

Figura 4.14: Matriz de dispersión para la Zona de Roadside. El ozono es la única partícula que tiene una correlación negativa con las demás partículas. Las partículas con óxidos tienen un alta correlación positiva entre ellas, al igual que tienen las partículas PM10 y PM2.5.

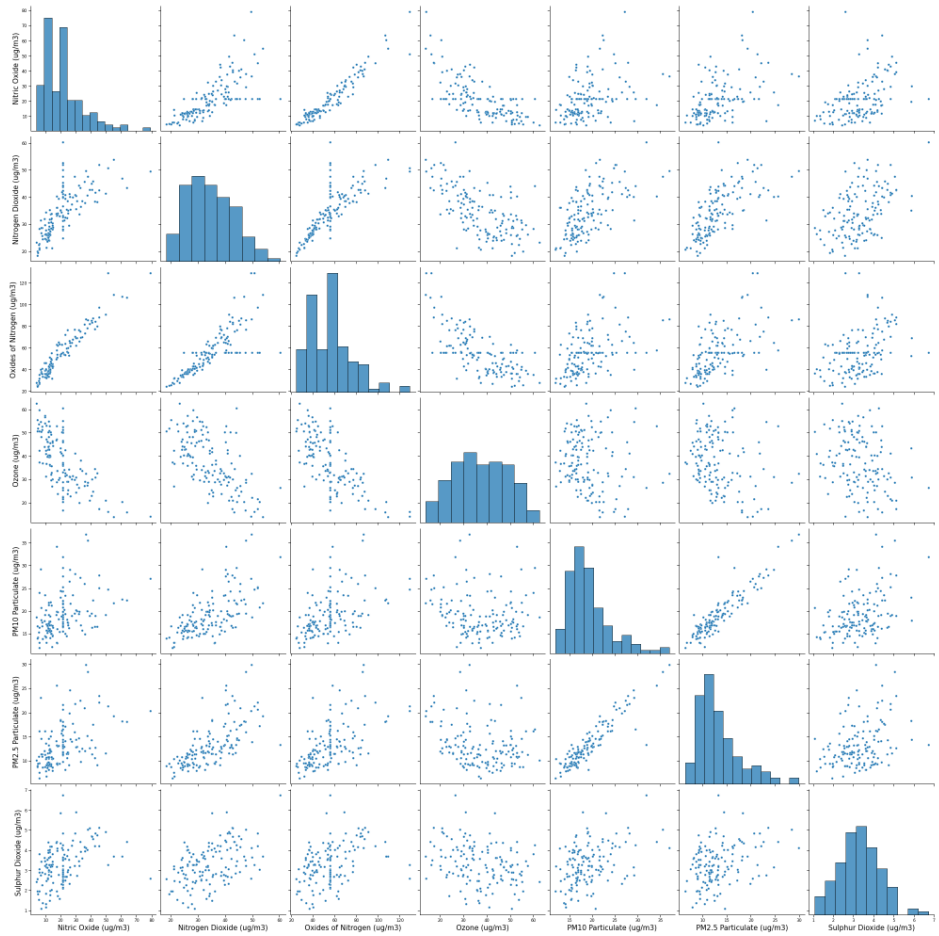

Figura 4.15: Matriz de dispersión para PR la zona Background. El ozono es la única partícula que tiene una correlación negativa con las demás partículas. Las partículas con óxidos tienen un alta correlación positiva entre ellas, al igual que tienen las partículas PM10 y PM2.5.

En relación con la correlación entre las distintas métricas, otra representación que podemos hacer para ver si el comportamiento es parecido o no entre las distintas variables es un gráfico de líneas para cada zona, en el que mostramos en el eje x los meses y en el eje y los valores que alcanzan las partículas. Dibujaremos cada métrica de un color diferente. De esta forma podemos ver visualmente la relación entre las variables a lo largo del tiempo.

```
pr_LMR.plot(figsize=(15,10))
plt.title('London Mean Roadside')
plt.show()
pr_LMB.plot(figsize=(15,10))
plt.title('London Mean Background')
plt.show()
```

Si nos fijamos en las figuras 4.16 y 4.17, podemos confirmar lo visto con anterioridad.Vemos como las métricas con una alta correlación positiva crece y decrecen en sintonía. Sin embargo, las métricas con una correlación negativa, crecen y decrecen de forma antisimétrica.

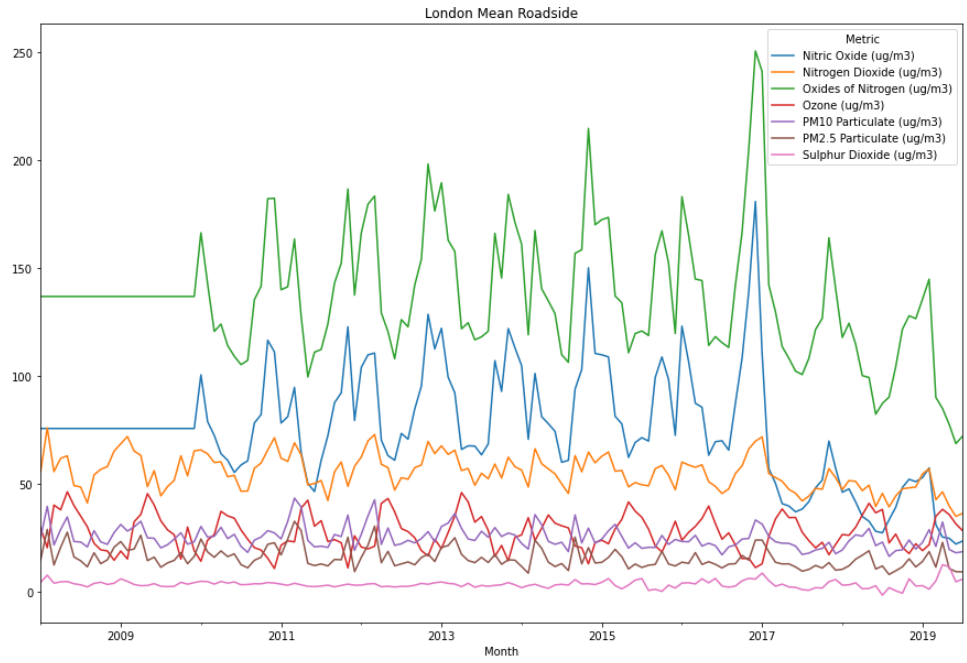

Figura 4.16: Comportamiento de todas las partículas en la zona de London Mean Roadside.

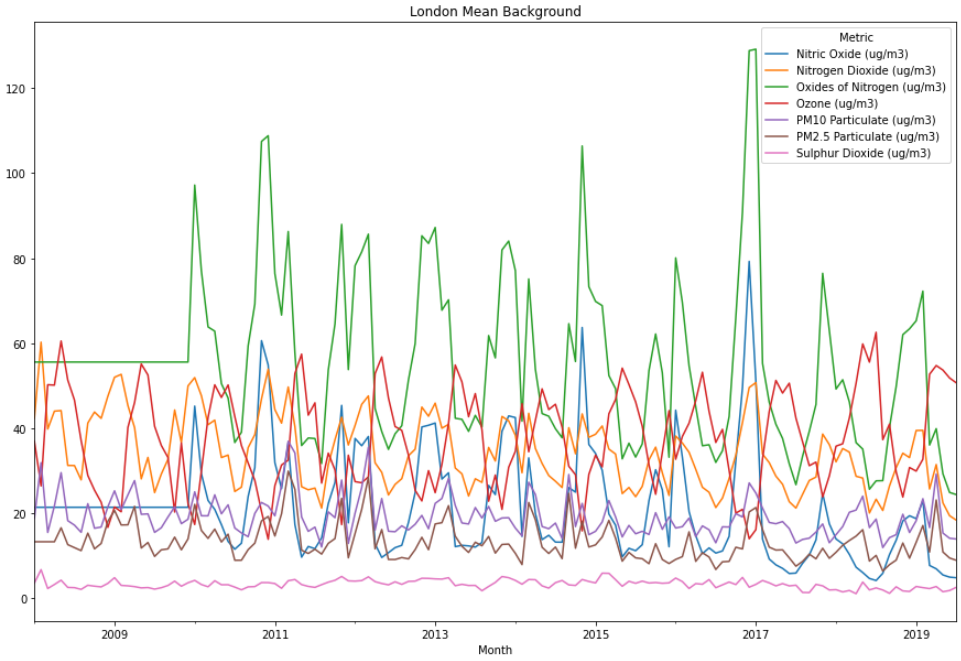

Figura 4.17: Comportamiento de todas las partículas en la zona de London Mean Background

A continuación, mostramos las distribuciones de los valores para cada una de las partículas. El gráfico que usamos es un histograma, que se representan con un diagrama de barras, en nuestro caso verticales, y nos muestra como se encuentran repartidos los valores de nuestro conjunto de datos.

En el eje y mostramos la cantidad de valores que hay con el valor que se muestra en el eje x.

```
print('Histograma para ambas zonas')
fig, ejes = plt.subplots(8, 1, figsize=(21, 40), sharex=True)
g = sns.histplot(pr, bins=50)
g.set_title('Histograma para Ambas Zonas')
for name, metric, eje in zip(names, metrics, ejes):
    sns.histplot(pr[metric], bins=50, ax=eje)
    eje.set_title(name)
    if eje != ejes[-1]:
        eje.set_xlabel('')

print('Histograma para London Mean Background')
fig, ejes = plt.subplots(8, 1, figsize=(21, 40), sharex=True)
g = sns.histplot(pr_LMB, bins=50)
g.set_title('Histograma para LMB')
for name, metric, eje in zip(names, metrics, ejes):
    sns.histplot(pr_LMB[metric], bins=50, ax=eje)
    eje.set_title(name)
    if eje != ejes[-1]:
        eje.set_xlabel('')
```

```
20
21 print('Histograma para London Mean Roadside')
22 fig, ejes = plt.subplots(8, 1, figsize=(21, 40), sharex=True)
23 g = sns.histplot(pr_LMR, bins=50)
24 g.set_title('Histograma para LMR')
25 for name, metric, eje in zip(names, metrics, ejes):
26     sns.histplot(pr_LMR[metric], bins=50, ax=eje)
27     eje.set_title(name)
28     if eje != ejes[-1]:
29         eje.set_xlabel('')
```

Para obtener este gráfico usamos la función `histplot` a la que le pasamos como parámetros el conjunto de datos que queremos representar y el número de divisiones que queremos. En nuestro caso hemos elegido 50.

En las figuras 4.18 y 4.19 podemos ver los histogramas para cada una de las variables y para el conjunto de ellas en la zona de Roadside.

El rango de valores del óxido nítrico y de los óxidos de nitrógeno son más grandes y por tanto, tienen más variación entre los valores máximo y mínimo. En general, no tienen valores que se repitan demasiado, salvo un rango que si destaca mucho por encima de los otros.

Tanto el ozono como el dióxido de nitrógeno tienen su rango de valores un poco más acotado y no contienen rangos de valores que sobresalgan demasiado sobre los demás.

Las partículas PM2.5 y PM10 tienen un rango de valores bastante más limitado, pero tampoco contienen rangos de valores que destacar.

Finalmente, el dióxido de azufre, tiene un rango de valores muy limitado y se centran sus valores más recurrentes en los centrales.

Podemos ver los histogramas para la zona de Background en las figuras 4.20 y 4.21. En general, tienen un comportamiento muy similar al visto en Roadside, salvo porque los valores en en Background son mucho menores que en Roadside.

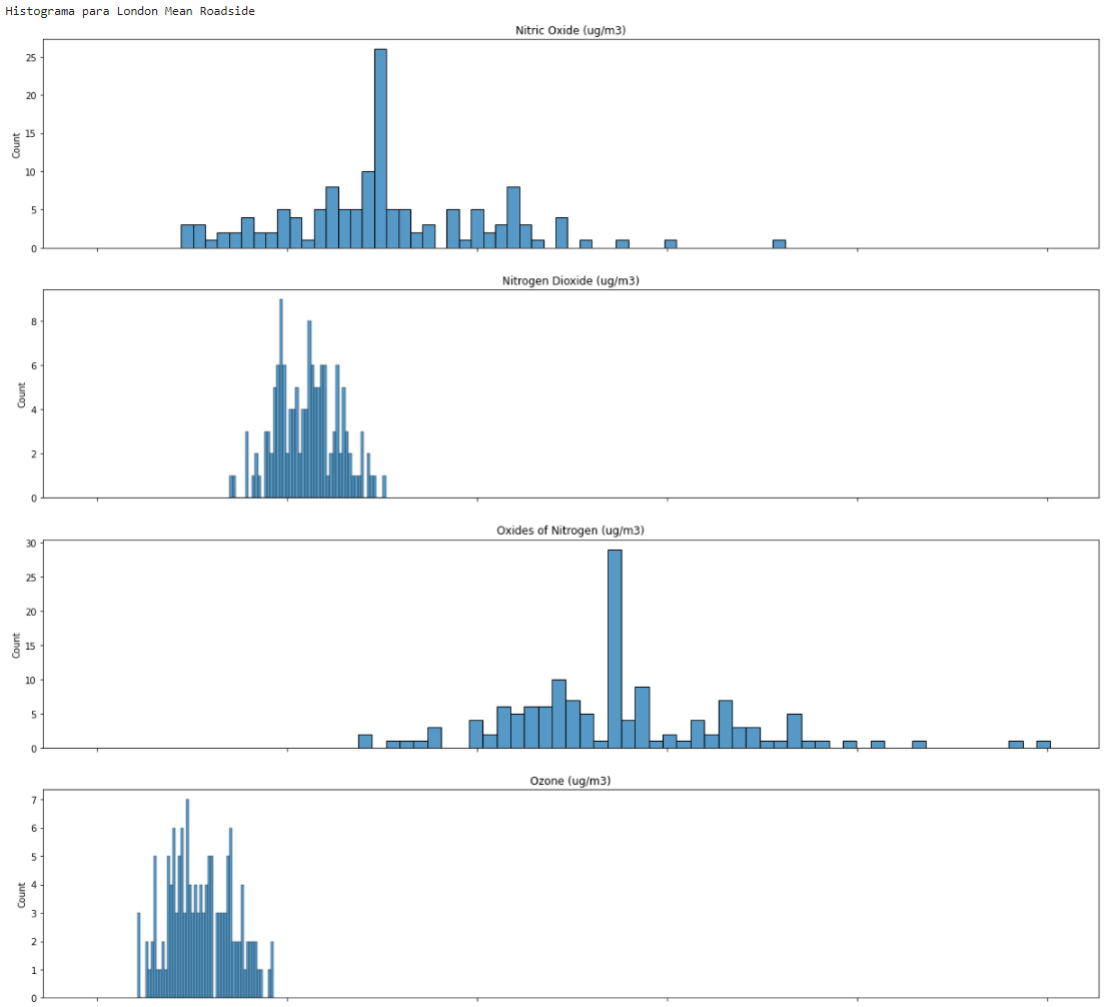

Figura 4.18: Histograma de las partículas de óxido nítrico, dióxido de nitrógeno y óxidos de nitrógeno y el ozono para la zona de Roadside.

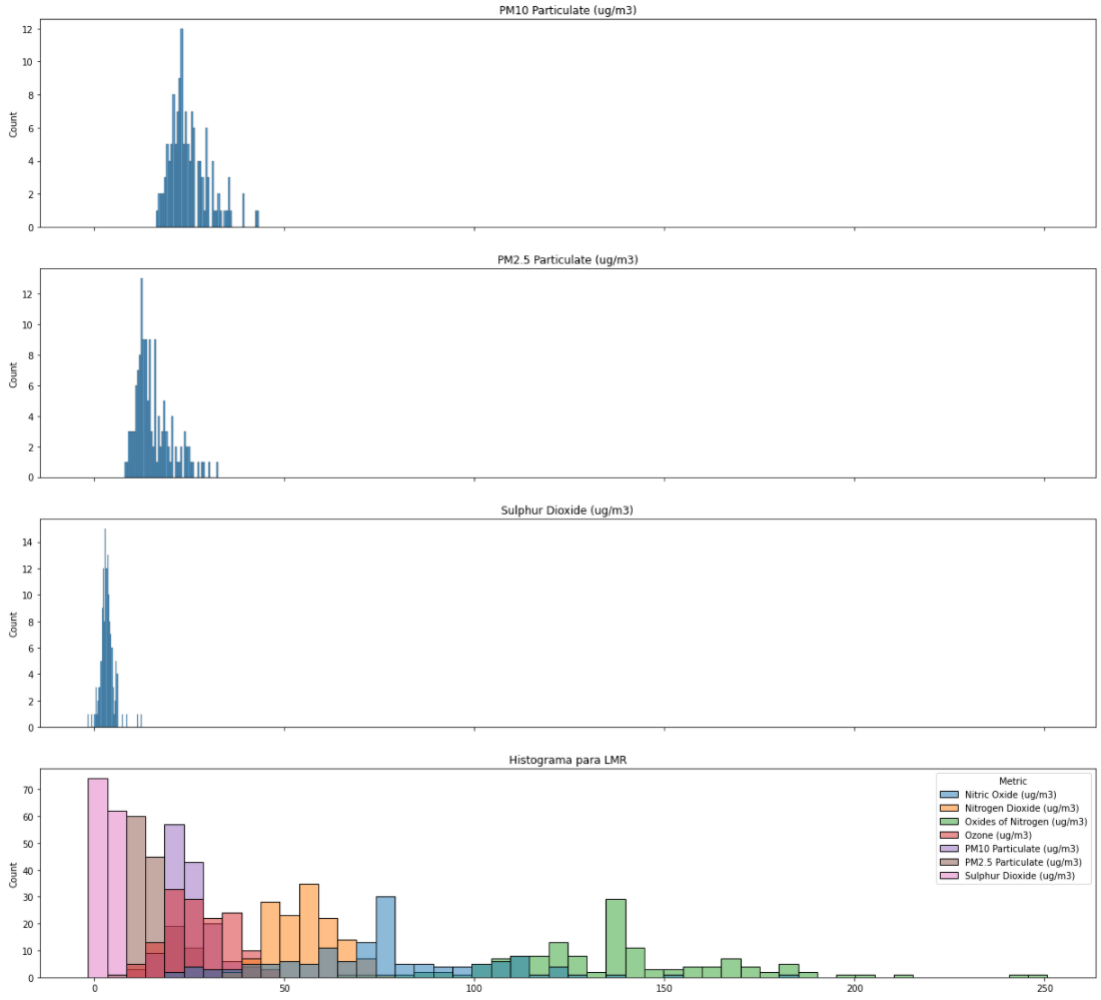

Figura 4.19: Histograma de las partículas PM10, partículas PM2.5 y dióxido de azufre para la zona de Roadside.

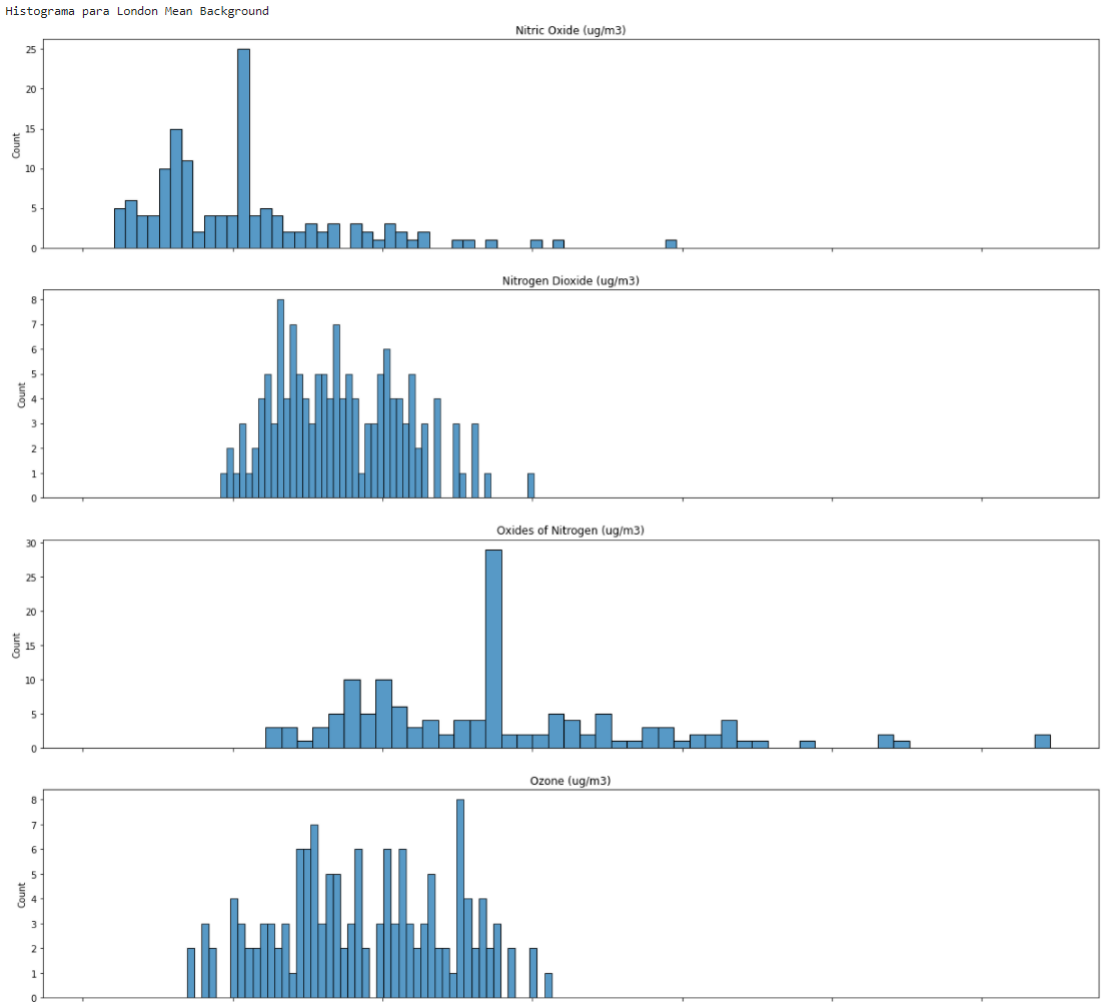

Figura 4.20: Histograma de las partículas de óxido nítrico, dióxido de nitrógeno y óxidos de nitrógeno y el ozono para la zona de Background.

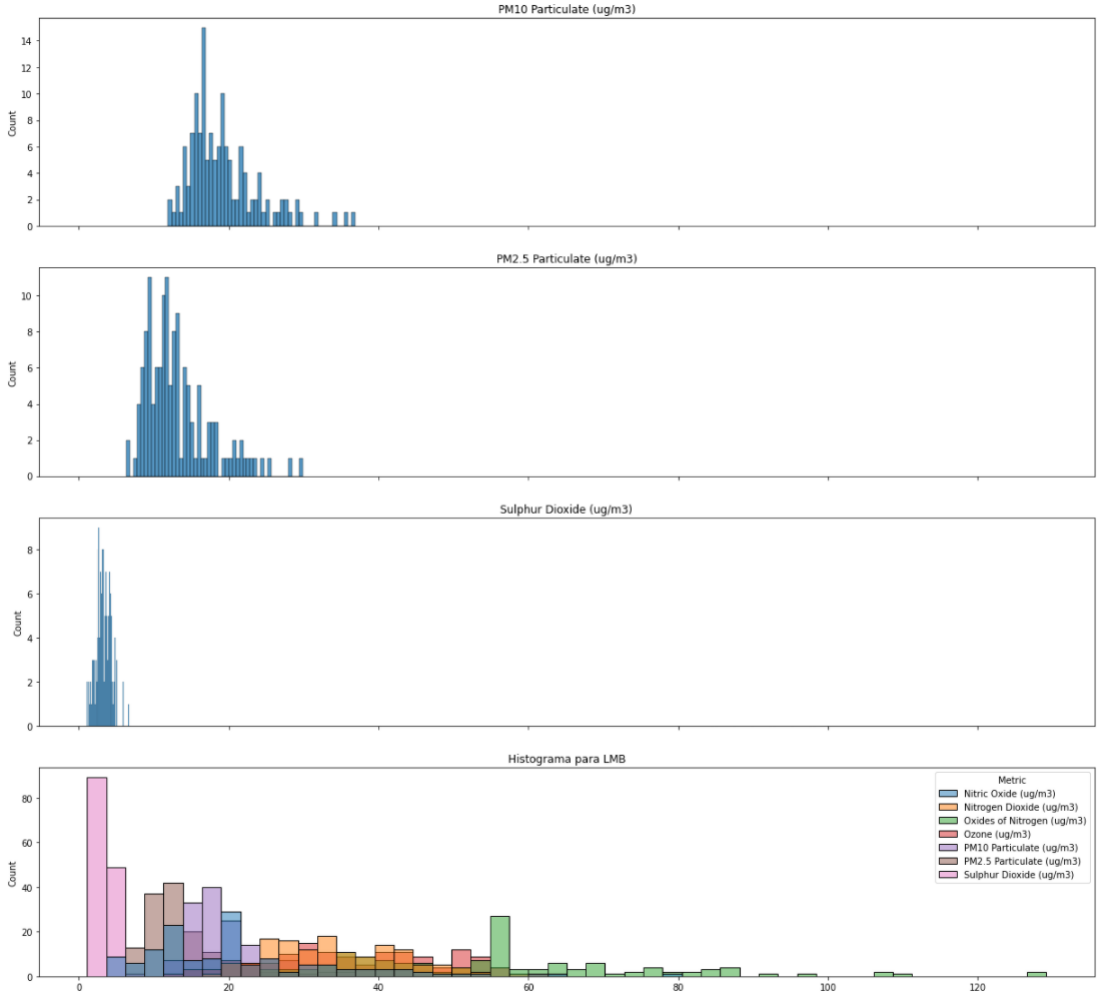

Figura 4.21: Histograma de las partículas PM10, partículas PM2.5 y dióxido de azufre para la zona de Background.

Existen varios modelo matemáticos que intentan ajustarse a los valores que se tienen. Estos, nos pueden ayudar a ver como se comportan.
Cuanto mayor sea el grado del polinomio que usemos, más se parecerá a la gráfica de nuestros datos. Sin embargo, cuanto más grande sea mayor coste computacional tendremos.

Vamos a probar con un modelo lineal y otro modelo cúbico.

El primer modelo que vamos a realizar es un polinomio de grado 1, de la forma

$$y = a + bx$$

El segundo modelo que vamos a ver es un polinomio de grado 3 de la forma

$$y = a + bx + cx^2 + dx^3$$

```
1  print('Modelo lineal para la zona de Roadside')
2  #Mostramos la pendiente y el error cuadratico medio.
3  for metric in metrics:
4      coefficients, residuals, _, _, _ = np.polyfit(range(len(pr_LMR[
           metric].index)),pr_LMR[metric],1,full=True)
5      mse = residuals[0]/(len(pr_LMR[metric].index))
6      nrmse = np.sqrt(mse)/(pr_LMR[metric].max() - pr_LMR[metric].min())
7      print('Slope for ' + metric + str(coefficients[0]))
8      print('NRMSE for ' + metric + str(nrmse))
9      plt.plot(pr_LMR[metric])
10     plt.plot(pr_LMR.index,[coefficients[0]*(x) + coefficients[1]  for
           x in range(len(pr_LMR[metric]))])
11     plt.show()
12
13 print('Modelo cubico para la zona de Roadside')
14 for metric in metrics:
15     coefficients, residuals, _, _, _ = np.polyfit(range(len(pr_LMR[
           metric].index)),pr_LMR[metric],4,full=True)
16     mse = residuals[0]/(len(pr_LMR[metric].index))
17     nrmse = np.sqrt(mse)/(pr_LMR[metric].max() - pr_LMR[metric].min())
18     print('Slope for ' + metric + str(coefficients[0]))
19     print('NRMSE for ' + metric + str(nrmse))
20     plt.plot(pr_LMR[metric])
21     plt.plot(pr_LMR.index,[coefficients[0]*(x**4) + coefficients[1]*(x
           **3) + coefficients[2]*(x**2) + coefficients[3]*(x) +
           coefficients[4]  for x in range(len(pr_LMR[metric]))])
22     plt.show()
```

Como podemos ver en las líneas 4 y 15, la función `polyfit` es la que nos da los coeficientes y los residuos para la función lineal, pasandole los parámetros del conjunto de datos y el grado con el que queremos que haga el ajuste polinomial.

En las líneas 5, 6 y 16, 17 calculamos el error cuadrático medio para a continuación imprimirlo por pantalla junto a la pendiente generada.

Finalmente dibujamos la gráfica generada por cada partícula junto a su aproximación lineal y cúbica.

Los resultados obtenidos con la aproximación lineal para la zona de Roadside la podemos observar en la figura 4.22. Los resultados para la aproximación cúbica lo vemos en la figura 4.23.

Si observamos la función polinómica de grado 1, vemos que todas las pendientes son negativas, es decir, todas las rectas son decrecientes.

Si nos centramos en el modelo cúbico, vemos que las partículas de óxido

nítrico, dióxido de nitrógeno y óxidos de nitrógeno termina la función siendo decreciente. Sin embargo, para todas las demás, el modelo termina creciendo.

Los resultados obtenidos con la aproximación lineal para la zona de Background la podemos observar en la figura 4.24. Los resultados para la aproximación cúbica lo vemos en la figura 4.25.

Viendo el modelo lineal, vemos que todas las pendientes son negativas excepto para el ozono que es positiva, es decir, todas las rectas son decrecientes, salvo en el ozono que es creciente.

Si nos centramos en el modelo cúbico, vemos que las partículas de óxido nítrico, dióxido de nitrógeno, óxidos de nitrógeno y el sulfuro de dióxido termina la función siendo decreciente. Sin embargo, para todas las demás, el modelo termina creciendo.

Ahora vamos a estudiar la estacionalidad de los datos con los diagramas de caja, de forma tanto mensual como anual.

En los diagrama de caja se representan tres cuartiles y los valores máximo y mínimo. Gracias a estos, podemos ver como es la distribución, la asimetría que tienen los valores y la posición de la mediana entre otros.

Podemos ver que se trata de una caja que está delimitada por el primer y el tercer cuartil. La línea que vemos dentro de la caja representa la mediana, es decir, el segundo cuartil.
La raya que encontramos debajo de cada caja es el mínimo y la superior el máximo. Los datos que observamos fuera de estos son valores anómalos ya que no cumple cierto requisitos de heterogeneidad.

```
print('Zona London Mean Roadside. Vista anual.')
fig, ejes = plt.subplots(7, 1, figsize=(15, 20), sharex=True)
for name, metric, eje in zip(names, metrics, ejes):
    sns.boxplot(data=pr_LMR, x=pr_LMR.index.year,y=metric,ax=eje)
    eje.set_title(name)
    if eje != ejes[-1]:
        eje.set_xlabel('')

print('Zona London Mean Roadside. Vista mensual.')
fig, ejes = plt.subplots(7, 1, figsize=(15, 20), sharex=True)
for name, metric, eje in zip(names, metrics, ejes):
    sns.boxplot(data=pr_LMR, x=pr_LMR.index.month,y=metric,ax=eje)
    eje.set_title(name)
    if eje != ejes[-1]:
        eje.set_xlabel('')
```

Para poder realizar esta visualización hemos usado la función `boxplot`, a la cual le pasamos el conjunto de datos con el que vamos a trabajar y los valores para los ejes x e y.

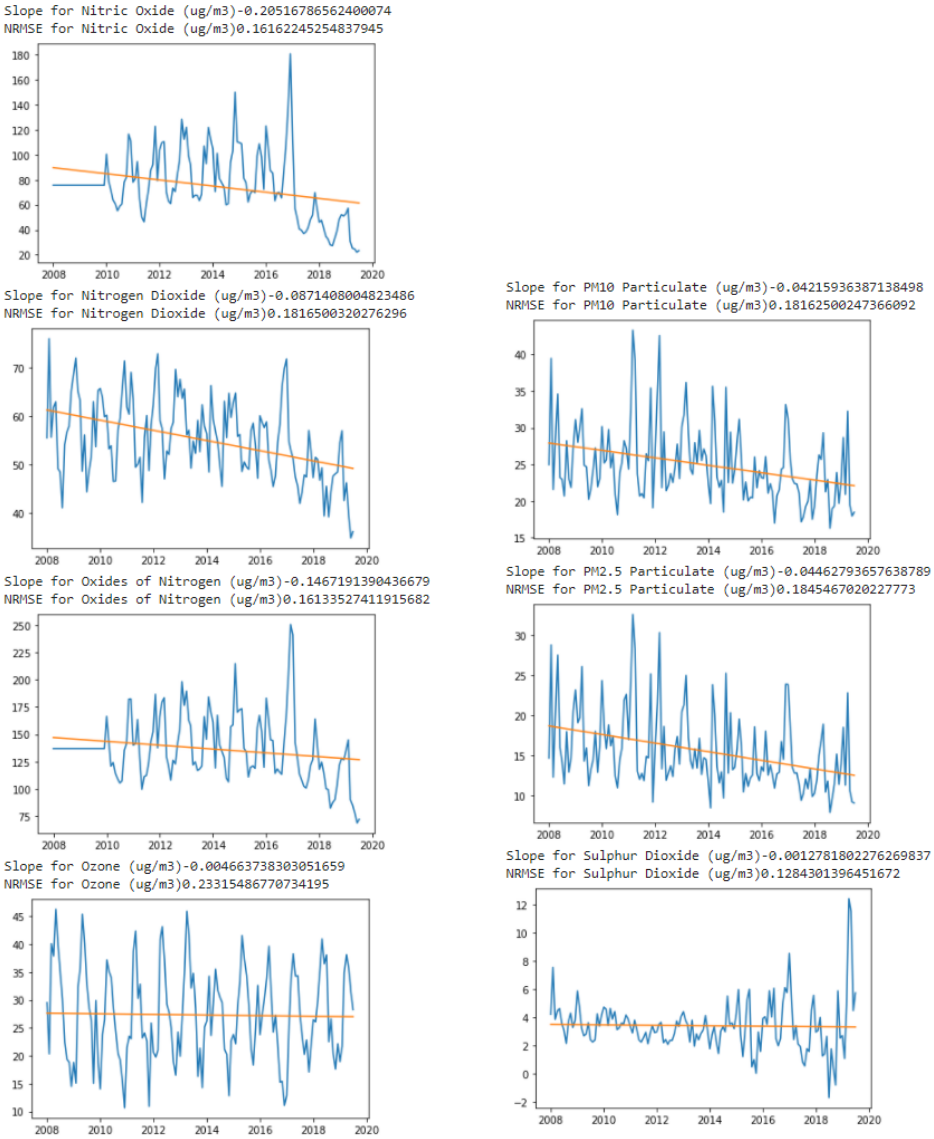

Figura 4.22: Ajuste lineal para las diferentes partículas, junto a su pendiente y su error en la zona London Mean Roadside.

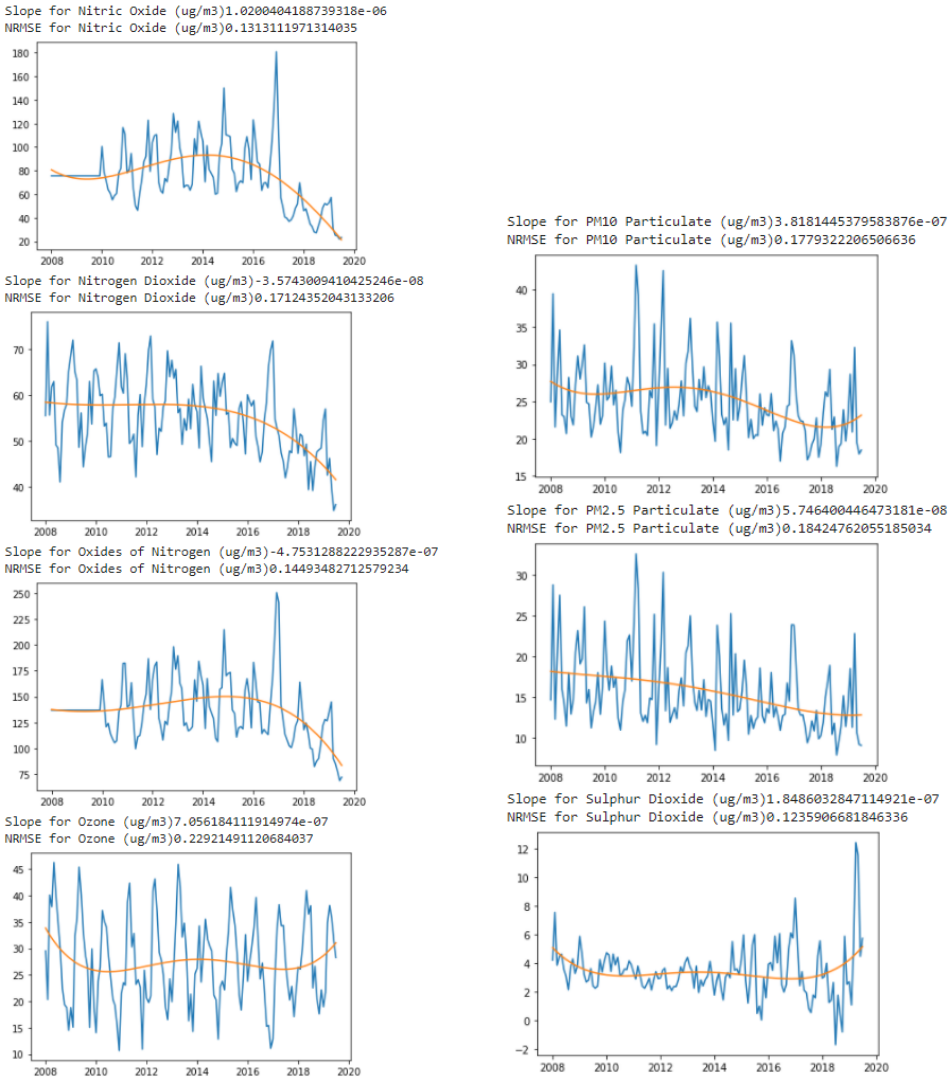

Figura 4.23: Ajuste cúbico para las diferentes partículas, junto a su pendiente y su error en la zona London Mean Roadside.

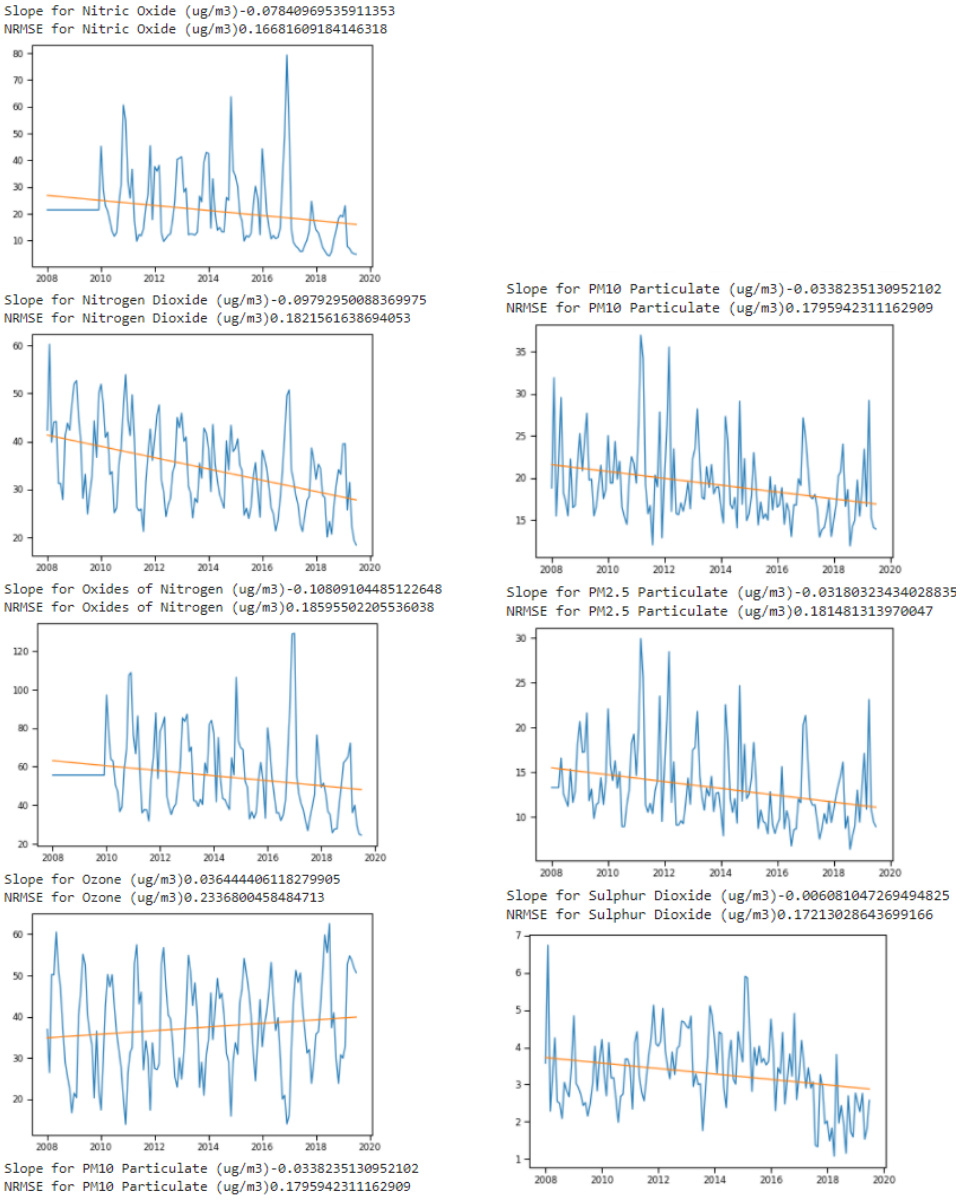

Figura 4.24: Ajuste lineal para las diferentes partículas, junto a su pendiente y su error en la zona London Mean Background.

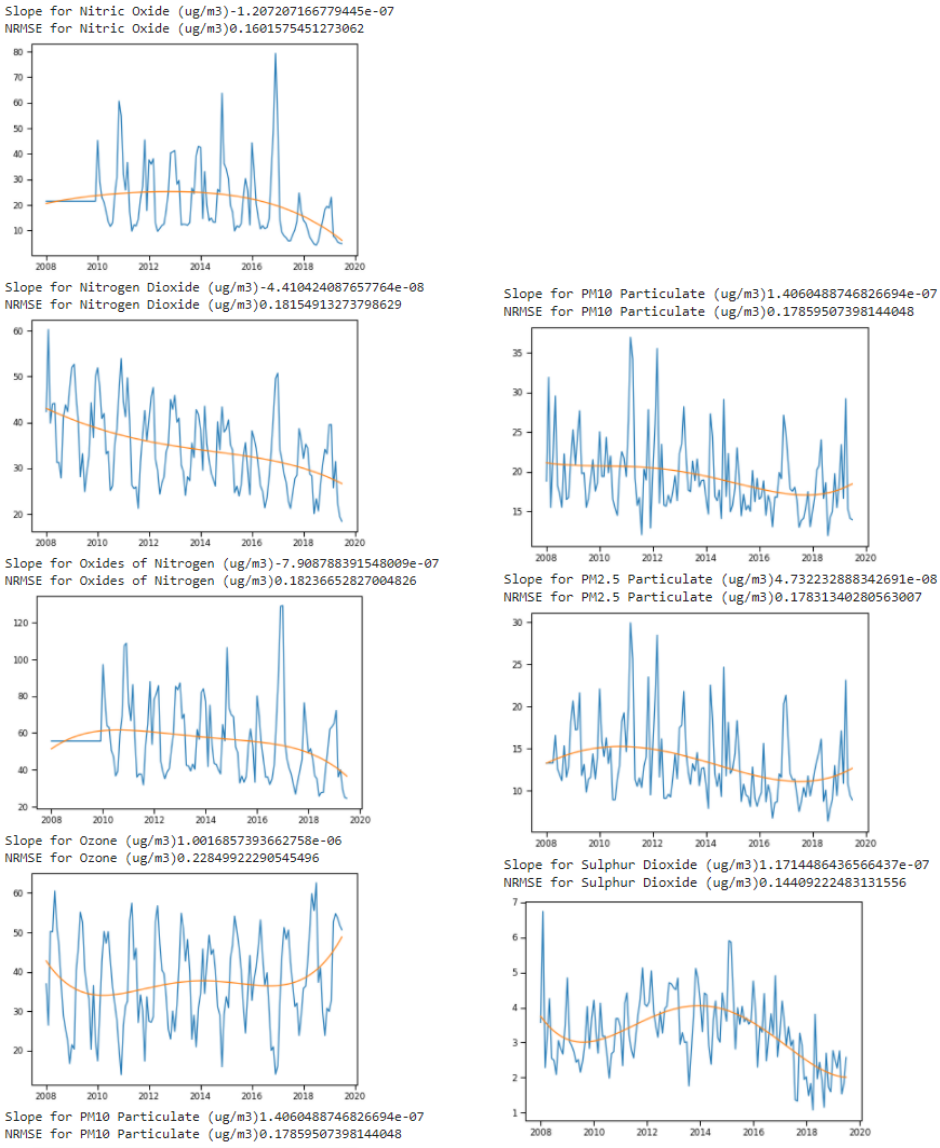

Figura 4.25: Ajuste cúbico para las diferentes partículas, junto a su pendiente y su error en la zona London Mean Background.

Para la gráficas mensuales ponemos el mes en el eje x y las métricas en el eje y, mientras que las anuales ponemos el año en el eje x.

Podemos ver en las gráficas mensuales de la zona de Roadside en la figura 4.27 que para el óxido nítrico y los óxidos de nitrógeno solo se ve una línea, esto es debido a que los dos primeros años no teníamos datos y rellenamos estos valores nulos con la media.

En la gráfica anual que podemos ver en la figura 4.26

Para la zona de background podemos ver en las gráficas mensuales que vemos en la figura 4.29

En la gráfica anual que podemos ver en la figura 4.28

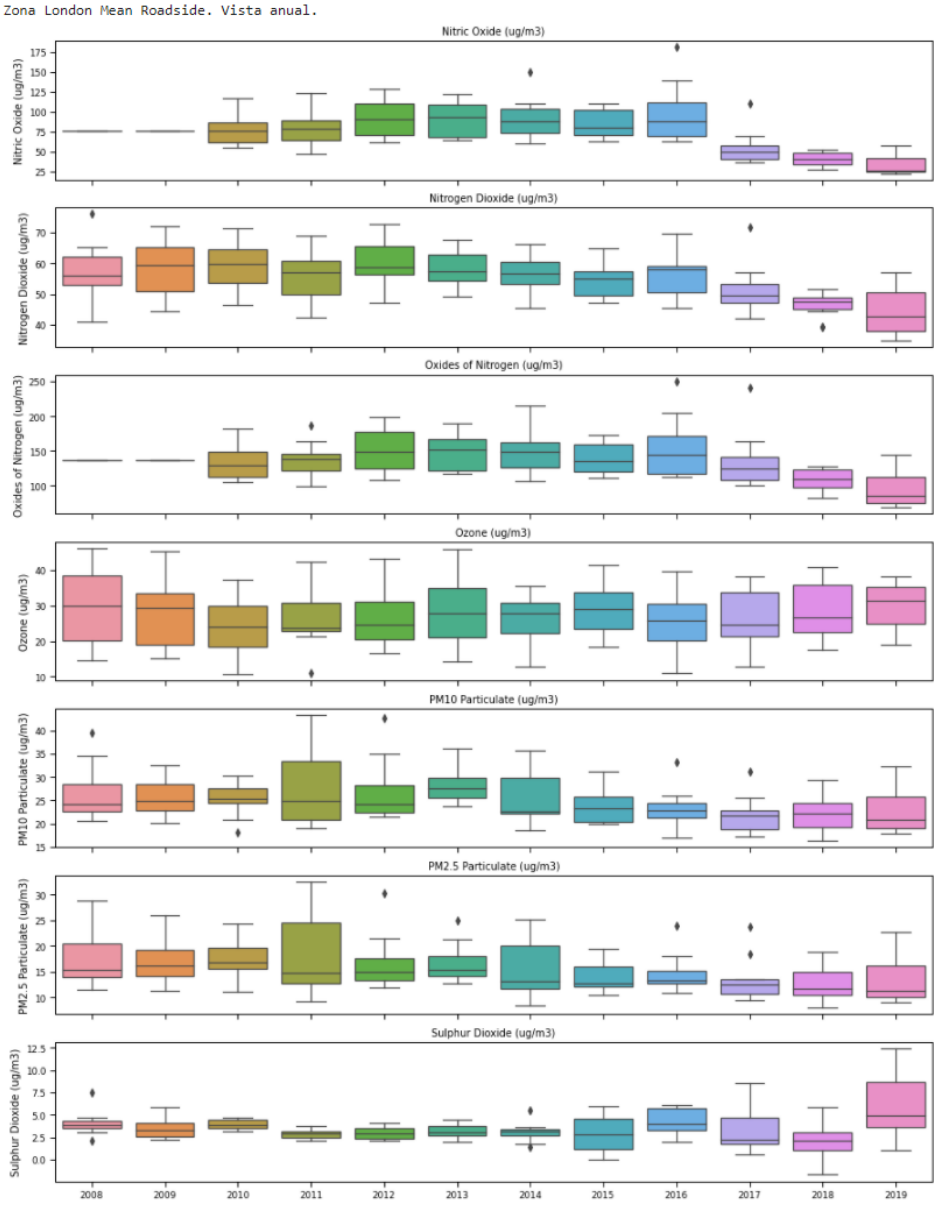

Figura 4.26: Diagrama de cajas anual para la zona Roadside.

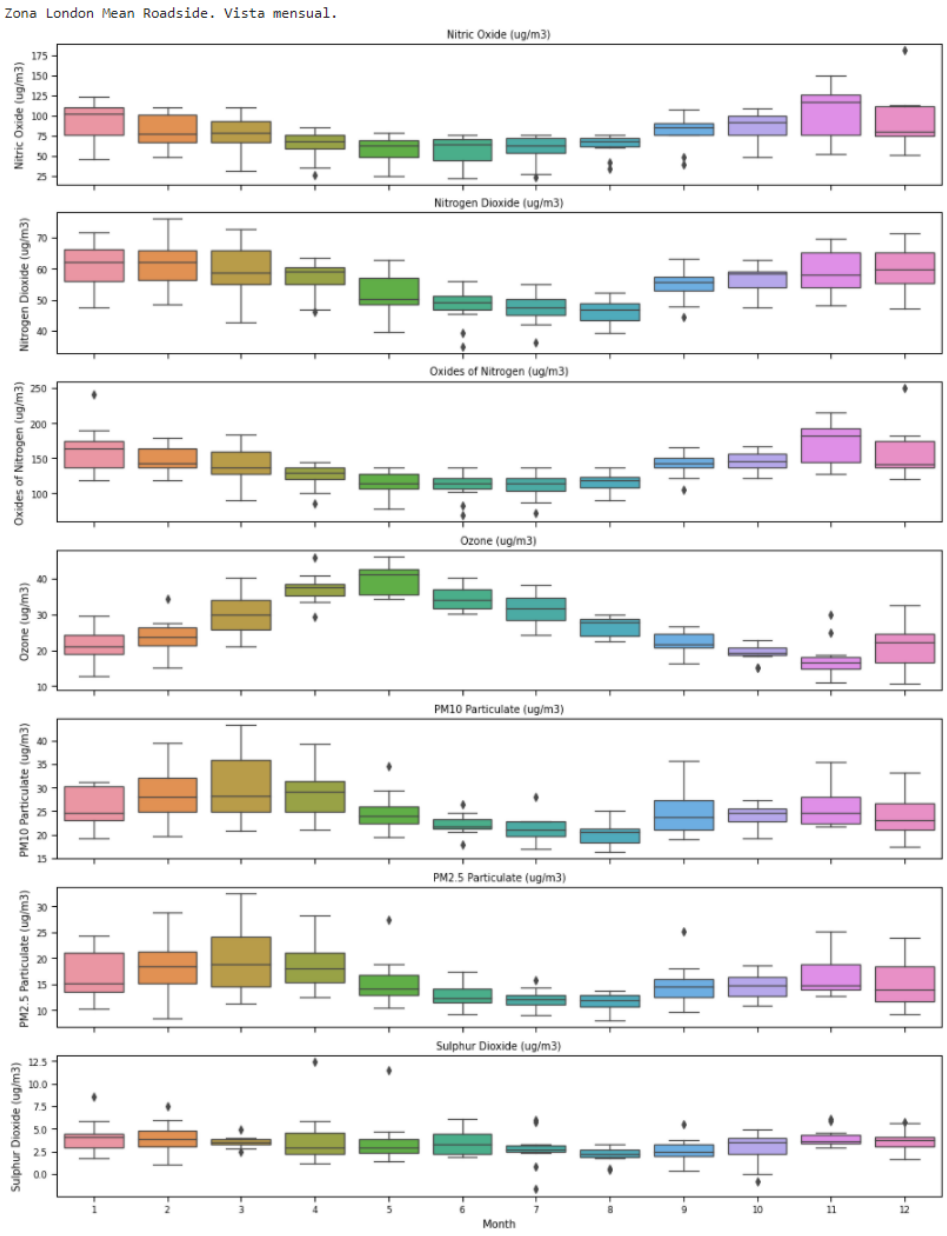

Figura 4.27: Diagrama de cajas mensual para la zona Roadside

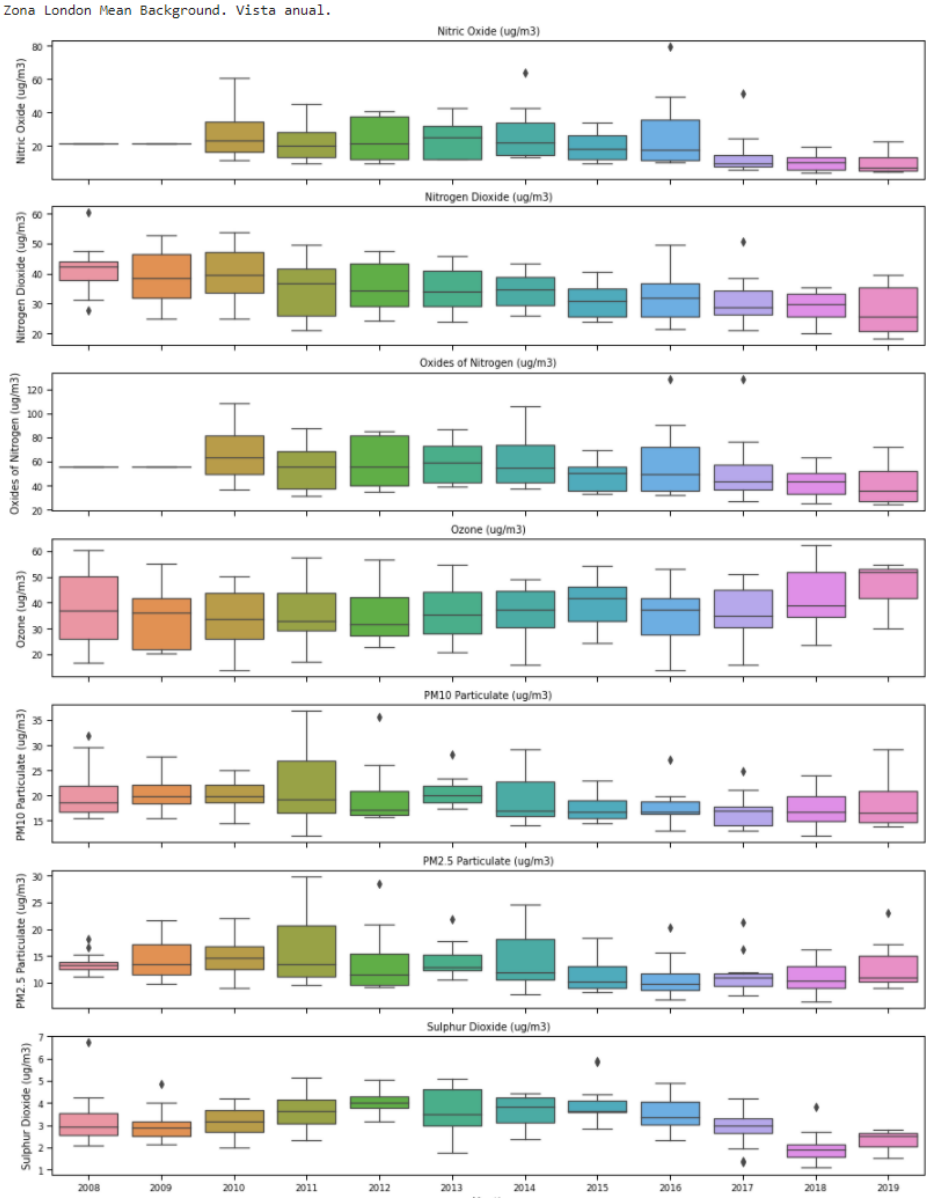

Figura 4.28: Diagrama de cajas anual para la zona Background

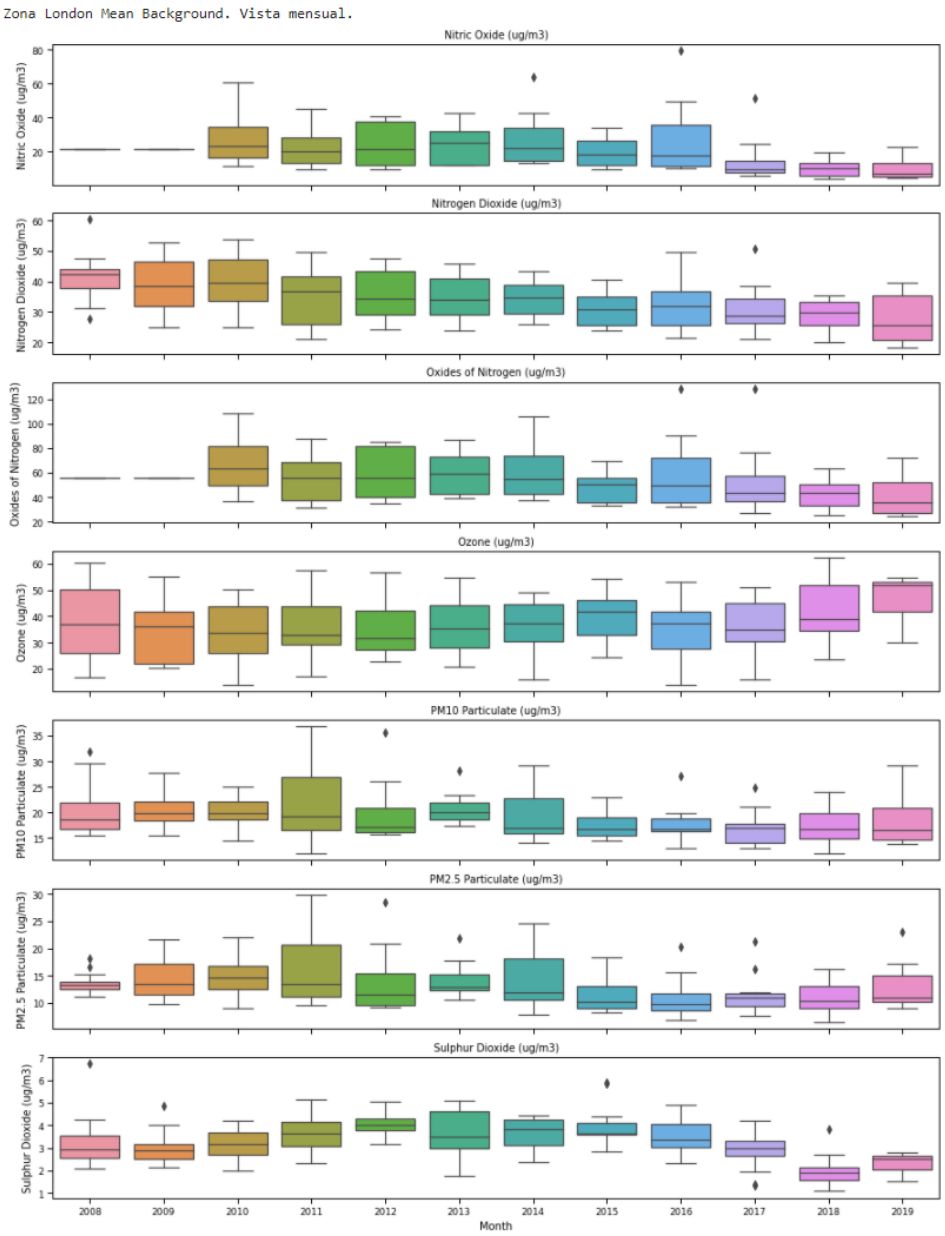

Figura 4.29: Diagrama de cajas mensual para la zona Background

Finalmente y antes de adentrarnos en la parte de predicción usando las técnicas que explicamos en el capítulo 3, hemos implementado una regresión lineal intentando predecir los valores de una variable con respecto a otra.

```
dataX = pr_LMR[['Nitric Oxide (ug/m3)']]
X_train = np.array(dataX)
y_train = pr_LMR['Oxides of Nitrogen (ug/m3)']
```

```
4
5  regr = linear_model.LinearRegression()
6
7  regr.fit(X_train,y_train)
8
9  y_pred = regr.predict(X_train)
10 # La Tangente
11 print('Coefficients: \n', regr.coef_)
12 # Termino Independiente
13 print('Independent term: \n', regr.intercept_)
14 # Error Cuadrado Medio
15 print("Mean squared error: %.2f" % mean_squared_error(y_train, y_pred)
       )
16 # Varianza
17 print('Variance score: %.2f' % r2_score(y_train, y_pred))
18
19 y_Dosmil = regr.predict([[42]])
20 print(int(y_Dosmil))
```

Para realizar esta regresión, elegimos dos métricas distintas del mismo conjunto de datos y usaremos el modelo lineal `LinearRegression()`. Usando la función `fit`, a la que le pasamos como parámetros las métricas elegidas) ajustamos el modelo lineal.

Tras esto, obtenemos la tangente y el término independiente llamando a los correspondientes parámetros y calculamos el error cuadrático medio y la varianza con sus fórmulas correspondientes.

Finalmente, calculamos los resultados con la función `predict`. Los resultados en la figura 4.30 nos muestran la tangente, el término independiente, el error cuadrático medio, la varianza y el resultado obtenido.

```
Coefficients:
 [1.03987949]
Independent term:
 58.233415513788856
Mean squared error: 111.48
Variance score: 0.88
101
```

Figura 4.30: Regresión lineal de Óxido Nítrico y Óxidos de Nitrógeno

Capítulo 5

Experimentos - Predicción

En este quinto capítulo vamos a explicar detalladamente los métodos de predicción que hemos usado sobre nuestro conjunto de datos, viendo los gráficos obtenidos y haciendo un análisis con los valores obtenidos. Podemos ver un diagrama con los pasos a realizar en la figura 5.1

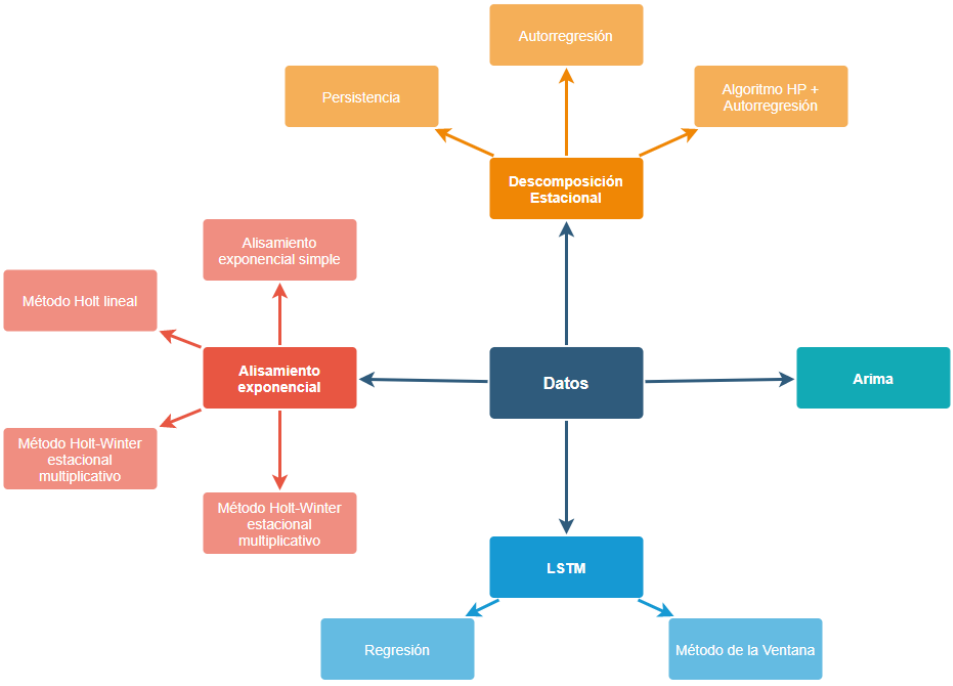

Figura 5.1: Proceso a seguir en el capítulo 5- Experimentos-Predicción

Una vez hemos hecho un análisis de los datos y tenemos una idea de su

comportamiento a lo largo de la serie, pasamos a intentar predecir futuros valores para estas variables. Para ello vamos a usar varias técnicas distintas con el objetivo de encontrar la más óptima.
Los distintos métodos que vamos a usar para predecir son descomposición estacional, alisamiento exponencial, ARIMA y finalmente LSTM.

Vamos a ver cada uno de los métodos en secciones distintas para que quede más claro y conciso los pasos usados en cada uno de ellos.

5.1. Descomposición estacional

Este es el método más simple de los que hemos implementados. Se han realizado varias versiones del mismo, intentando precisar un poco más la predicción conforme vamos progresando.

Lo primero que hacemos es ver la descomposición de las series con el método de Hodrick-Prescott (HP) y el método de Descoposición estacional y de tendencia con Loess(STL).

```
1  print('Seasonal Descomp - London Mean Roadside')
2  for name, metric in zip(names, metrics):
3    print(name)
4    print('Usamos el metodo HP para extraer la tendencia de una serie
         temporal.')
5    series = pr_LMR[metric]
6    train_pr = pr_LMR[metric].iloc[:int(len(series) * 0.8)]
7    test_pr = pr_LMR[metric].iloc[int(len(series) * 0.8):]
8    cycle, trend = sm.tsa.filters.hpfilter(series, 1600)
9    fig, ax = plt.subplots(3,1)
10   ax[0].plot(series)
11   ax[0].set_title('Value')
12   ax[1].plot(trend)
13   ax[1].set_title('Trend')
14   ax[2].plot(cycle)
15   ax[2].set_title('Cycle')
16   plt.show()
17   print('Descomponemos la series temporal con el algoritmo STL')
18   series = pr_LMR[metric]
19   result = STL(series).fit()
20   result.plot()
21   plt.show()
```

En la línea 5 hacemos una copia de los valores de la métrica de nuestro conjunto, a la que denomminamos serie. En las líneas 6 y 7, separamos nuestros datos en dos conjuntos, uno para entrenamiento y otro para testear. El primero de ellos contiene el 80 % de los datos y el segundo el 20 % restante.

A continuación, en la línea 8 usamos el método HP para extraer la tendencia y el ciclo. A este método le pasamos la serie y un parámetro de

alisamiento. Para este parámetro hemos usado el valor 1.600, ya que es el que recomiendan por defecto. Una vez que tenemos estos valores, dibujamos la serie, su tendencia y su ciclo entre las líneas 10 y 16.

Después, usamos el algoritmo STL en la línea 19 para obtener una descomposición de nuestra serie temporal, tras esto mostramos los resultados obtenidos.

Las gráficas obtenidas para el método HP muestran los valores reales de las partículas, la tendencia y la estacionalidad. En el eje x se encuentra la parte temporal, es decir los meses, y en el eje y se encuentran los valores que toman las partículas.

Para el algoritmo STL se muestran las mismas gráficas que en el algoritmo HP y además, muestra también el residuo. Al igual que con HP, los meses se visualizan en el eje x y los valores en el y.

Podemos ver las gráficas obtenidas para la zona de Roadside en las figuras 5.2 y 5.3.
Vamos a describir las similitudes y diferencias de los resultados obtenidos con cada algoritmo y partícula.

- Óxido Nítrico y óxidos de nitrógeno: Vemos que ambas tendencias tienden al final de la serie a decrecer y son bastantes similares para ambos métodos. Sin embargo, la estacionalidad no se asemeja tanto entre ellas. Con el método HP se parece más a los valores reales, incluso muestra los valores que sobresalen al comportamiento usual en el gráfico. Con STL, obtenemos una estacionalidad más simétrica y suavizada, ajustandose un poco menos a los valores reales.
- Dióxido de nitrógeno: Las tendencias son para ambos métodos decrecientes y son bastantes similares. Para la estacionalidad ocurre como con las partículas anteriores, para el método HP es muy similar a los valores reales y para el algoritmo STL tenemos una estacionalidad más simétrica entre ella pero más diferenciada de los valores reales.
- Ozono: Para esta partícula la tendencia se comporta de forma diferente en ambas zonas. El comportamiento desde el principio hasta 2018 es bastante parecido, sin embargo a partir de esta fecha, para el método HP la tendencia es creciente y para STL es decreciente. Por otro lado, la estacionalidad es muy parecida para ambos.
- Partículas PM10 y PM2.5: La tendencia para ambos se ccomportan de forma bastante similar y las dos terminan de una forma decreciente,

aunque la pendiente para el método HP sea bastante menor. La estacionalidad difiera un poco más, ya que con STL los picos están más suavizados. En general, tienen un comportamiento parecido, pero se comporta con más suavidad para STL.

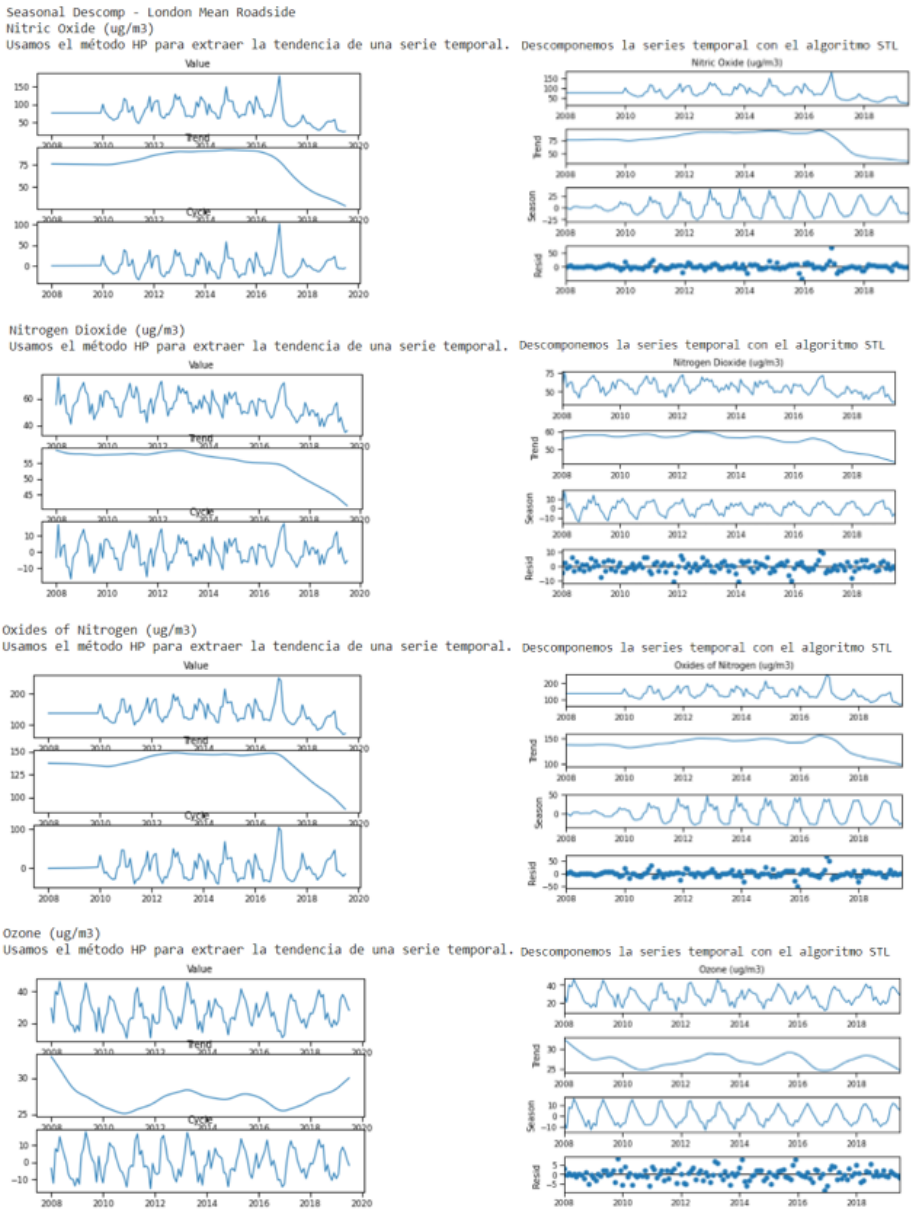

Figura 5.2: Descomposición estacional con el algoritmo HP y STL para la zona de Roadside y las partículas Óxido nítrico, Dióxido de nitrógeno, óxidos de nitrógeno y Ozono.

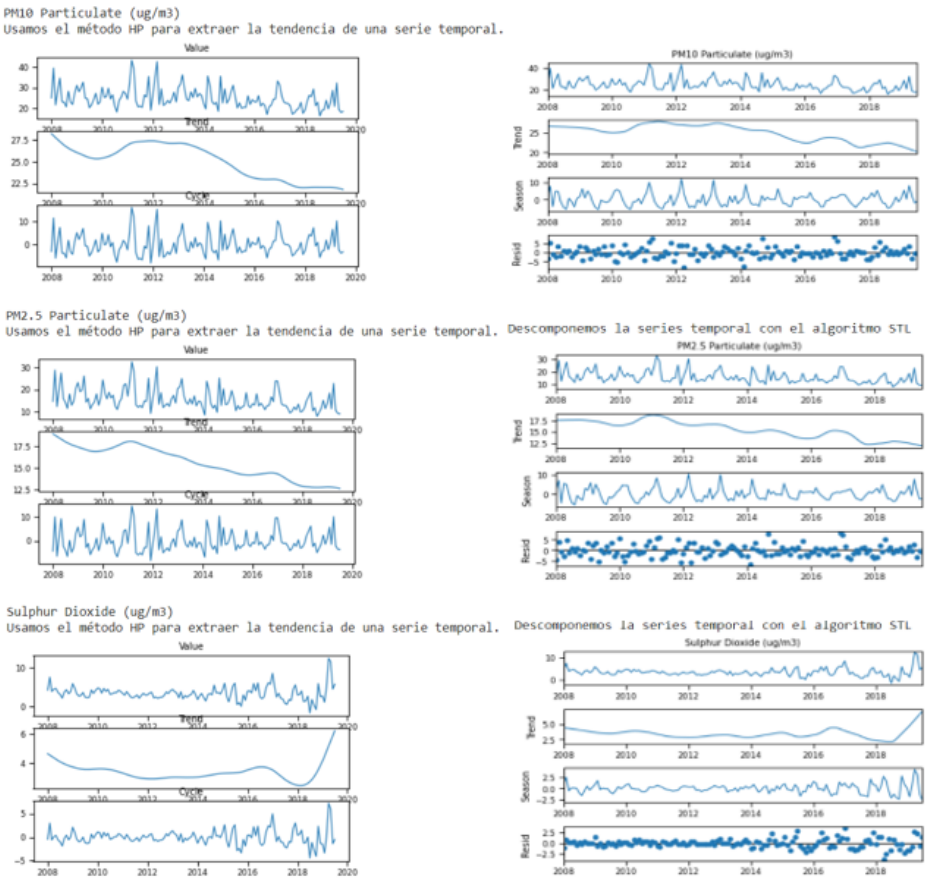

Figura 5.3: Descomposición estacional con el algoritmo HP y STL para la zona de Roadside y las partículas PM10, PM2.5 y sulfuro de dióxido.

De igual forma, las gráficas para la zona de Background las podemos ver en las figuras 5.4 y 5.5.
Las similitudes y diferencias de los resultados obtenidos con cada algoritmo y partícula son los siguientes.

- Óxido nítrico y óxidos de nitrógeno: La tendencia se comporta de forma muy similar al principio de la serie para ambos métodos, sin embargo termina siendo decreciente para HP y prácticamente nula para STL. Para la estacionalidad no se comportan de forma muy similar. Con el método HP encontramos varios picos y un comportamiento idéntico a los valores reales y con STL no esta tana centuada y sigue un patrón en su movimiento.

- Dióxido de nitrógeno: Las tendencias se comportan de forma casi identica en ambas y acaban siendo decrecientes. La estacionalidad es tam-

bién bastante similar, aunque se comporta más suave para el segundo.

- Ozono: Las tendencias se comportan de forma muy similar a lo largo de la serie, sin embargo al final el método HP considera la tendencia como creciente y el algoritmo STL como decreciente. La estacionalidad es muy parecida para ambas, aunque más acentuada con HP.

- Partícula PM10: Las tendencias se comportan de forma parecida pero difieren al final, para HP se queda más neutral y para STL termina siendo decreciente. La estacionalidad es muy similar esntre ambasm siendo bastante más suave con STL.

- Partícula PM2.5: Las tendencias son muy similares a lo largo de la serie, pero difieren al final de esta. Para HP termina siendo creciente y para STL casi nula. El comportamiento de la estacionalidad difiere poco entre ambas.

- Sulfuro de dióxido: Las tendencias se comportan de forma casi idéntica para las dos zonas, siendo al final nula para ambas. La estacionalidad se asemeja un poco en ambas, siendo HP mucho más preciso con respecto a los valores reales que STL.

Una vez que tenemos nuestra serie descompuesta, vamos a probar a hacer un modelo de persistencia. Este modelo consiste en tomar el último valor observado como predicción para el siguiente momento. Aunque este método no sea muy eficaz, nos sirve para empezar la predicción.

```
print('Seasonal Descomp - London Mean Roadside')
for name, metric in zip(names, metrics):
  print(name)
  series = pr_LMR[metric]
  train_pr = pr_LMR[metric].iloc[:int(len(series) * 0.8)]
  test_pr = pr_LMR[metric].iloc[int(len(series) * 0.8):]
  print('Prediccion')
  predictions = series.shift(1).dropna()
  test_score = np.sqrt(mean_squared_error(series[int(len(series) *
      0.8)+1:], predictions.iloc[int(len(series) * 0.8):]))
  print('Test RMSE: %.5f' % test_score)
  value = plt.plot(series.iloc[int(len(series) * 0.8)+1:], label='
      Value')
  plt.plot(predictions[int(len(series) * 0.8):], color='red', label='
      Prediction')
  plt.legend()
  plt.show()
  pred = pd.concat([series.iloc[-int(len(series) * 0.2):].pct_change()
      , predictions.iloc[-int(len(series) * 0.2):].pct_change()], axis
      =1)
  pred.columns = ['Value', 'preds']
  pred['hit'] = np.where(np.sign(pred['Value']) == np.sign(pred['preds
      ']), 1, 0)
  print(f"Hit rate: {round((pred['hit'].sum() / pred['hit'].count()) *
       100,2)}%")
```

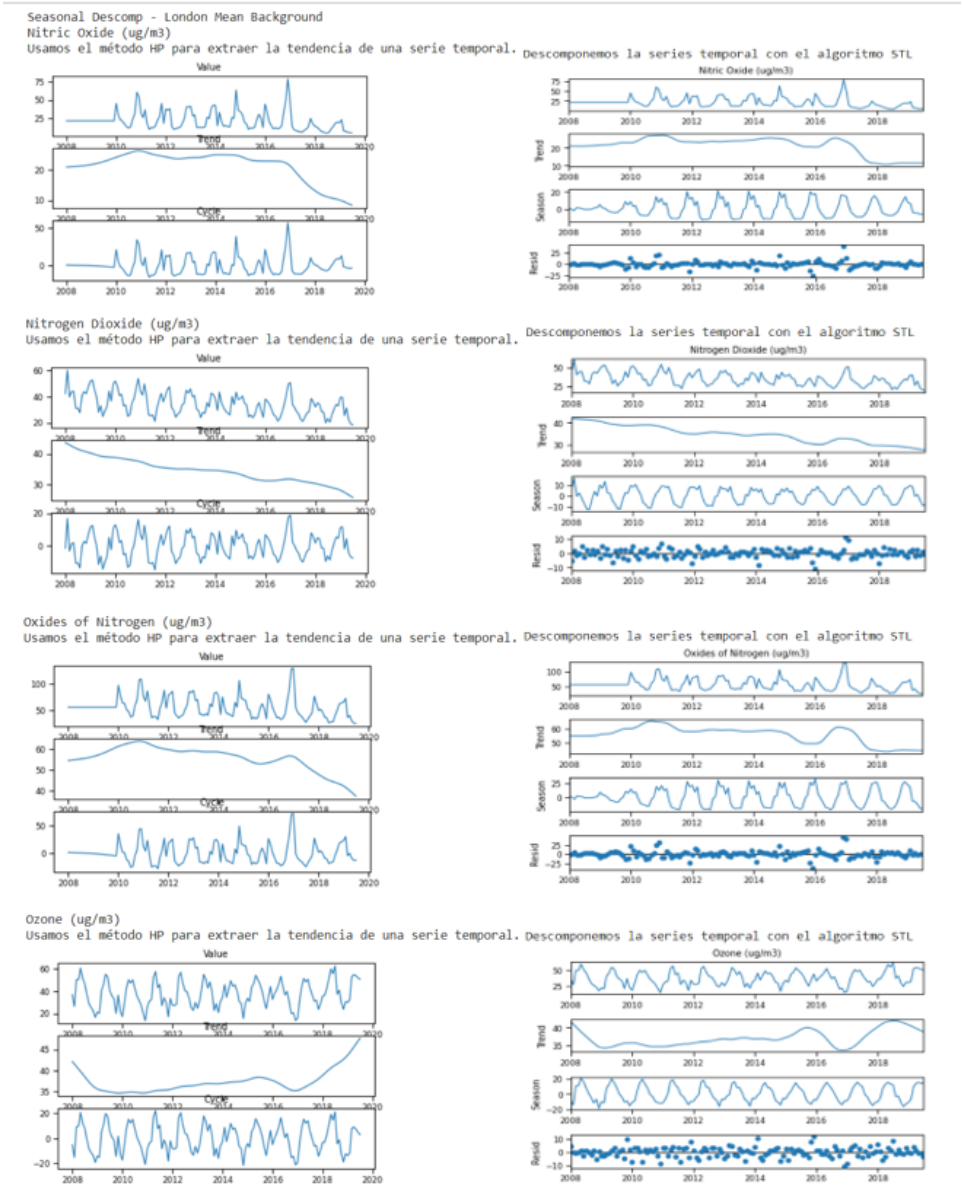

Figura 5.4: Descomposición estacional con el algoritmo HP y STL para la zona de Background y las partículas Óxido nítrico, Dióxido de nitrógeno, óxidos de nitrógeno y Ozono.

En la línea 8 realizamos la predicción usando la función `shift`. Le pasamos como variable el valor 1 para indicar que se va a desplazar el valor actual al siguiente.
En la línea 9 calculamos el error cuadrático medio que sse ha obtenido para esta predicción, e imprimirlo en la línea 10.

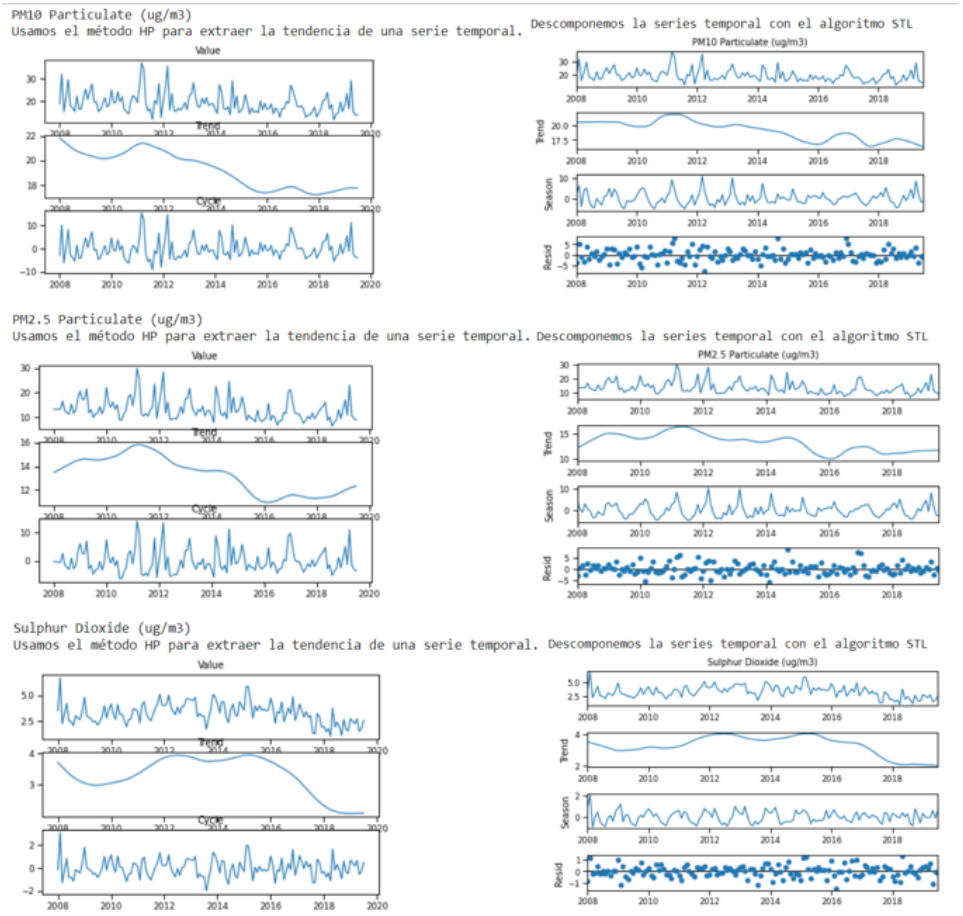

Figura 5.5: Descomposición estacional con el algoritmo HP y STL para la zona de Roadside y las partículas PM10, PM2.5 y sulfuro de dióxido.

Con las líneas 11, 12, 13 y 14 mostramos un gráfico con los valores reales y los valores predichos. Finalmente, entre las líneas 15 y 16, calcula la tasa de aciertos de la dirección del próximo valor y así mostrarla en la línea 17.

Para la zona de Roadside podemos ver los valores reales y la predicción de estos en la figura 5.6 y para la otra zona podemos ver los valores reales y la predicción de estos en la figura 5.7 .

Como vemos, la predicción obtenida es una regresión de los valores con $t = 1$. Las gráficas obtenidas para algunas partículas, en algunos periodos de tiempo son aceptables, por ejemplo para el óxido nítrico desde enero de 2018 hasta julio del mismo año.
Observamos que cuando hay algunas subidas o bajadas y las pendientes no son muy grandes, antes de llegar a los máximos y mínimos, el valor real y el

predicho son muy parecidos y tienen un pequeño margen de error.
Por el contrario, cuando hay un máximo y un mínimo seguidos, o una diferencia grande entre los valores para los momentos t y $t+1$, la diferencia entre el valor real y el predicho es bastante grande.

Este método no nos serviría si pretendemos realizar una predicción para los próximos años, ya que no tendríamos los valores reales y se quedaría una línea recta con el último valor.

A pesar de eso, el error que obtenemos para estos datos no es malo, ya que es un error muy pequeño. Esto es debido a que la diferencia entre los valores no es demasiado grande, y depende mucho del rango de valores de cada partícula.
Podemos ver la tabla con el error cuadrático medio y la tasa de acierto de la dirección para la zona de Roadside en la tabla 5.1 y para la zona de Background en la tabla 5.2.

	MRSE	Hit Rate
Óxido nítrico	8,01	62,96 %
Dióxido de nitrógeno	5,35	40,74 %
Óxidos de nitrógeno	16,51	62,96 %
Ozono	5,56	37,04 %
Partículas PM10	4,93	40,74 %
Partículas PM2.5	4,45	40,74 %
Dióxido de azufre	3,02	48,15 %

Tabla 5.1: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para el modelo de persistencia en la zona de Roadside.

Una vez terminado con estre primer modelo, vamos a usar un modelo autoregresivo. Como hemos visto en el capítulo 3, se trata de un modelo de regresión lineal que usan valores anteriores como valores futuros.

```
print('Seasonal Descomp - London Mean Roadside')
for name, metric in zip(names, metrics):
  print(name)
  series = pr_LMR[metric]
  train_pr = series.iloc[:int(len(series) * 0.8)].to_list()
  test_pr = series.iloc[int(len(series) * 0.8):]
  predictions = []

  for i in range(len(test_pr)):
    lags=(ar_select_order(train_pr,maxlag=10))
    model = AutoReg(train_pr, lags.ar_lags)
    model_fit = model.fit()
    pred = model_fit.predict(start=len(train_pr), end=len(train_pr),
```

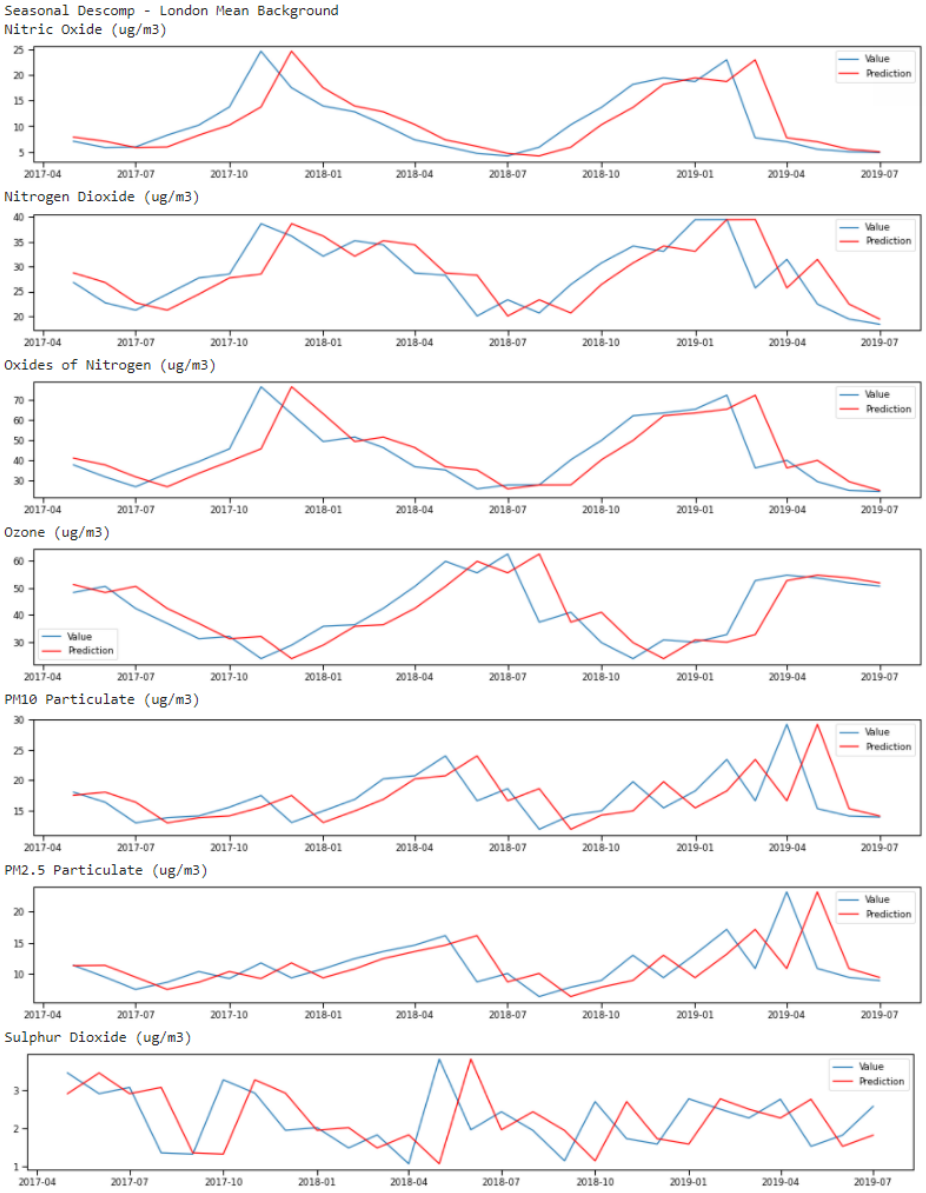

Figura 5.6: Resultados de la predicción usando persistencia en la zona de Roadside.

```
        dynamic=False)
    predictions.append(pred[0])
    train_pr.append(test_pr[i])

predictions = pd.Series(predictions, index=test_pr.index)

test_score = np.sqrt(mean_squared_error(test_pr, predictions))
print('Test MSE: %.5f' % test_score)
```

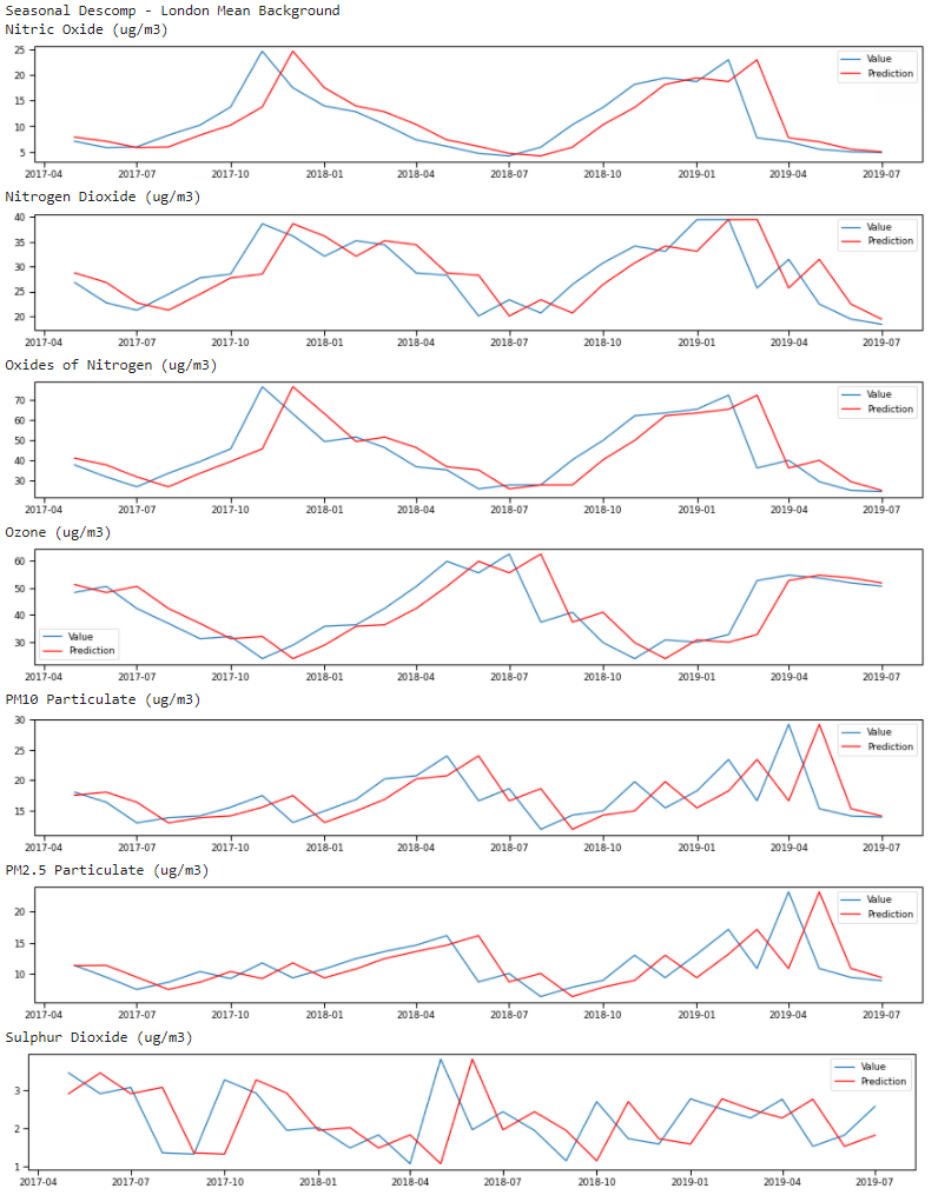

Figura 5.7: Resultados de la predicción usando persistencia en la zona de Background.

```
21    # plot results
22    plt.plot(test_pr.iloc[-int(len(series) * 0.8):], label='Value')
23    plt.plot(predictions.iloc[-int(len(series) * 0.8):], color='red',
          label='Prediction')
24    plt.legend()
25    plt.show()
26
27    value_pred = pd.concat([test_pr.pct_change(), predictions.pct_change
```

	MRSE	Hit Rate
Óxido nítrico	4,46	74,07 %
Dióxido de nitrógeno	5,16	51,85 %
Óxidos de nitrógeno	11,61	66,67 %
Ozono	8,22	44,44 %
Partículas PM10	4,92	48,15 %
Partículas PM2.5	4,33	40,74 %
Dióxido de azufre	1,03	22,22 %

Tabla 5.2: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para el modelo de persistencia en la zona de Background.

```
        ()], axis=1)
28  value_pred.dropna(inplace=True)
29  value_pred.columns = ['Value', 'preds']
30  value_pred['hit'] = np.where(np.sign(value_pred['Value']) == np.sign
        (value_pred['preds']), 1, 0)
31  print(f"Hit rate: {round((value_pred['hit'].sum() / value_pred['hit
        '].count()) * 100,2)}%")
```

Como podemos ver en las líneas 5 y 6, hemos vuelto a separar al igual que antes nuestra serie temporal en dos conjuntos, uno con el 80 % de datos para entrenar y otro con el 20 % para testear.

Vamos a guardar en una lista los valores que vamos a predecir. Para realizar la predicción vamos a usar un bucle que se ejecutará tantas veces como la longitud de nuestros datos de prueba. Cada vez que entramos en el bucle vamos a usar la función `AutoReg`, que entrena el modelo de regresión lineal. En la línea 10 usamos la función `ar_select_order` para que seleccione automáticamente el retardo adecuado que usamos en la línea 11 a la función `AutoReg`. Tras esto, en la línea 12 entrenamos el modelo y en la línea 13 predecimos el siguiente valor y lo añadimos en la línea 14 en la lista.

Por último, en la línea 15 añadimos al final de nuestro conjunto de entrenamiento el valor de los datos de prueba inicial. De esta forma, el modelo se irá entrenando y ajustando con el último valor que hemos añadido. Volvemos a calcular el error cuadrático medio en la línea 19 y la tasa de aciertos de la dirección entre las líneas 27 y 30. Además, entre las líneas 22 y 25 dibujamos los valores reales y la predicción realizada.

Este modelo imita al paso del tiempo en la vida real, ya que no tenemos un día adicional de datos disponibles en comparación con el día anterior.

Si observamos los resultados que obtenemos en la figura 5.8 para la zona de Roadside vemos que los valores obtenidos difieren bastante de la realidad, por lo que este método tampoco es demasiado bueno para predecir y estimar los valores futuros con nuestro conjunto de datos.
Vamos a ver detalladamente como se comporta cada partícula. La tabla 5.3 contiene los errores de predicción.

- Óxido nítrico, dióxido de nitrógeno y óxidos de nitrógeno: La gran mayoria de los valores predichos superan a los reales. Como hemos visto al principio de la sección con los métodos HP y STL, la tendencia para estas partículas es decreciente. Por este motivo al usar los valores anteriores como base para la predicción, los valores predichos son mayores a los reales.

- Ozono: Para esta partícula los valores de predicción oscilan alrededor de los valores reales, siendo en derterminados momentos mayores y en otros menores.

- Partículas PM10 y partículas PM2.5: La mayoría de los valores predichos son mayores a los reales. La predicción que se hace es similar a una línea recta, es decir, tiene pocos picos.

- Sulfuro de azufre: Al igual que con las partículas finas, la predicción es similar a una recta con poca diferencia entre máximos y mínimos. Sin embargo, los valores predichos oscilan alrededor de los reales, es decir, no se quedan por arriba ni por debajo.

Los resultados que obtenemos para la zona de Background en la figura 5.9 nos muestran también unos valores predichos que no se ajustan demasiado bien a la realidad.

- Óxido nítrico y óxidos de nitrógeno: Al igual que ocurre con estas partículas en la otra zona, la predicción de los valores es en su mayoría superior a la real. Sin embargo para el óxido nítrico, los valores entre septiembre de 2018 y febrero de 2019 son muy buenos y se ajustan realmente bien.

- Dióxido de nitrógeno: En esta zona, los valores estimados para esta partícula se encuentran iterando entre sobre el valor real.

- Ozono: La mayoría de los valores de esta zona se encuentran por debajo de los valores reales, pero existen bastante momentos en que los valores son muy parecidos.

- Partículas PM10 y partículas PM2.5: Ocurren al igual que el zona de Roadside. La mayoría de los valores predichos son mayores a los reales y la línea de predicción parecida a una línea recta.

- Sulfuro de azufre: La gran mayoría de los valores predichos son mayores a los reales.

Podemos observar el error cuadrático medio y el porcentaje de acierto sobre la dirección en la tabla 5.4

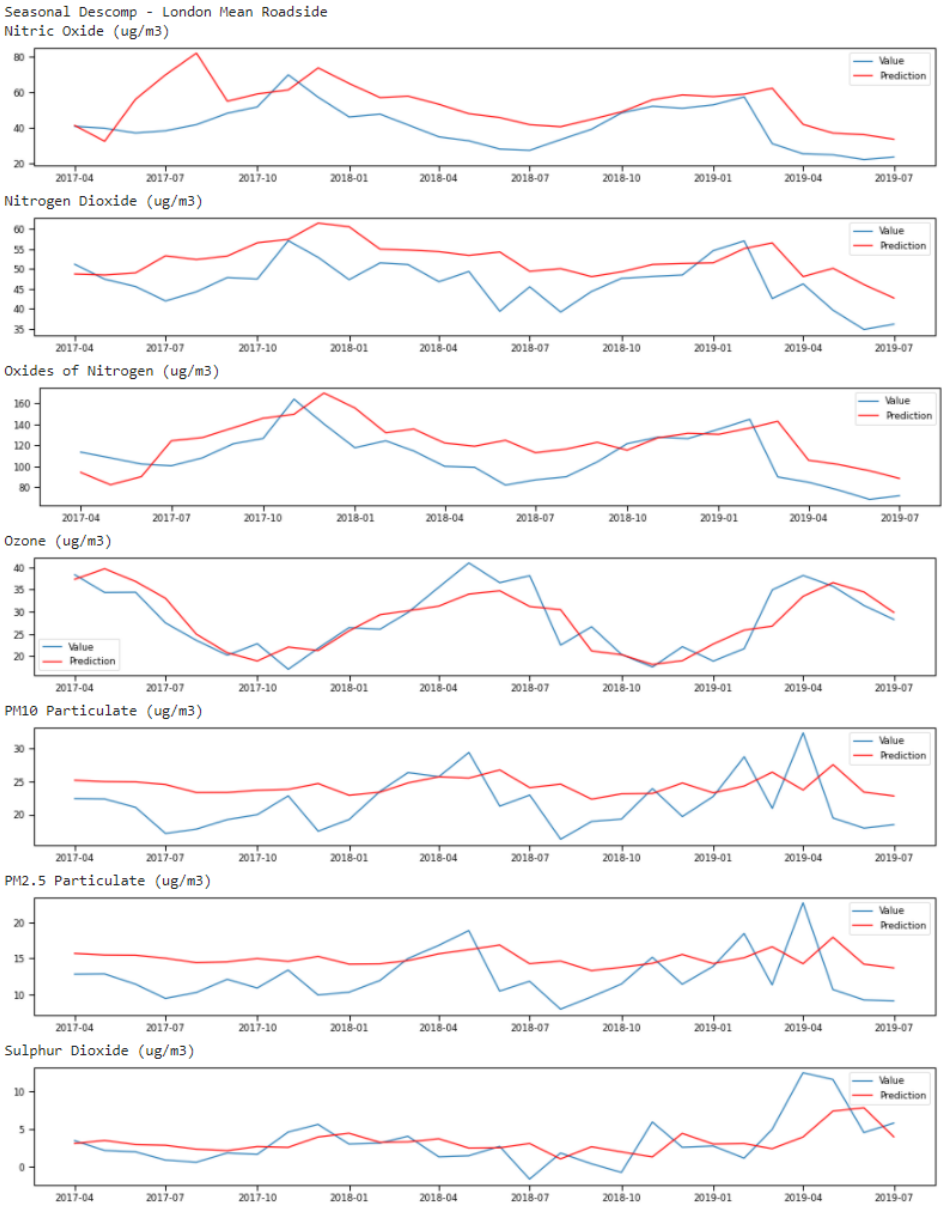

Figura 5.8: Resultados de la predicción usando un modelo autorregresivo en la zona de Roadside.

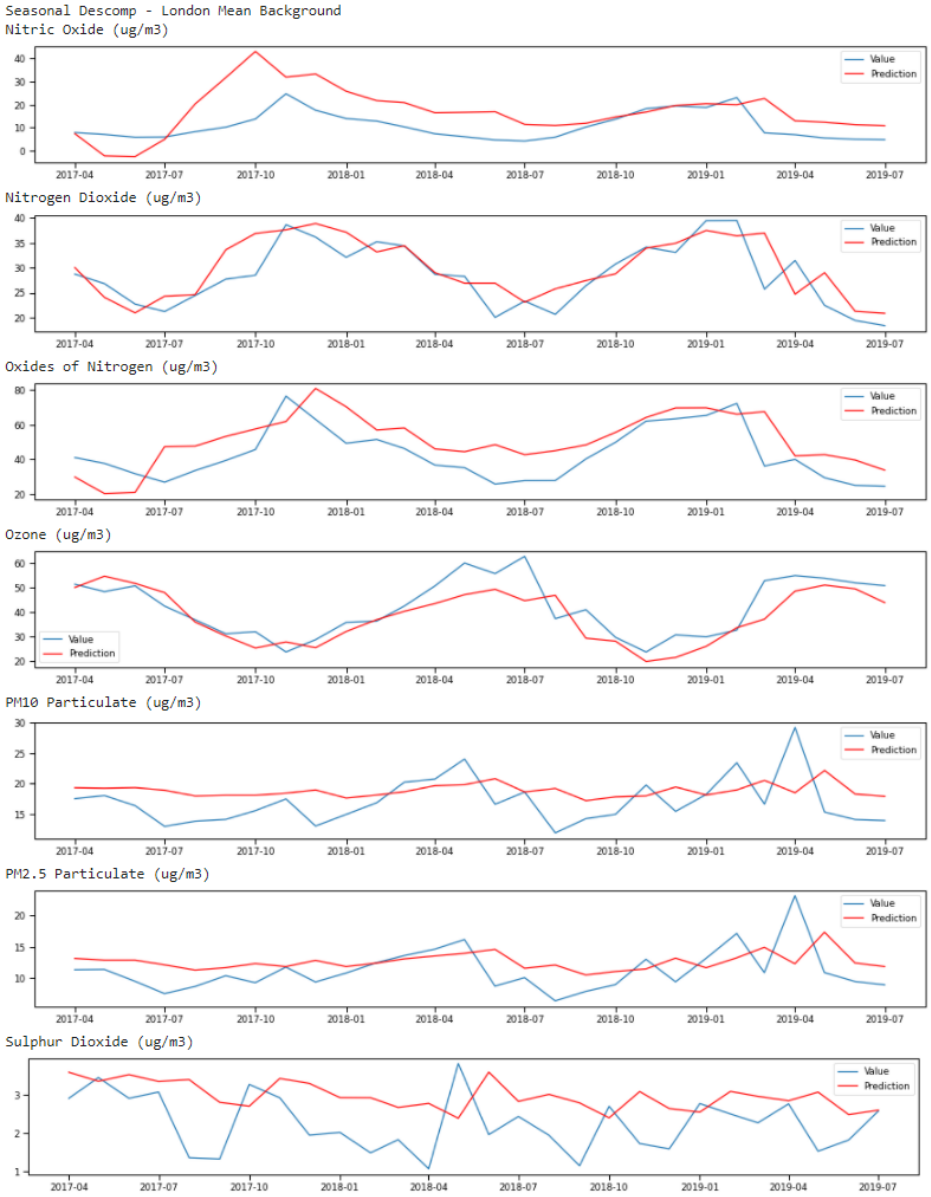

Figura 5.9: Resultados de la predicción usando un modelo autorregresivo en la zona de Background.

Al principio de este capitulo hemos usado el filtro HP para ver la tendencia y la estacionalidad. A continuación lo vamos a utilizar, utilizando los métodos de descomposición para que nos ayuden a mejorar nuestra predicción.

	RMSE	Hit Rate
Óxido nítrico	15,99	62,96 %
Dióxido de nitrógeno	7,40	44,44 %
Óxidos de nitrógeno	23,27	55,56 %
Ozono	4,10	59,26 %
Partículas PM10	4,67	44,44 %
Partículas PM2.5	4,26	44,44 %
Dióxido de azufre	2,69	37,04 %

Tabla 5.3: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para el modelo autorregresivo en la zona de Roadside.

	RMSE	Hit Rate
Óxido nítrico	10,45	62,96 %
Dióxido de nitrógeno	4,17	59,26 %
Óxidos de nitrógeno	14,07	59,26 %
Ozono	7,13	59,26 %
Partículas PM10	4,18	44,44 %
Partículas PM2.5	3,68	40,74 %
Dióxido de azufre	1,06	25,93 %

Tabla 5.4: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para el modelo autorregresivo en la zona de Background.

El siguiente modelo que vamos a realizar obtiene de la serie temporal la tendencia y la estacionalidad para predecirlas por separado y después recomponer la serie usando estos valores predichos.

```
1  print('Seasonal Descomp - London Mean Roadside')
2  for name, metric in zip(names, metrics):
3    print(name)
4    series = pr_LMR[metric]
5    cycle, trend = sm.tsa.filters.hpfilter(series, 1600)
6    component_dict = {'cycle': cycle, 'trend': trend}
7    prediction_results = []
8    for component in ['trend', 'cycle']:
9        train_pr = component_dict[component].iloc[:int(len(series) *
             0.8)].to_list()
10       test_pr = component_dict[component].iloc[int(len(series) * 0.8)
             :]
11       predictions = []
12       for i in range(len(test_pr)):
13           lags=(ar_select_order(train_pr,maxlag=10))
14           model = AutoReg(train_pr, lags.ar_lags)
15           model_fit = model.fit()
```

```
16          pred = model_fit.predict(start=len(train_pr), end=len(
                train_pr), dynamic=False)
17          predictions.append(pred[0])
18          train_pr.append(test_pr[i])
19
20      predictions = pd.Series(predictions, index=test_pr.index, name=
            component)
21      prediction_results.append(predictions)
22      test_score = np.sqrt(mean_squared_error(test_pr, predictions))
23      print(f'Test for {component} MSE: {test_score}')
24      # plot results
25      plt.figure(figsize=(15, 2))
26      plt.plot(test_pr.iloc[:], label='Observed '+component)
27      plt.plot(predictions.iloc[:], color='red', label='Predicted '+
            component)
28      plt.legend()
29      plt.show()
30
31  recomposed_preds = pd.concat(prediction_results,axis=1).sum(axis=1)
32  recomposed_preds.name = 'recomposed_preds'
33  print("Recomposed")
34  plt.figure(figsize=(15, 2))
35  plt.plot(series.iloc[int(len(series) * 0.8):], label='Observed')
36  plt.plot(recomposed_preds, color='red', label='Predicted')
37  plt.legend()
38  plt.show()
39  test_score = np.sqrt(mean_squared_error(series.iloc[int(len(series)
        * 0.8):], recomposed_preds))
40  #print(f'RMSE: {test_score}')
41
42  value_pred = pd.concat([series.iloc[-int(len(series) * 0.2):].
        pct_change(),recomposed_preds.iloc[-int(len(series) * 0.2):].
        pct_change()], axis=1)
43  value_pred.dropna(inplace=True)
44  value_pred.columns = ['Value', 'preds']
45  value_pred['hit'] = np.where(np.sign(value_pred['Value']) == np.sign
        (value_pred['preds']), 1, 0)
```

Lo primero que hacemos en la línea 5 es obtener el ciclo y la tendencia de la serie, y las almacenamos en la línea 6. A continuación, hacemos un bucle para cada par ciclo-tendencia. En las líneas 9 y 10 dividimos el conjunto de datos en dos, uno de entrenamiento y otro de prueba, con el 80 % y 20 % de los datos respectivamente. Entre las líneas 11 y 27, hacemos exactamente lo mismo que en el modelo anterior.

En la línea 29 recomponemos nuestros datos de forma aditiva, dibujando entre las líneas 31 y 34 la gráfica de la función real y la de los valores predichos. Finalmente, volvemos a calcular el error cuadrático medio y la tasa de acierto de la dirección para los valores predichos con respecto a los reales.

Las gráficas obtenidas para la zona de roadside las podemos observar en las figuras 5.10, 5.11 y 5.12. Podemos ver el error obtenido en la tabla 5.5. 50

Si observamos las predicciones hechas para la tendencia observamos una única línea roja de predicción y no vemos la azul de los valores reales. Esto

ocurre porque la precisión en la tendencia es tan buena, que los valores se sobreponen y parece una única línea.

Vamos a ver que ocurre con la estacionalidad y con la predicción completa. Antes de verlo, tenemos que comentar que ambas se comportan de forma practicamente identica, podemos comprobar que no son iguales ya que su error difiere un poco entre ambas, aunque apenas unas centesimas.

- Óxido nítrico y óxidos de nitrógeno: Al principio de la predicción, desde abril de 2017 hasta febrero de 2018, se diferencian más los valores reales y los predichos. Sin embargo, después la predicción es bastante mejor y se ajusta más a los valores buscados.

- Dióxido de nitrógeno: Esta predicción no se ajusta tan bien como la anterior, aunque consigue llevar de forma parecida las curvas de la gráfica, no consigue predecir los picos producidos por cambios en valores seguidos.

- Ozono: Esta predicción se ajusta al movimiento de la curvatura de los valores reales pero al igual que la anterior no consigue reflejar sus máximos y mínimos.

- Partículas PM10 y PM2.5: Los valores predichos se encuentran en el centro de la gráfica. Sus valores no se ajustan demasiado bien a los movimientos que realiza, aunque para ciertos momentos concretos consiga predecir el valor con un margen de error muy pequeño.

- Dióxido de azufre: Los valores reales y su predicción se encuentran en un rango pequeño de valores. Los datos predichos se ajustan hasta julio de 2018 bastante mejor a los valores reales, que a partir de esta fecha.

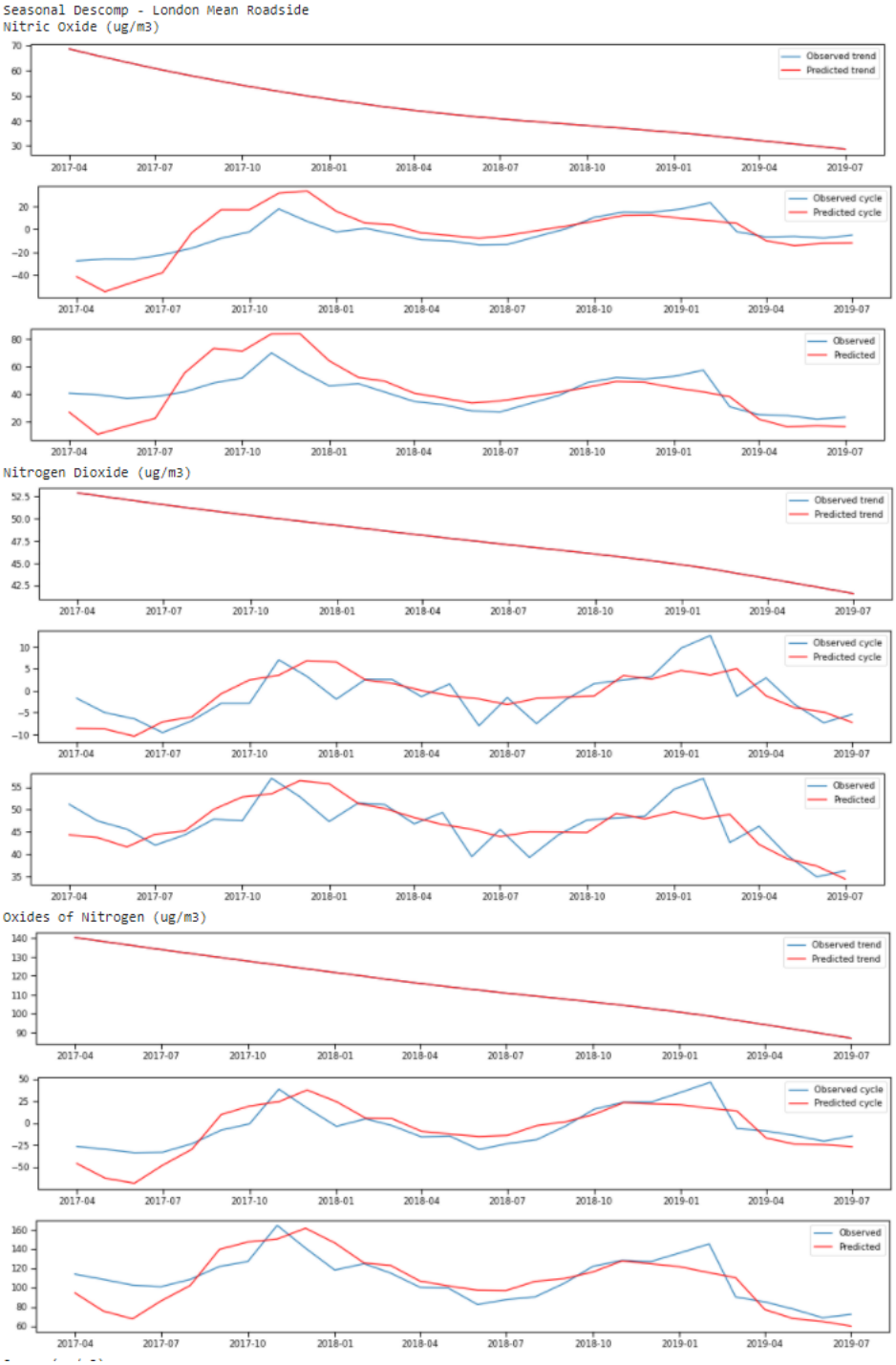

Figura 5.10: Tendencia, estacionalidad y gráfica completa usando la descomposición estacional con el algoritmo HP para la zona de Roadside y las partículas de óxido nítrico, dióxido de nitrógeno y óxidos de nitrógeno.

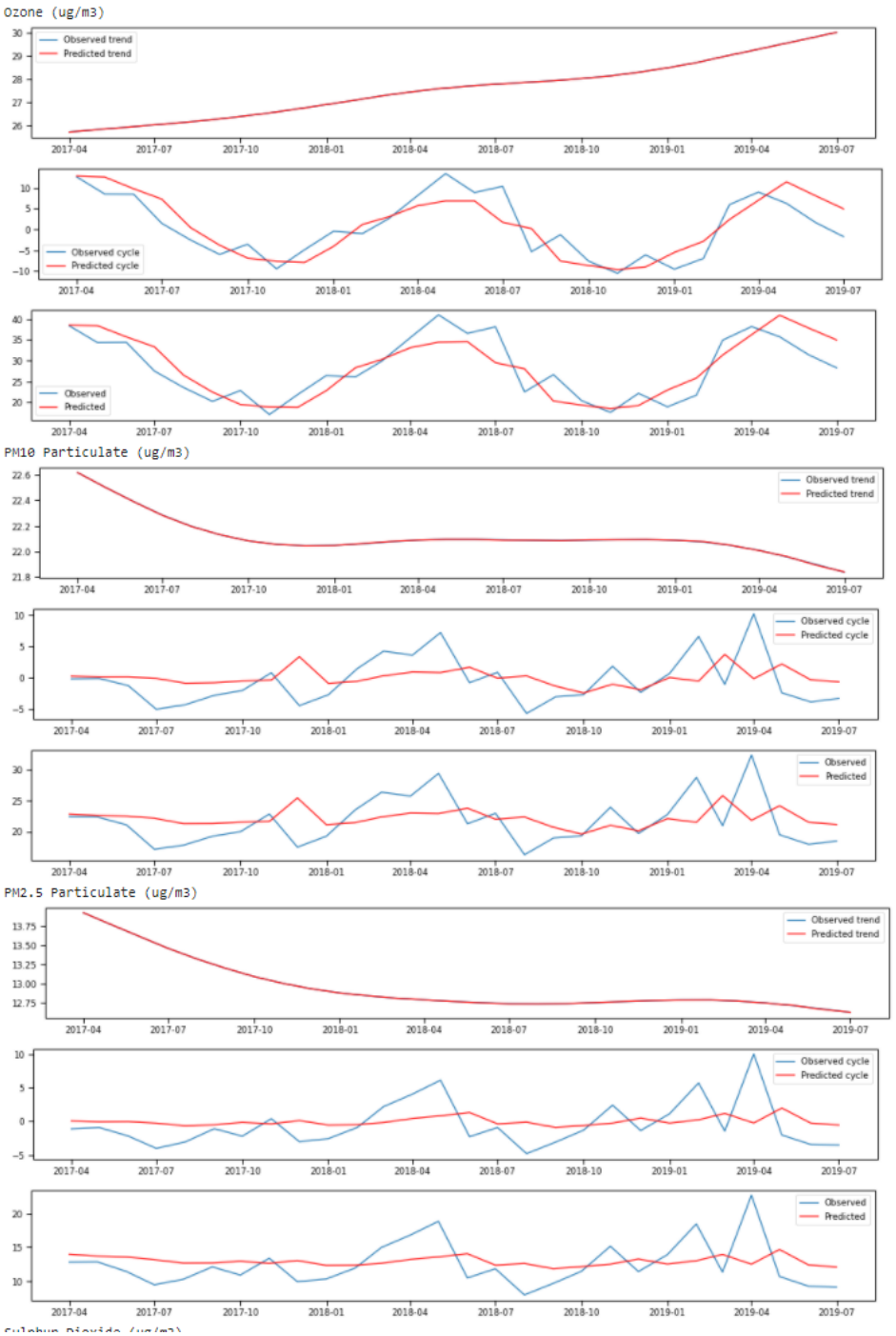

Figura 5.11: Tendencia, estacionalidad y gráfica completa usando la descomposición estacional con el algoritmo HP para la zona de Roadside y las partículas de ozono. partículas PM10 y PM 2.5.

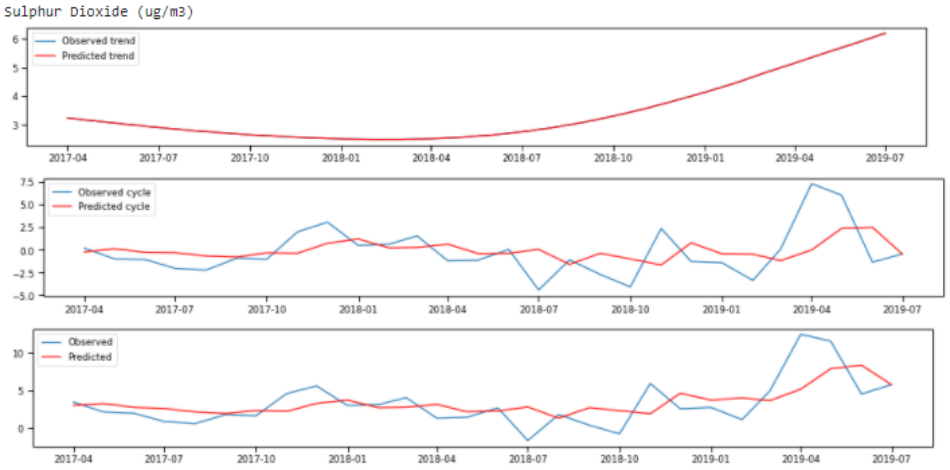

Figura 5.12: Tendencia, estacionalidad y gráfica completa usando la descomposición estacional con el algoritmo HP para la zona de Roadside y la partícula de sulfuro de dióxido.

	$RMSE_{Tendencia}$	$RMSE_{Estacionalidad}$	RMSE	Tasa Acierto
Óxido nítrico	0,01	13,15	13,15	65,38 %
Dióxido de nitrógeno	0,00	4,15	4,15	46,15 %
Óxidos de nitrógeno	0,01	16,46	16,46	73,08 %
Ozono	0,00	4,13	4,13	65,38 %
Partículas PM10	0,00	4,06	4,06	42,31 %
Partículas PM2.5	0,00	3,41	3,41	46,15 %
Dióxido de azufre	0,00	2,48	2,48	38,46 %

Tabla 5.5: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para la descomposición estacional con el algoritmo HP en la zona de Roadside.

Para la zona de Background vemos en las figuras 5.14, 5.15 y 5.13 las predicciones obtenidas con este método. Podemos ver el error obtenido en la tabla 5.6.
Al igual que hemos visto para Roadside, la tendencia la observamos como una única línea de predicción, esto es también debido a la precisión tan buena que hay en la tendencia.

No ocurre lo mismo con la predicción del ciclo y de la serie, ya que dependiendo de la partícula esta se ajusta más o menos. De igual forma, tanto la estacionalidad como la serie se comportan de forma casi identica, por lo

que nos referiremos a las dos juntas.

- Óxido nítrico y óxidos de nitrógeno: Se comportan igual en ambas zonas. Desde el principio de la predicción hasta febrero/marzo de 2018 se diferencian más los valores reales y los predichos. Sin embargo, después la predicción es bastante mejor y se ajusta más a los valores buscados.

- Dióxido de nitrógeno: En esta zona se ajusta un poco mejor que en la anterior, pero tampoco es buena. Consigue llevar de forma parecida el movimiento producido por los valores reales, no consigue predecir los valores máximos y mínimos que se salen de la sintonía.

- Ozono: Esta predicción se ajusta al movimiento de la curvatura de los valores reales pero al igual que la anterior no consigue reflejar sus máximos y mínimos.

- Partículas PM10 y PM2.5: Aunque los valores sean cercanos, los valores predichos se encuentran en el centro de la gráfica sin ajustarse demasiado bien a as subidas y bajadas de los valores reales.

- Dióxido de azufre: Los valores reales y su predicción difieren poco en los valores ya que su rango es muy pequeño. Sin embargo, la predicción es casi una recta, mientras que los valores reales tienen mucha más variación.

Figura 5.13: Tendencia, estacionalidad y gráfica completa usando la descomposición estacional con el algoritmo HP para la zona de Background y la partícula de sulfuro de dióxido

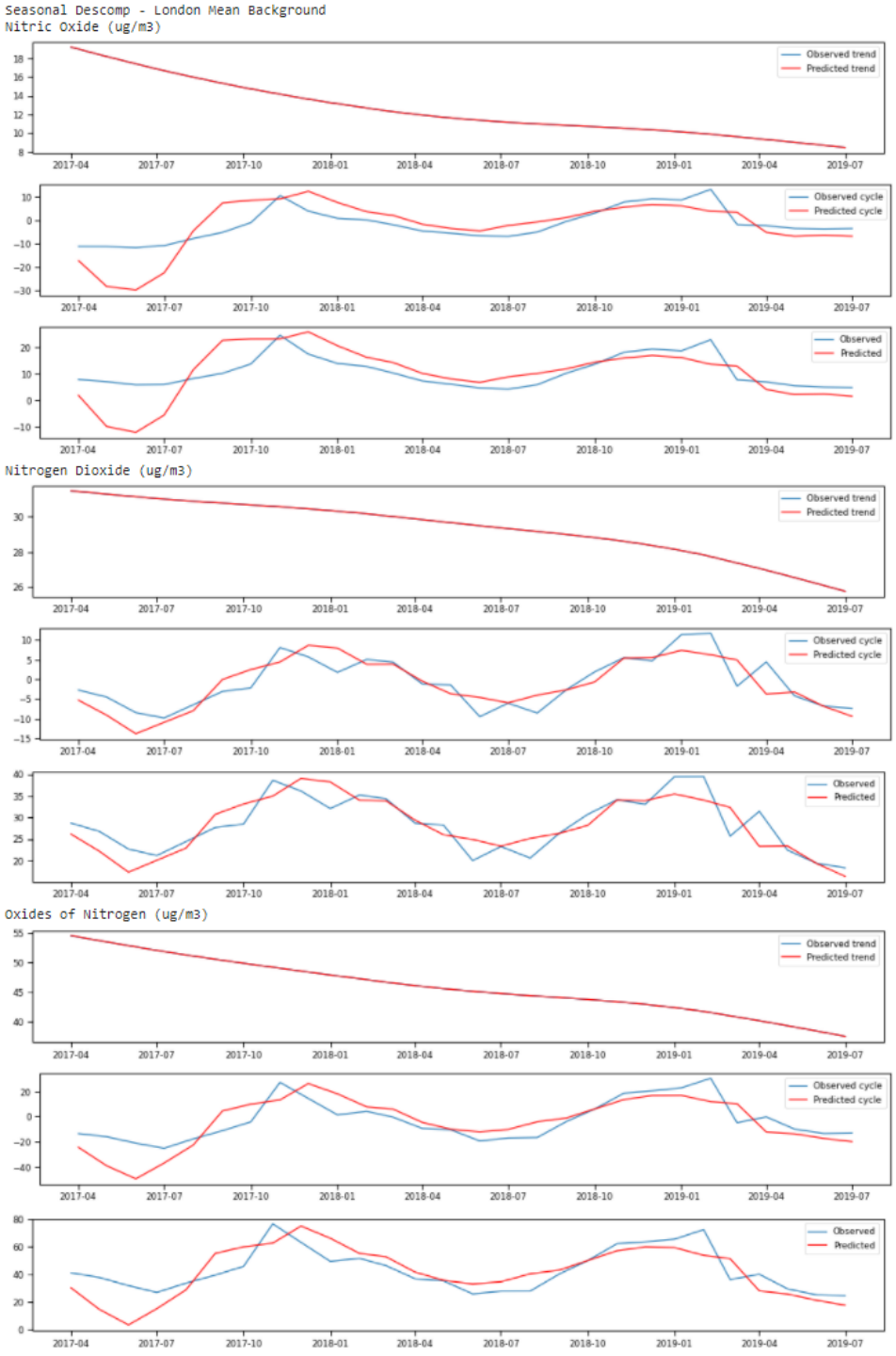

Figura 5.14: Tendencia, estacionalidad y gráfica completa usando la descomposición estacional con el algoritmo HP para la zona de Background y las partículas de óxido nítrico, dióxido de nitrógeno y óxidos de nitrógeno.

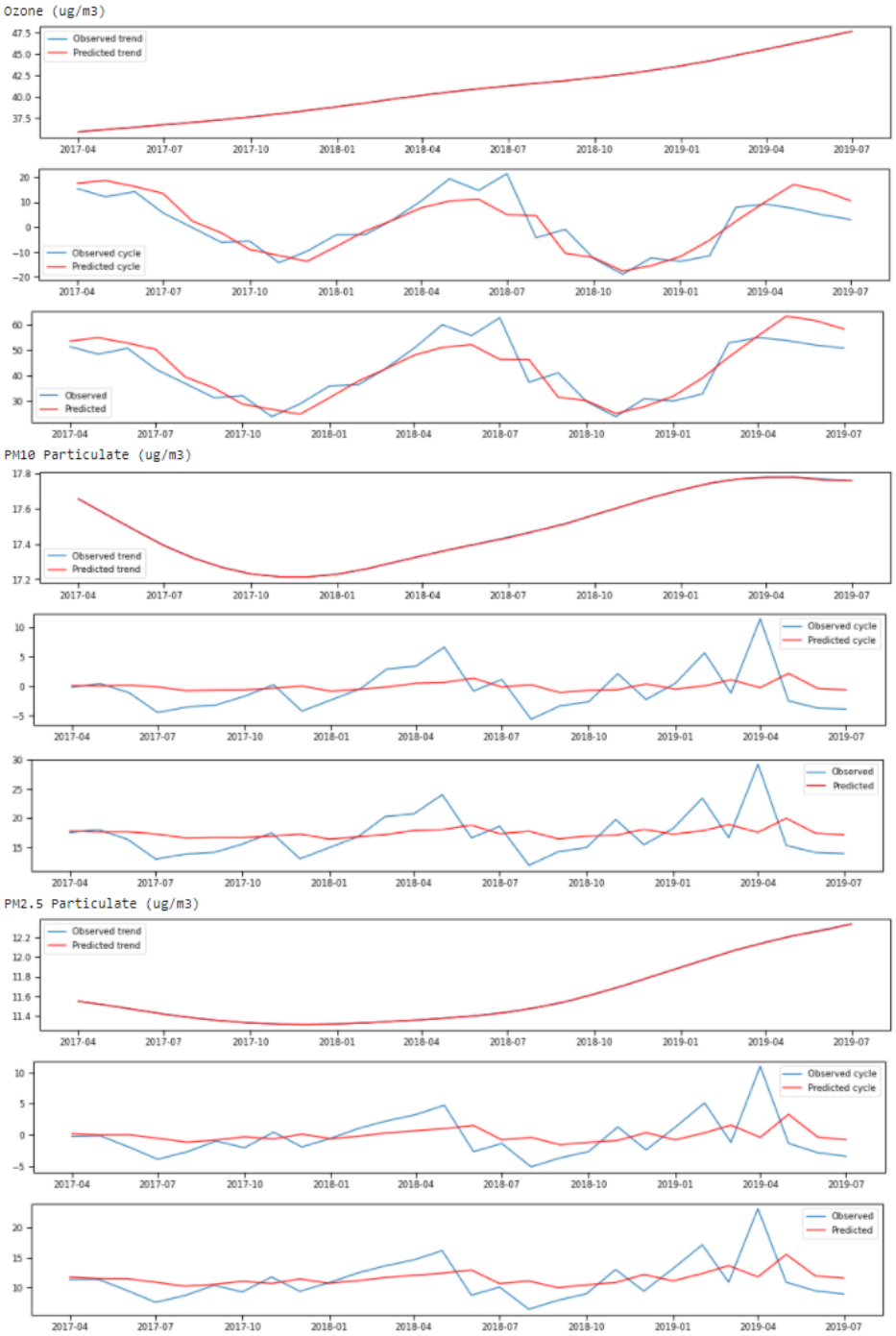

Figura 5.15: Tendencia, estacionalidad y gráfica completa usando la descomposición estacional con el algoritmo HP para la zona de Background y las partículas de ozono. partículas PM10 y PM 2.5.

	$RMSE_{Tendencia}$	$RMSE_{Estacionalidad}$	RMSE	Tasa Acierto
Óxido nítrico	0,00	7,12	7,12	73,08 %
Dióxido de nitrógeno	0,00	3,63	3,63	69,23 %
Óxidos de nitrógeno	0,01	11,81	11,81	76,92 %
Ozono	0,00	6,12	6,12	69,23 %
Partículas PM10	0,00	3,72	3,72	46,15 %
Partículas PM2.5	0,00	3,32	3,32	46,15 %
Dióxido de azufre	0,00	0,71	0,71	30,77 %

Tabla 5.6: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para la descomposición estacional con el algoritmo HP en la zona de Background.

Aunque hayamos visto algo de mejora en nuestro método, pero el filtro HP utiliza para tanto valores anteriores como futuros para realizar la predicción del valor actual, por lo que este método nos "engaña un poco" y no sería tan preciso si realizamos una predicción a futuros años.

Por este motivo, hemos visto también el método STL ya que de esta forma, no utilizamos valores futuros.

```
print('Seasonal Descomp - London Mean Roadside')
for name, metric in zip(names, metrics):
  print(name)
  series = pr_LMR[metric]
  descomposition = STL(series).fit()
  component_dict = {'seasonal': descomposition.seasonal, 'trend':
      descomposition.trend, 'residual': descomposition.resid}
  prediction_results = []
  for component in ['seasonal', 'trend', 'residual']:
      train_pr = component_dict[component].iloc[:int(len(series) *
          0.8)].to_list()
      test_pr = component_dict[component].iloc[int(len(series) * 0.8)
          :]
      predictions = []
      for i in range(len(test_pr)):
          lags=(ar_select_order(train_pr,maxlag=10))
          model = AutoReg(train_pr, lags.ar_lags)
          model_fit = model.fit()
          pred = model_fit.predict(start=len(train_pr), end=len(
              train_pr), dynamic=False)
          predictions.append(pred[0])
          train_pr.append(test_pr[i])

      predictions = pd.Series(predictions, index=test_pr.index, name=
          component)
      prediction_results.append(predictions)
      test_score = np.sqrt(mean_squared_error(test_pr, predictions))
      #print(f'Test for {component} MSE: {test_score}')
```

```
24        # plot results
25        plt.figure(figsize=(15, 2))
26        plt.plot(test_pr.iloc[:], label='Observed '+component)
27        plt.plot(predictions.iloc[:], color='red', label='Predicted '+
              component)
28        plt.legend()
29        plt.show()
30
31    recomposed_preds = pd.concat(prediction_results,axis=1).sum(axis=1)
32    recomposed_preds.name = 'recomposed_preds'
33    #print("Recomposed")
34    plt.figure(figsize=(15, 2))
35    plt.plot(series.iloc[int(len(series) * 0.8):], label='Observed')
36    plt.plot(recomposed_preds, color='red', label='Predicted')
37    plt.legend()
38    plt.show()
39    test_score = np.sqrt(mean_squared_error(series.iloc[int(len(series)
          * 0.8):], recomposed_preds))
40    #print(f'RMSE: {test_score}')
41
42    value_pred = pd.concat([series.iloc[-int(len(series) * 0.2):].
          pct_change(),recomposed_preds.iloc[-int(len(series) * 0.2):].
          pct_change()], axis=1)
43    value_pred.dropna(inplace=True)
44    value_pred.columns = ['Value', 'preds']
45    value_pred['hit'] = np.where(np.sign(value_pred['Value']) == np.sign
          (value_pred['preds']), 1, 0)
```

Este código es idéntico al anterior usado, salvo que el algoritmo que usamos es `STL`. En la línea 5 obtenemos la descomposición con este método y pasamos esta descomposición a las correspondientes varaibles en la siguiente línea. Todo lo demás es como el código de arriba descrito.

Los resultados obtenidos para cada partícula muestran en orden las predicciones para la estacionalidad, la tendencia, los valores residuales y finalmente la predicción de la serie completa.

En las figuras 5.16, 5.17, 5.18 y 5.19 vemos las gráficas obtenidas para la zona de Roadside y los errores obtenido lo vemos en la tabla 5.7. Vamos a ver detalladamente cada una de las partículas.

- Óxido nítrico y óxidos de nitrógeno: La predicción de la tendencia es casi perfecta, solo hay algunos puntos en los que se puede diferenciar un poco entre el valor real y el predicho. La predicción de la estacionalidad es también muy buena, hay varios momentos en los que se sobrepone la línea generada por la predicción sobre los reales y se ajusta con bastante precisión en los demás momentos. Sin embargo, los valores para los residuos si difieren y no se ajustan muy bien.
  Finalmente, la predicción de la serie no es mala ya que se ajusta bien a la trazabilidad de los valores reales y consigue predecir bien los valores para ciertos instantes.
- Dióxido de nitrógeno: La predicción de la tendencia se ajusta a la

realidad casi a la perfección. De igual forma, la predicción de la estacionalidad es bastante buena, en general hay poca variación entre los valores reales y los predichos. Sin embargo, la predicción del error es bastante mala, ya que predice una línea recta cuando por el contrario tiene mucha variación los valores reales.
Finalmente, la predicción de la serie es aceptable, es un poco peor que para otras partículas pero consigue ajustarse al movimiento de los datos e incluso predecir con poco error varios valores reales.

- Ozono: La predicción para la tendencia roza la perfección. Si nos centramos en la estacionalidad, vemos que se ajusta bastante bien a la trazabilidad de los valores reales aunque haya algunos momentos en que no consigue representar algunos máximos y mínimos más marcados. Por otro lado, la predicción del error es muy mala. Ocurre al igual que con el dióxido de nitrógeno, su predicción es una línea recta que no se ajusta al movimiento ni a los valores reales.
  Finalmente, la predicción es aceptable,consigue ajsutarse bastante bien al movimiento de los datos, aunque se vean diferencias en los extremos relativos de los valores reales.

- Partículas PM10 y PM2.5: La tendencia se ajusta muy bien, aunque desde agosto a noviembre de 2018 se equivocan en la predicción de los valores. La predicción de la estacionalidad es aceptable, pero no consigue ajustarse a los valores máximos y mínimos que contienen los valores reales. La predicción para el error es muy mala, predice una línea recta cuando por el contrario los valores reales varían mucho. Finalmente, la predicción de la serie no es buena, ya que se aprecian bastantes diferencias entre ambas.

- Dióxido de azufre: La predicción de la tendencia es realmente buena, solo se aprecia una mayor diferencia entre octubre de 2018 y enero de 2019. Por otro lado, la predicción de la estacionalidad es aceptable. Para algunos sitios si hay bastante diferencia entre los valores que predice y los reales, pero también hay bastantes momentos en los que acierta y su error es muy pequeño. Para el error, la predicción es bastante mala ya que no consigue parecerse al movimiento y los valores reales.
  Finalmmente, la predicción de la serie no es demasiado mala. Consigue ajustarse un poco mejor al principio de la serie, pero después no predice bien con tantos cambios de sentido de los valores reales.

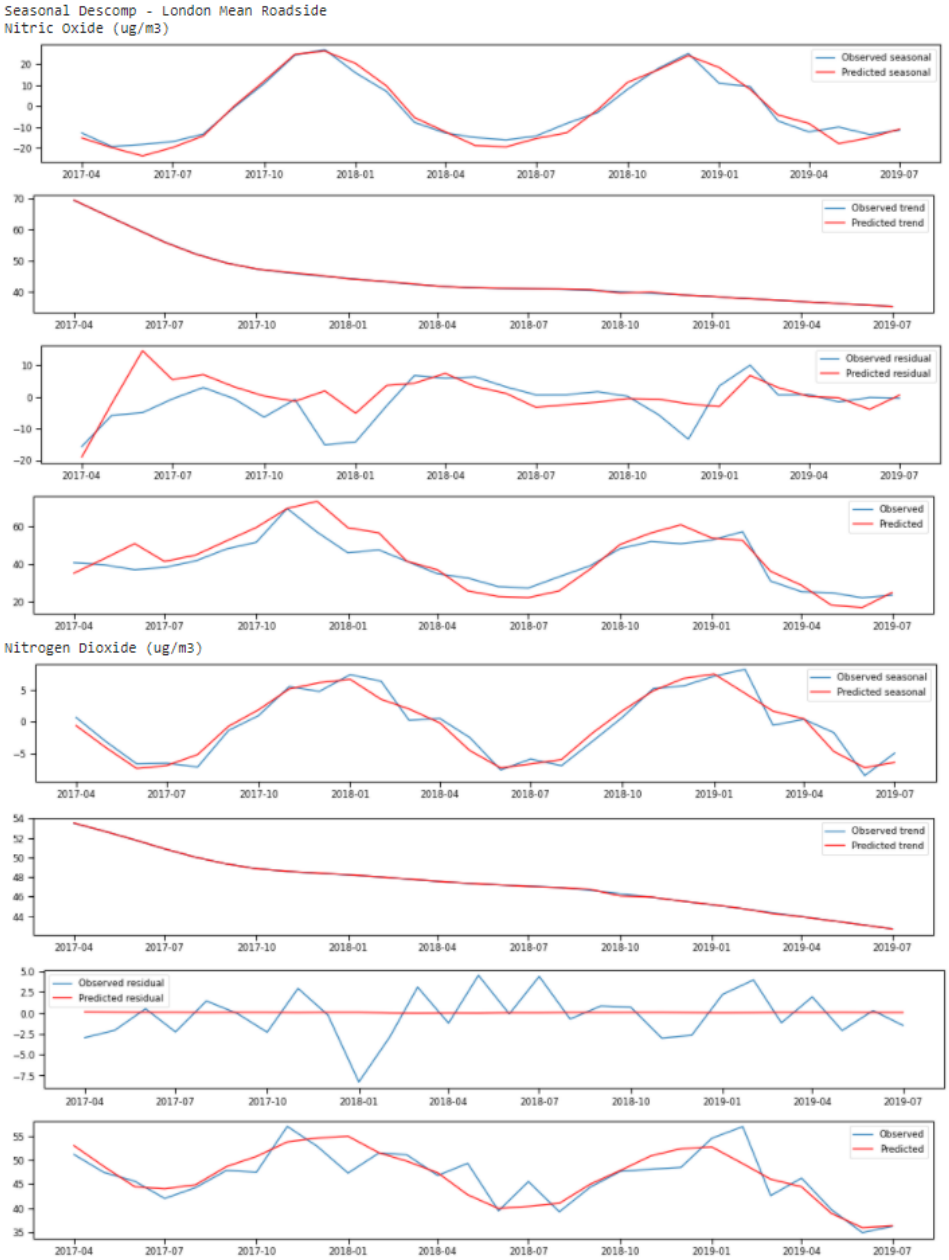

Figura 5.16: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Roadside y las partículas de óxido nítrico y dióxido de nitrógeno.

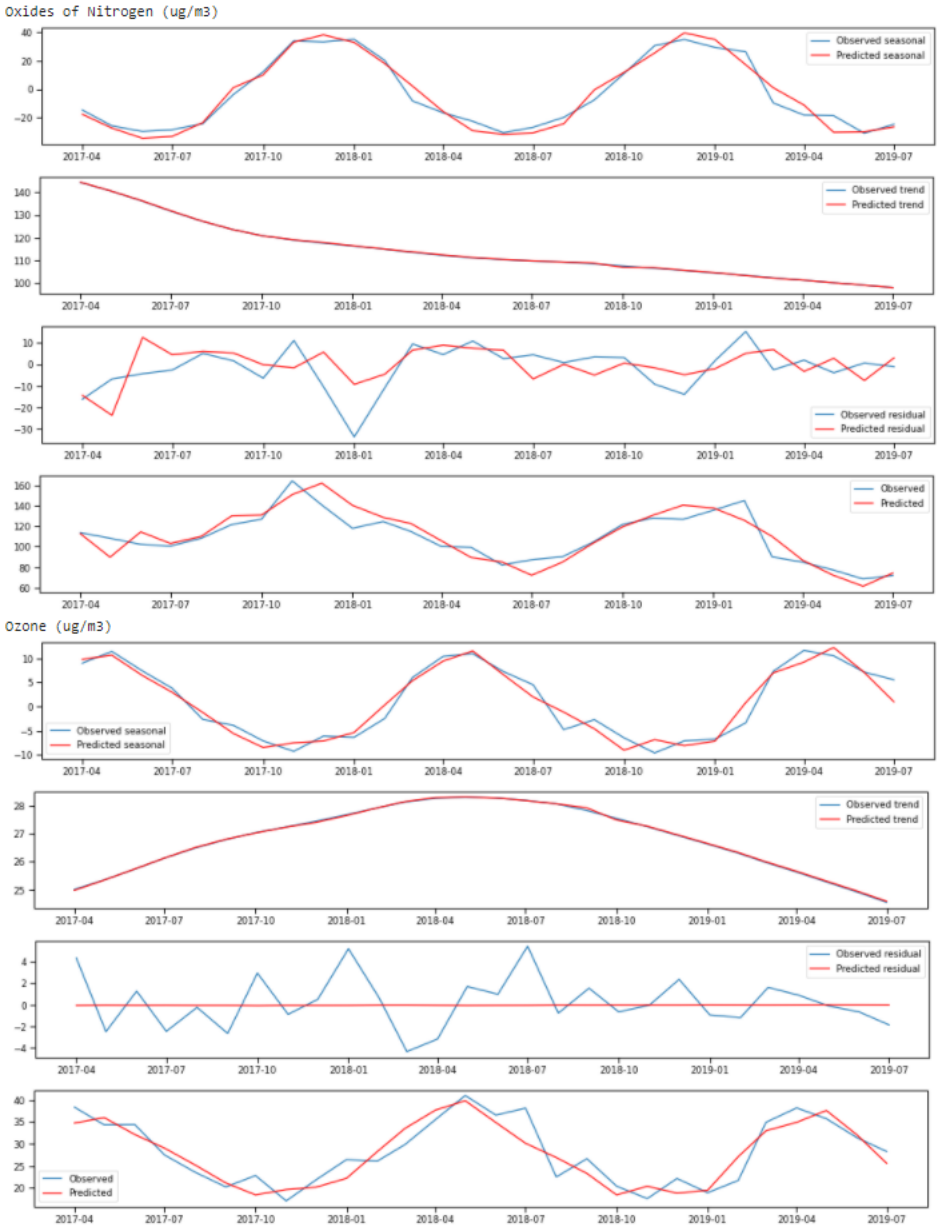

Figura 5.17: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Roadside y las partículas de óxidos de nitrógeno y ozono.

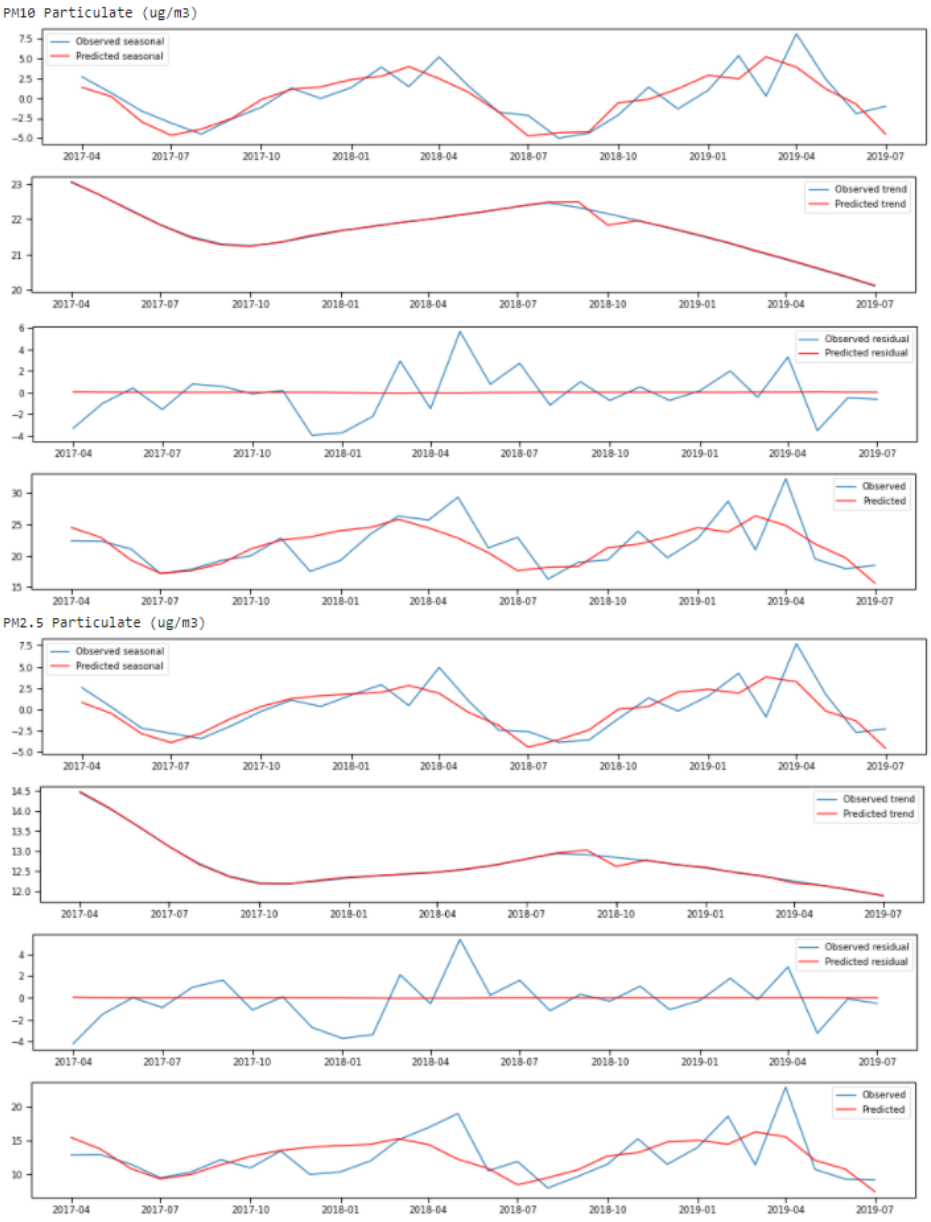

Figura 5.18: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Roadside y las partículas PM10 y PM2.5.

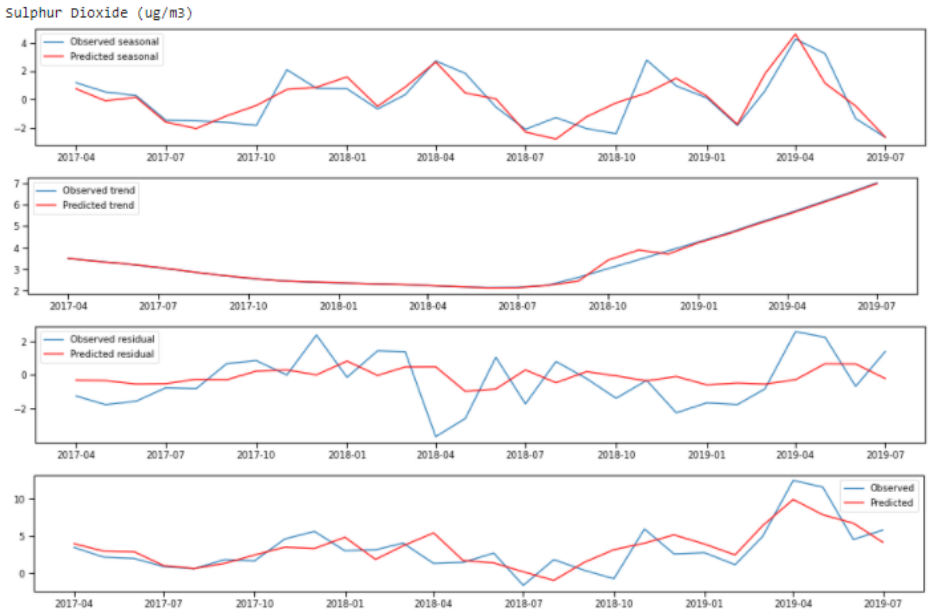

Figura 5.19: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Roadside y la partícula de sulfuro de dióxido.

	$RMSE_{Est.}$	$RMSE_{Tend.}$	$RMSE_{Error}$	MRSE	Tasa
Óxido nítrico	3,23	0,13	6,59	6,83	73,08 %
Dióxido de nitrógeno	1,52	0,05	2,78	3,12	69,23 %
Óxidos de nitrógeno	5,44	0,18	9,40	10,77	73,08 %
Ozono	1,99	0,03	2,36	3,11	61,54 %
Partículas PM10	2,01	0,07	2,20	3,20	65,38 %
Partículas PM2.5	1,86	0,05	2,09	2,90	61,54 %
Dióxido de azufre	1,01	0,12	1,57	1,92	53,85 %

Tabla 5.7: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para la descomposición estacional con el algoritmo STL en la zona de Roadside.

Las figuras 5.20, 5.21, 5.22 y 5.23 son las correspondientes a la zona de Background y los errores obtenido lo vemos en la tabla 5.8. Vamos a ver detalladamente los resultados.

- Óxido nítrico y óxidos de nitrógeno: La predicción de la tendencia es casi perfecta, solo hay algunos momentos en los que difieren los valores

reales y predichos. Para la estacionalidad la predicción es también muy buena, se ajusta bastante bien al movimiento de los valores reales y es bastante preciso en general. Sin embargo, los valores para los residuos si son muy distintos y no se ajustan bien.
Finalmente, la predicción de la serie no llega a ser buena, se ajusta bien a la trazabilidad de los valores reales, pero podemos ver bastantes diferencias entre los valores y no consigue ajustarse a los valores máximos ni mínimos.

- Dióxido de nitrógeno: La predicción de la tendencia se ajusta a la realidad casi a la perfección. De igual forma, la predicción de la estacionalidad es muy buena, en general hay poca variación entre ambos. La predicción del error es bastante mala, ya que predice una línea recta cuando por el contrario tiene mucha variación los valores reales.
Finalmente, la predicción de la serie es aceptable, es un poco peor que para otras partículas pero consigue ajustarse al movimiento de los datos e incluso predecir con poco error varios valores reales.

- Ozono: La predicción para la tendencia roza la perfección. Si nos centramos en la estacionalidad, vemos que se ajusta bastante bien a la trazabilidad, aunque no consiga ajustarse a los extremos de los valores reales.. Por otro lado, la predicción del error es muy mala. Ocurre al igual que con el dióxido de nitrógeno, su predicción es similar a una línea recta que no se ajusta al movimiento ni a los valores reales.
Finalmente, la predicción es aceptable,consigue ajsutarse bastante bien al movimiento de los datos, aunque se vean diferencias en los extremos relativos de los valores reales.

- Partículas PM10 y PM2.5: La tendencia se ajusta muy bien, aunque desde agosto a noviembre de 2018 se equivocan en la predicción de los valores. La predicción de la estacionalidad es aceptable, pero no consigue ajustarse a los valores máximos y mínimos que contienen los valores reales. La predicción para el error es muy mala, predice una línea recta cuando por el contrario los valores reales varían mucho. Finalmente, la predicción de la serie no es buena, ya que se aprecian bastantes diferencias entre ambas.

- Dióxido de azufre: La predicción de la tendencia es realmente buena, solo se aprecia una mayor diferencia entre octubre de 2018 y enero de 2019. Por otro lado, la predicción de la estacionalidad es aceptable. Para algunos sitios si hay bastante diferencia entre los valores que predice y los reales, pero también hay bastantes momentos en los que acierta y su error es muy pequeño. Para el error, la predicción es bastante mala ya que no consigue parecerse al movimiento y los valores reales.

Finalmmente, la predicción de la serie no es demasiado mala. Consigue ajustarse un poco mejor al principio de la serie, pero después no predice bien con tantos cambios de sentido de los valores reales.

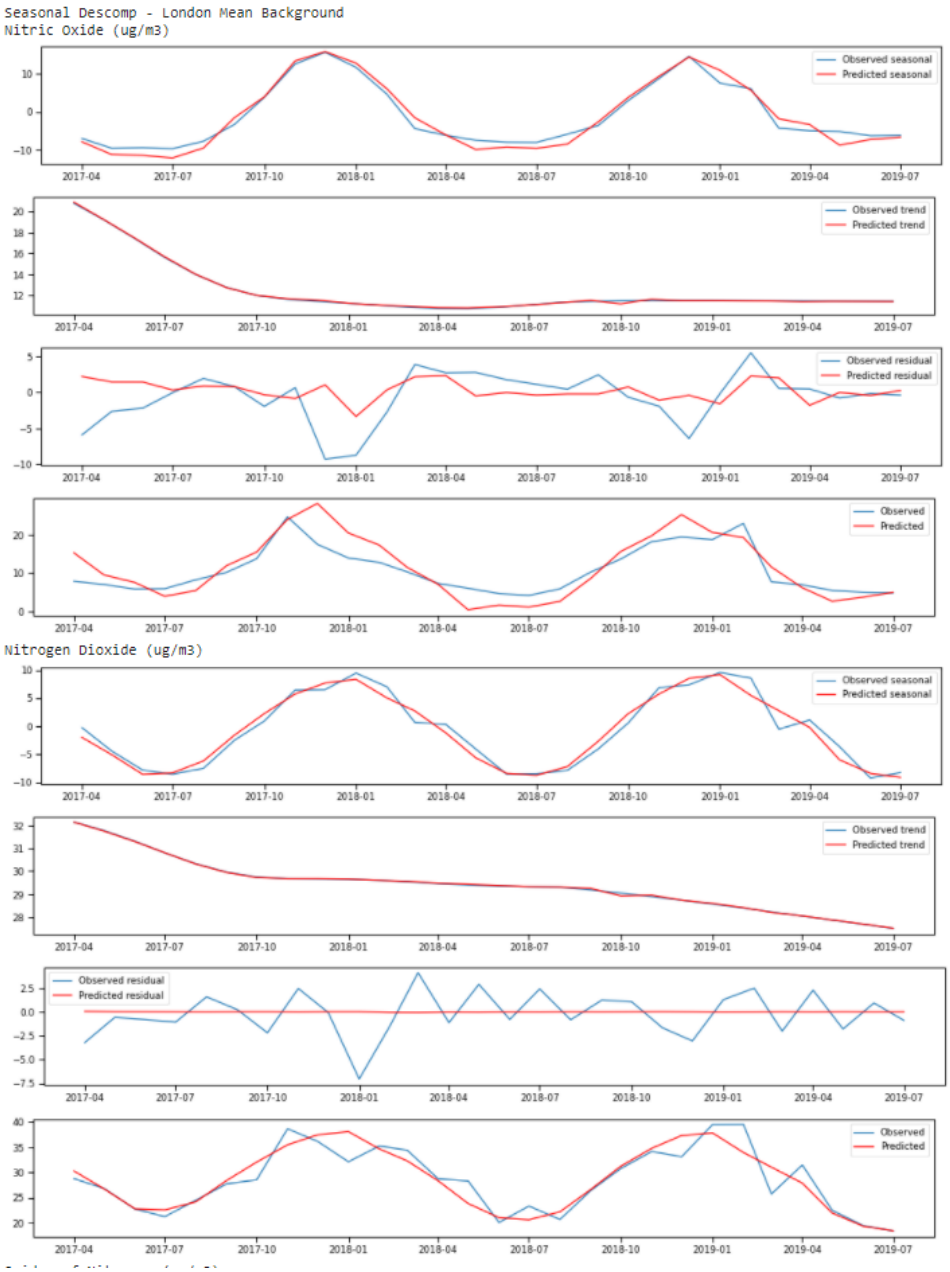

Figura 5.20: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Background y las partículas de óxido nítrico y dióxido de nitrógeno.

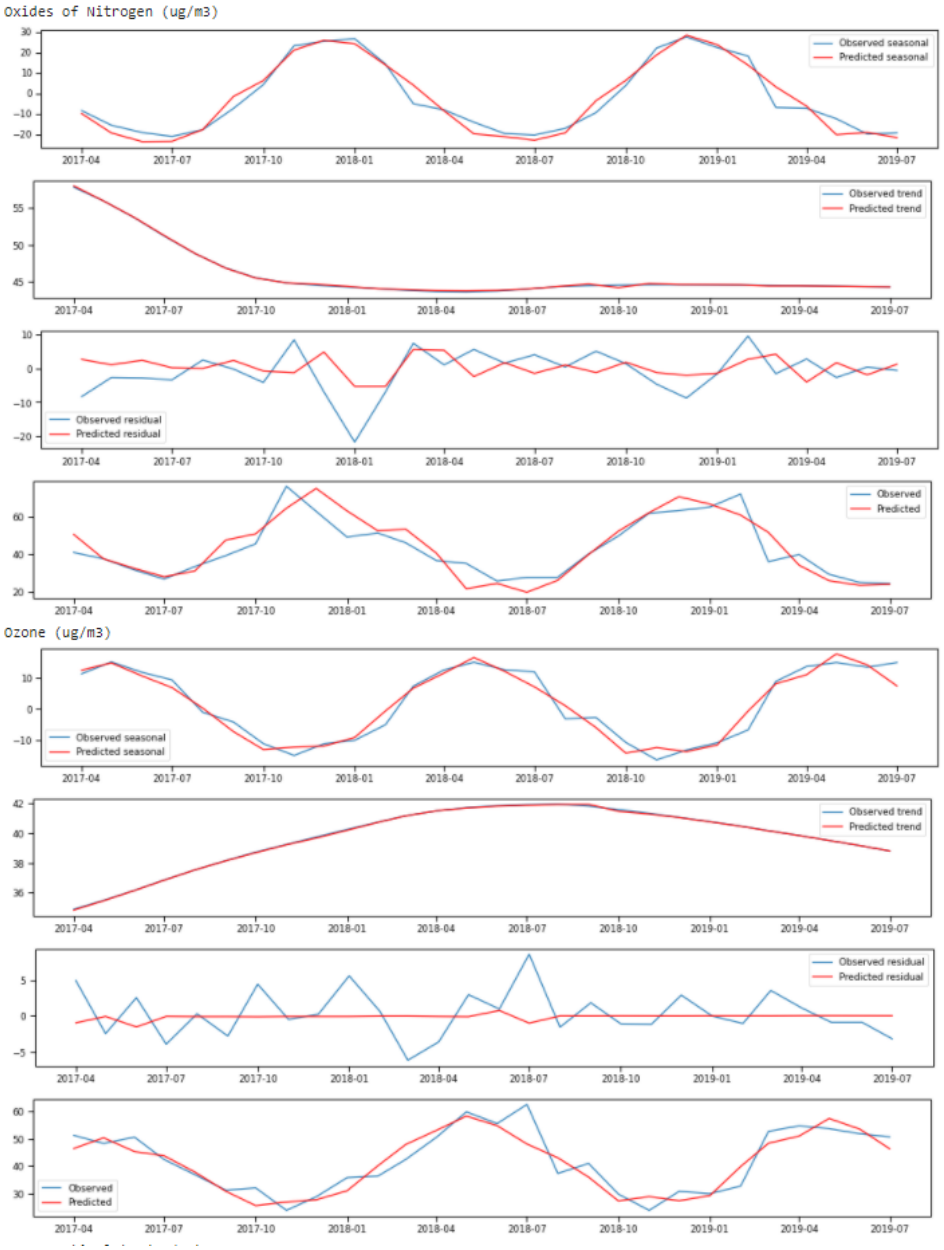

Figura 5.21: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Background y las partículas de óxidos de nitrógeno y ozono.

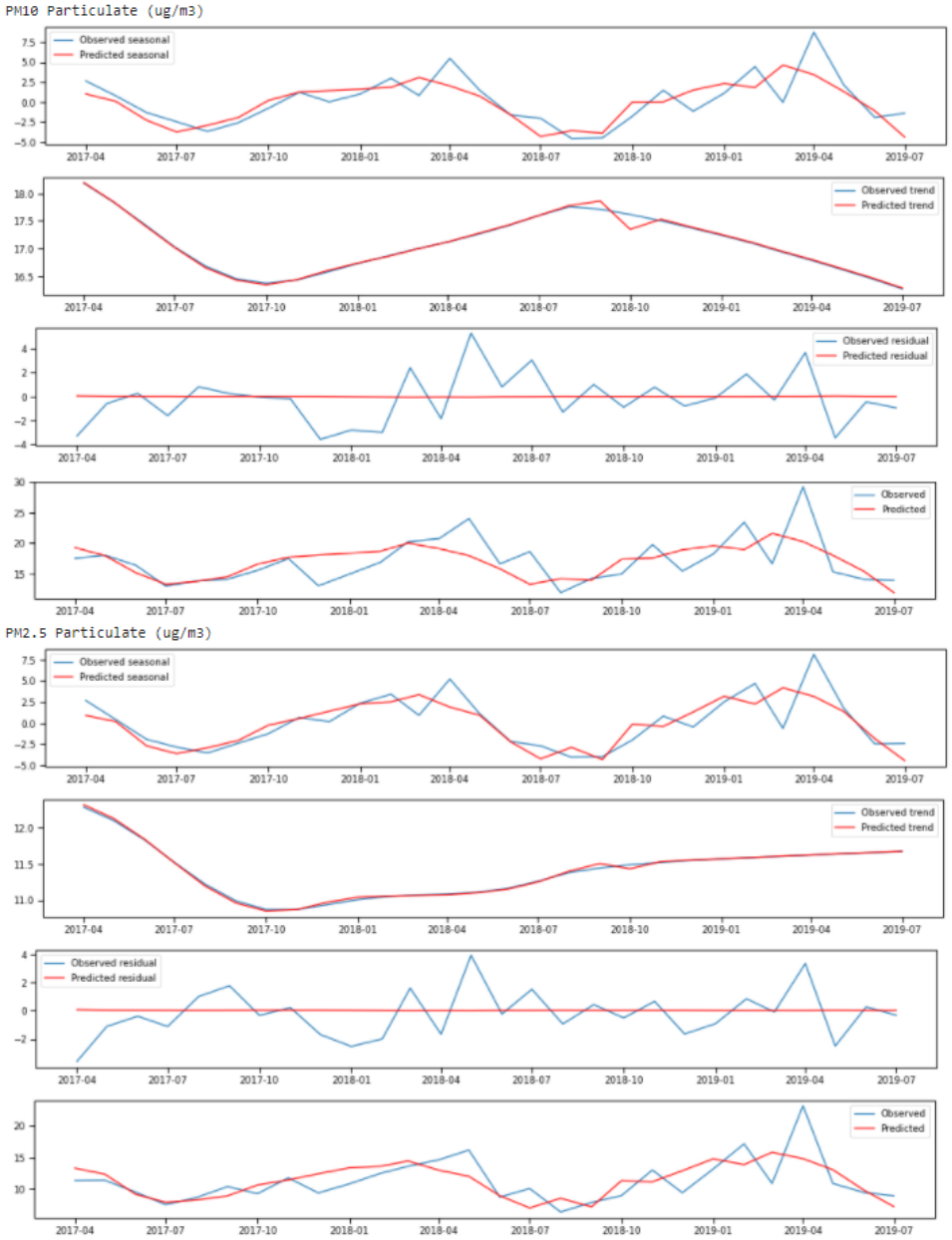

Figura 5.22: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Background y las partículas PM10 y PM2.5.

Figura 5.23: Tendencia, estacionalidad, error y gráfica completa usando la descomposición estacional con el algoritmo STL para la zona de Background y la partícula de sulfuro de dióxido.

	$RMSE_{Est.}$	$RMSE_{Tend.}$	$RMSE_{Error}$	MRSE	Tasa
Óxido nítrico	1,72	0,08	3,47	3,81	76,92 %
Dióxido de nitrógeno	1,48	0,03	2,32	2,63	69,23 %
Óxidos de nitrógeno	4,05	0,10	6,16	7,30	65,38 %
Ozono	2,97	0,04	3,43	4,70	65,38 %
Partículas PM10	2,03	0,06	2,13	3,17	61,54 %
Partículas PM2.5	1,85	0,02	1,71	2,65	53,85 %
Dióxido de azufre	0,33	0,00	0,55	0,64	65,38 %

Tabla 5.8: Tabla con el error cuadrático medio y la tasa de acierto de la orientabilidad para la descomposición estacional con el algoritmo STL en la zona de Background.

5.2. Alisamiento Exponencial

En esta sección, vamos a ver como implementar y los resultados que obtenemos con los distintos tipos de alisamiento exponencial que describimos en el capítulo 3.

Lo primero que vamos a hacer es usar el método más simple del alisamiento exponenial que se denomina alisamiento exponencial simple.

Recordamos que con este modelo la predicción se calcula como una media ponderada de los datos. Tenemos que elegir el parámetro de alisamiento α, que puede tomar los valores entre 0 y 1.
Si α es 0, la predicción será la media de los datos históricos y si α es 1, la predicción será igual al último valor.

Para nuestro ejemplo, vamos a hacer tres pruebas de predicción con tres valores distintos de α.
El primero de ellos será un valor pequeño de $\alpha = 0.2$, el segundo de ellos el propio algoritmo eligirá el mejor α para que el error cuadrático medio sea menor y el tercero será un valor grande, $\alpha = 0.8$.

```
print('Simple Exponential Smoothing - London Mean Roadside')
for name, metric in zip(names, metrics):
  series = pr_LMR[metric]
  train_pr = series.iloc[:int(len(series) * 0.8)]
  test_pr = series.iloc[int(len(series) * 0.8):]

  # specific smoothing level 0.2
  fit1 = SimpleExpSmoothing(train_pr).fit(smoothing_level=0.2,
      optimized=False)
  fcast1 = fit1.forecast(int(len(series) * 0.2)).rename(r'$\alpha={}$'
      .format(0.2))
  mse1 = ((fcast1 - test_pr) ** 2).mean()
  print('The Root Mean Squared Error of our forecasts with smoothing
      level of {} is {}'.format(0.2,round(np.sqrt(mse1), 2)))

  ## auto optimization
  fit2 = SimpleExpSmoothing(train_pr).fit()
  fcast2 = fit2.forecast(int(len(series) * 0.2)).rename(r'$\alpha=%s$'
      %fit2.model.params['smoothing_level'])
  mse2 = ((fcast2 - test_pr) ** 2).mean()
  print('The Root Mean Squared Error of our forecasts with auto
      optimization is {}'.format(round(np.sqrt(mse2), 2)))

  # specific smoothing level 0.8
  fit3 = SimpleExpSmoothing(train_pr).fit(smoothing_level=0.8,
      optimized=False)
  fcast3 = fit3.forecast(int(len(series) * 0.2)).rename(r'$\alpha={}$'
      .format(0.8))
  mse3 = ((fcast3 - test_pr) ** 2).mean()
  print('The Root Mean Squared Error of our forecasts with smoothing
      level of {} is {}'.format(0.8,round(np.sqrt(mse3), 2)))

  series.plot(marker='o', color=u'#1f77b4', legend=True, figsize=(16,
      5))
  fcast1.plot(marker='o', color=u'#9acd32', legend=True)
  fcast2.plot(marker='o', color=u'#ffae42', legend=True)
  fcast3.plot(marker='o', color=u'#b05195', legend=True)
  fit1.fittedvalues.plot(marker='o', color=u'#9acd32')
  fit2.fittedvalues.plot(marker='o', color=u'#ffae42')
  fit3.fittedvalues.plot(marker='o',  color=u'#b05195')

```

```
33  plt.show()
```

Lo primero que hacemos es una copia de los valores de nuestra métrica en una variable que denominamos serie. A continuación, dividimos nuestro conjunto en dos: uno de entrenamiento que contiene el 80 % de los datos y otro de testeo que contiene el 20 % restante.

Para hacer esta predicción vamos a usar la función `SimpleExpSmoothing` y vamos a hacer tres tipos de predicciones distintas variando el parámetro de alisamiento.

En la línea 8 entrenamos nuestro modelo pasandole el conjunto de entrenamiento y establecemos el parámetro de alisamiento a $0,2$. Seguidamente, en la línea 9 realizamos la predicción con el conjunto de datos de testeo. En las líneas 10 y 11 calculamos el error cuadrático medio y lo mostramos.

Nuestra segunda opción se encuentra en la línea 14, pero esta vez sin pasarle manualmente el parámetro de alisamiento. De esta forma, el método es el encargado de elegir el valor óptimo. En la línea 15 hacemos la predicción para esta opción. Calculamos el error cuadrático medio y lo mostramos en las líneas 16 y 17.

Nuestra última predicción con el método de alisamiento simple se encuentra en la línea 20. Entrenamos nuestro modelo con un parámetro de alisamiento de $0,8$ y en la línea 21 realizamos la predicción con nuestro último 20 % de los datos. En las líneas 22 y 23 calculamos y mostramos el error cuadrático medio.

Finalmente, desde la línea 25 a la 33, dibujamos los valores reales de la métrica y las tres predicciones realizadas.

En las gráficas que visualizamos vemos los valores reales en azul, la predicción con el parámetro $\alpha = 0.2$ en verde, con el parámetro $\alpha = 0.8$ en naranja y α optimizado y elegido por el algoritmo en rosa.

Vamos a ver como se comportan cada una de las partículas en cada zona. Primero vamos a ver las gráficas de la zona de Roadside. Vamos a usar a la vez la tabla 5.9 en la que vemos los errores de predicción.

Tenemos que indicar que la predicción que se hace para futuro es una línea recta, la cuál se calcula como indicamos en el capítulo 3 de metodología.
Este modelo puede utilizarse para obtener una línea de base para comparar.

- Óxido nítrico:Vemos en las figura 5.24 que el valor de α optimizado es bastante alto, concretamente 0.89. El error cuadrático menor se consigue con el valor elegido por el algoritmo, seguido del $\alpha = 0.8$. Si usamos un valor bajo, como es 0.2, el error cuadrático medio es bastante grande, llegando a 19.04.

- Dióxido de nitrógeno: Podemos ver en la figura 5.25 que el valor más óptimo es también muy alto, casi un 0.87. Al igual que hemos visto con la partícula anterior, el valor optimizado es el que nos da un menor error cuadrático medio.

- Óxidos de nitrógeno: Podemos ver el comportamiento de las partículas en la figura 5.26 el valor optimizado de α es muy elevado, exactamente su valor es 0.91. Aunque con el valor optimizado tenemos el menor error obtenido de las tres predicciones, los tres errores tienen valores muy altos, siendo el de $\alpha = 0.2$ superior a 30.

- Ozono: Podemos ver sus predicciones en la figura 5.27 en la que vemos que el valor optimizado es extremadamente alto, exactamente de 0.99. Para esta partícula obtenemos menos error con el $\alpha = 0.8$.

- Partículas PM10 y PM2.5: Sus gráficas las podemor ver en la figura 5.28 y 5.29 el valor que ha considerado el algorimto como óptimo es en ambas muy cercano a 0. El error obtenido por $\alpha = 0.8$ es el más bajo para las partículas de tamaño PM10, sin embargo para las de PM2.5 es el obtenido por el optimizado.

- Dióxido de azufre: vemos en la figura 5.30, vemos que el valor de α más óptimo es también bastante pequeño sobre 0.8 y que el valor con el que hemos conseguido el mejor error cuadrático medio no es el que considera el algoritmo más óptimo, si no el del $\alpha = 0.2$.

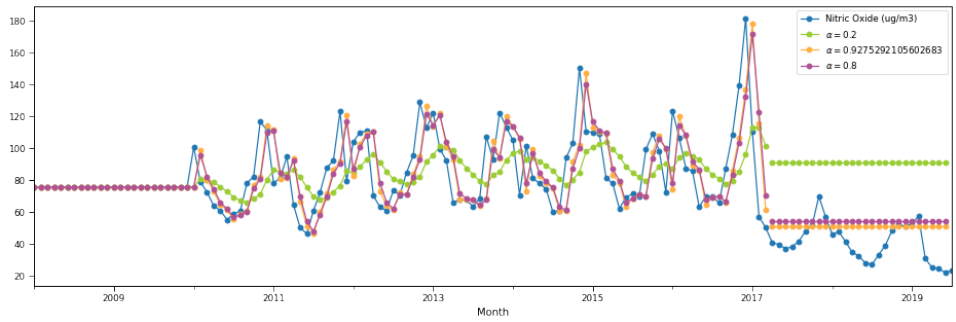

Figura 5.24: Alisamiento exponencial Simple para la partícula Óxido nítrico en la zona de Roadside. Los valores del parámetro de alisamiento son 0.2, 0.8 y optimizado por el algoritmo.

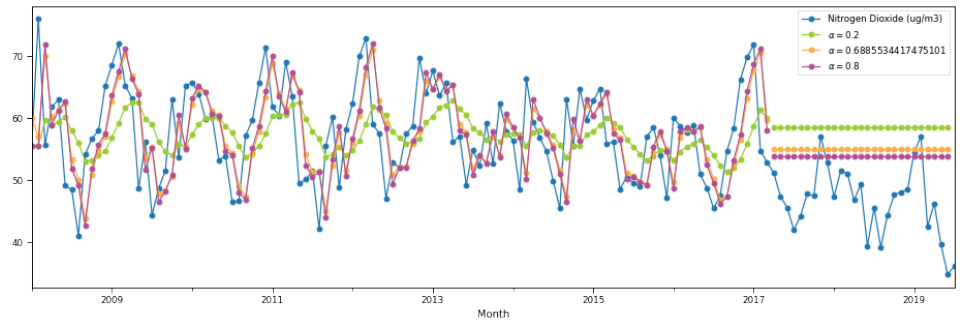

Figura 5.25: Alisamiento exponencial Simple para la partícula de dióxido de nitrógeno en la zona de Roadside. Los valores del parámetro de alisamiento son 0.2, 0.8 y optimizado por el algoritmo.

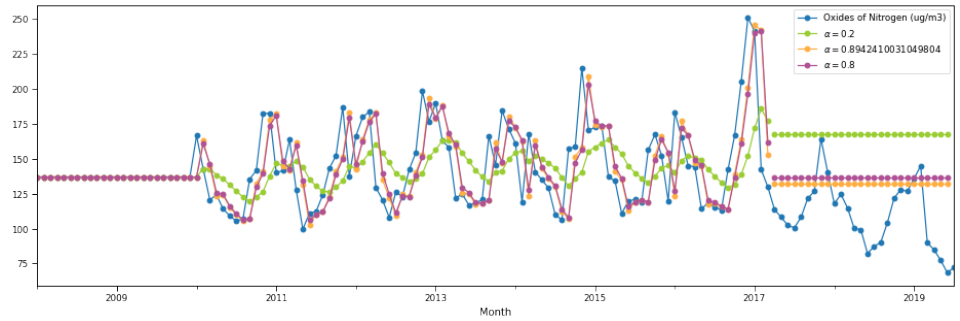

Figura 5.26: Alisamiento exponencial Simple para la partícula óxidos de nitrógeno en la zona de Roadside. Los valores del parámetro de alisamiento son 0.2, 0.8 y optimizado por el algoritmo.

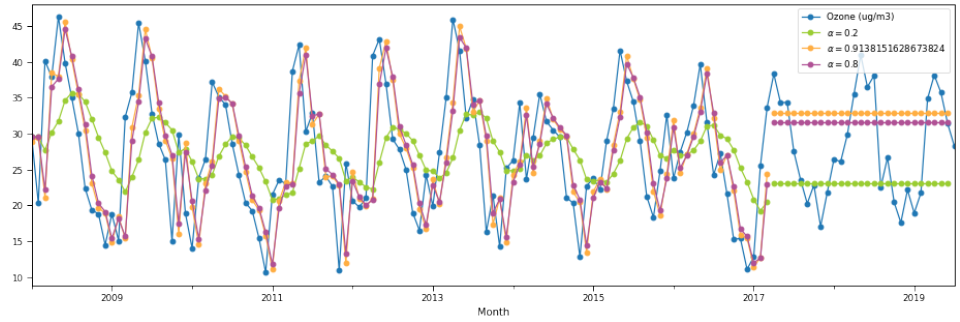

Figura 5.27: Alisamiento exponencial Simple para la partícula de Ozono en la zona de Roadside. Los valores del parámetro de alisamiento son 0.2, 0.8 y optimizado por el algoritmo.

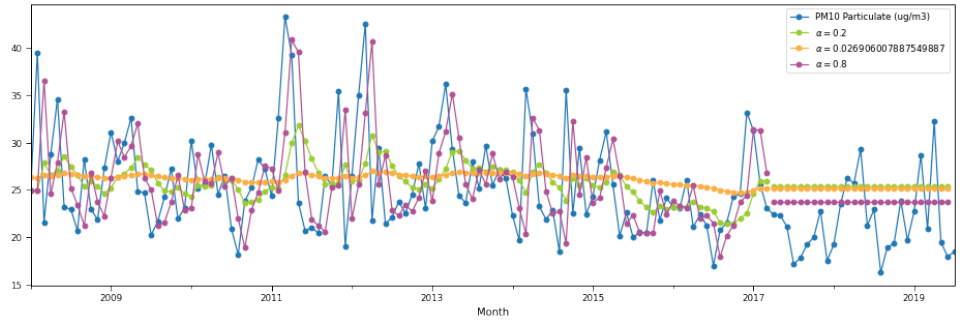

Figura 5.28: Alisamiento exponencial Simple para la partícula PM10 en la zona de Roadside. Los valores del parámetro de alisamiento son 0.2, 0.8 y optimizado por el algoritmo.

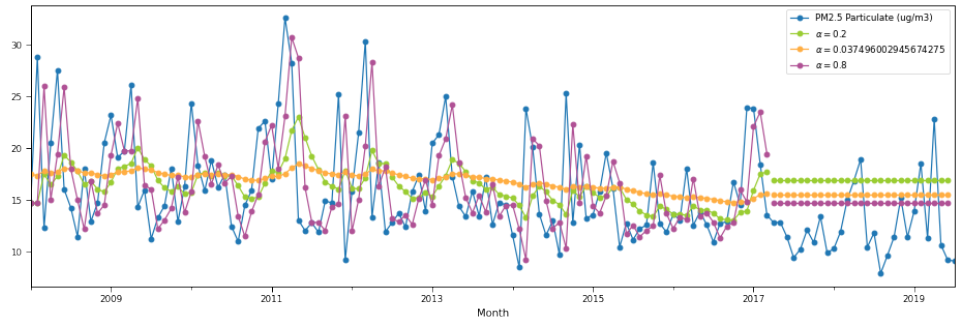

Figura 5.29: Alisamiento exponencial Simple para la partícula PM2.5 en la zona de Roadside. Los valores del parámetro de alisamiento son 0.2, 0.8 y optimizado por el algoritmo.

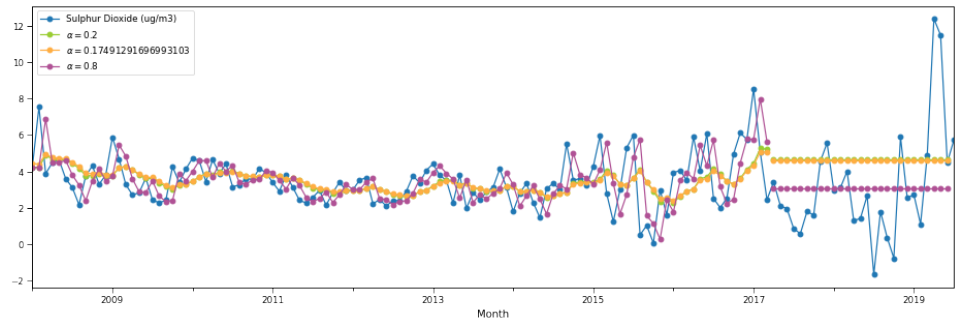

Figura 5.30: Alisamiento exponencial Simple para la partícula de dióxido de azufre en la zona de Roadside. Los valores del parámetro de alisamiento son 0.2, 0.8 y optimizado por el algoritmo.

	$\alpha = 0.2$	$\alpha = Optimizado$	$\alpha = 0.8$
Óxido nítrico	51.06	17.9	17.12
Dióxido de nitrógeno	12.65	9.53	8.65
Óxidos de nitrógeno	61.07	30.86	33.79
Ozono	8.92	8.6	7.97
Partículas PM10	5.23	5.08	4.34
Partículas PM2.5	5.42	4.44	3.96
Dióxido de azufre	3.47	3.44	3.08

Tabla 5.9: Tabla con errores cuadráticos medios para el método Simple Exponential Smoothing en la zona de Roadside

	$\alpha = 0.2$	$\alpha = Optimizado$	$\alpha = 0.8$
Óxido nítrico	19.04	5.85	5.89
Dióxido de nitrógeno	9.4	6.97	7.21
Óxidos de nitrógeno	30.84	15.2	16.35
Ozono	15.67	11.38	11.22
Partículas PM10	4.73	4.58	4.04
Partículas PM2.5	4.3	3.64	3.79
Dióxido de azufre	1.54	1.58	1.62

Tabla 5.10: Tabla con errores cuadráticos medios para el método Simple Exponential Smoothing en la zona de Background

A continuación, vamos a probar con otro método distinto, el método de Holt con tendencia lineal.
De esta forma pronosticamos datos que tienen una tendencia, por tanto ahora tenemos que especificar dos parámetros. Elegiremos el parámetro de suavizado α y un parámetro de suavización de la tendencia β^*, ambos tienen sus valores entre 0 y 1.

Al igual que con el método anterior, vamos a hacer tres pruebas de predicción con tres valores distintos de α y β^*.

En general, hemos obtenido los mejores valores del error cuadrático medio con $\alpha = 0.8$, por ello, la primera prueba que hacemos tendrá un valor de $\alpha = 0.8$ y un parámetro de suavización de tendencia $\beta^* = 0.2$, además el modelo que usamos es el modelo aditivo de Holt por defecto.
La segunda prueba tendrá un valor de $\alpha = 0.8$ y un parámetro de suavización de tendencia $\beta^* = 0.2$, pero usaremos el modelo exponencial.

Finalmente, el tercer método vuelve a ser el modelo aditivo de Holt pero amortiguado. De nuevo el valor de $\alpha = 0.8$ y el de $\beta^* = 0.2$.

```
1  print('Holt's Linear Trend Method - London Mean Roadside')
2  for name, metric in zip(names, metrics):
3    series = pr_LMR[metric]
4    train_pr = series.iloc[:int(len(series) * 0.8)]
5    test_pr = series.iloc[int(len(series) * 0.8):]
6
7    fit1 = Holt(train_pr).fit(smoothing_level=0.8, smoothing_trend=0.2,
         optimized=False)
8    fcast1 = fit1.forecast(int(len(series) * 0.2)).rename("Holt's linear
          trend")
9    mse1 = ((fcast1 - test_pr) ** 2).mean()
10   print('The Root Mean Squared Error of Holt''s Linear trend {}'.
         format(round(np.sqrt(mse1), 2)))
11
12   fit2 = Holt(train_pr, exponential=True).fit(smoothing_level=0.8,
         smoothing_trend=0.2, optimized=False)
13   fcast2 = fit2.forecast(int(len(series) * 0.2)).rename("Exponential
         trend")
14   mse2 = ((fcast2 - test_pr) ** 2).mean()
15   print('The Root Mean Squared Error of Holt''s Exponential trend {}'.
         format(round(np.sqrt(mse2), 2)))
16
17   fit3 = Holt(train_pr, damped_trend=True, initialization_method="
         estimated").fit(smoothing_level=0.8, smoothing_trend=0.2)
18   fcast3 = fit3.forecast(int(len(series) * 0.2)).rename("Additive
         damped trend")
19   mse3 = ((fcast3 - test_pr) ** 2).mean()
20   print('The Root Mean Squared Error of Holt''s Additive damped trend
         {}'.format(round(np.sqrt(mse2), 2)))
21
22   series.plot(marker='o', color='black', legend=True, figsize=(14, 7))
23   fit1.fittedvalues.plot(marker="o", color='blue')
24   fcast1.plot(color='blue', marker="o", legend=True)
25   fit2.fittedvalues.plot(marker="o", color='red')
26   fcast2.plot(color='red', marker="o", legend=True)
27   fit3.fittedvalues.plot(marker="o", color='green')
28   fcast3.plot(color='green', marker="o", legend=True)
29
30   plt.show()
```

Como en todos los demás métodos que hemos visto, lo primero que hacemos es una copia de los valores de nuestra métrica y separar nuestro conjunto de datos en uno de entrenamiento y otro de prueba, lo podemos ver entre las líneas 3 y 5.
Al igual que en el modelo anterior vamos a hacer tres predicciones distintas, pero esta vez usando la función `Holt`.

La primera de ellas la hacemos en la línea 7, entrenamos a nuestro conjunto de datos de entrenamiento con el método de Holt, usando un nivel de alisamiento de $0,8$ y una tendencia de $0,2$. Si recordamos del capítulo 3, tendríamos que v. En la siguiente línea hacemos la predicción para el último 20 % de los datos. A continuación en las líneas 9 y 10 calculamos el error

cuadrático medio y los mostramos. Por defecto se usa el modelo aditivo de Holt.

La siguiente predicción que hacemos en las líneas 12 y 13 es igual que la anterior, pero eligiendo un modelo exponencial. En la línea 14 calculamos su error y la mostramos en la línea 15.

En la línea 17 hacemos la última predicción para este modelo, que sería la versión amortiguada del modelo aditivo. Al igual que en los dos métodos anteriores fijamos $\alpha = 0,8$ y $\beta^* = 0,2$, pero dejamos que el modelo elija el valor más óptimo para el parámetro de amortiguación. En las dos siguientes líneas calculamos su errores y lo mostramos.

Con Holt vemos que se capta la tendencia de los datos. En las gráficas que visualizamos vemos los valores reales en azul, la predicción con el modelo aditivio de Holt en verde, con el modelo exponencial en naranja y con el modelo aditivo amortiguado en rosa.

Vamos a ver como se comportan cada una de las partículas en cada zona. Primero vamos a ver las gráficas de la zona de Roadside para cada partícula. Vamos a usar a la vez la tabla 5.11 en la que vemos los errores de predicción.

- Óxido nítrico: Podemos ver los resultados de la gráfica para el óxido nítrico en la figura 5.31. Vemos que la tendencia lineal es muy drástica y da como resultado una recta decreciente con bastante pendiente y cuyo error cuadrático medio es muy grande, exactamente 154.9.
  El método exponencial nos da un error de 24,9, por tanto es una mejor aproximación con respecto al aditivo amortiguado, que da un error de 29,51.Podemos ver como ambas aproximaciones empiezan con una pequeña curvatura y finalmente tienden a una recta por debajo de los valores reales.

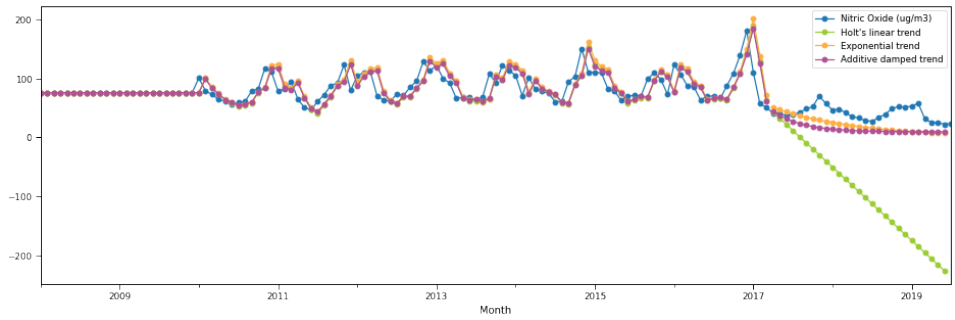

Figura 5.31: Método de Holt lineal para las partículas de óxido nítrico en la zona de Roadside. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Dióxido de nitrógeno: Podemos ver su gráfica en la figura 5.32. Al igual que nos ocurre con la partícula anterior, el método lineal de Holt nos da una recta decreciente aunque en este caso con una pendiente menor. Esta pendiente se va alejando de los datos conforme aumenta el tiempo. El error que tenemos con esta primera aproximación es de 14,93.
  La tendencia exponencial se ajusta un poco, dentro de sus posibilidades, a los datos reales. Para saber como se comporta en un futuro necesitariamos ver la ecuación y estudiar su comportamiento cuando t crece. Tenemos un error de 6,97 con esta aproximación.
  El método de Holt amortiguado es el que obtiene un mejor error cuadrático medio, siendo este de 5,33. Vemos que esta aproximación se encuentra dentro del rango de los datos reales.

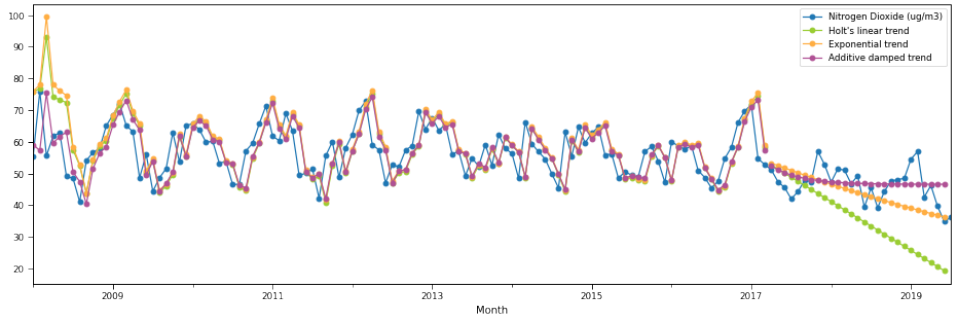

Figura 5.32: Método de Holt lineal para las partículas de dióxido de nitrógeno en la zona de Roadside. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Óxidos de nitrógeno: Podemos ver su gráfica en la figura 5.33. Con el método líneal de Holt ocurre como en el caso del óxido nítrico, obtenemos una recta decreciente con bastante pendiente. El error que obtenemos es muy grande, exactamente de 115,58.
  El método eponencial consigue un error de 28,38 y parece que tiende a ir por debajo de los valores reales.
  El método de Holt amortiguado obtiene de nuevo el menor error con 27,49. Este método realiza una pequeña curva para finalmente terminar en una recta que se encuentra entre los valores más bajos de la serie.

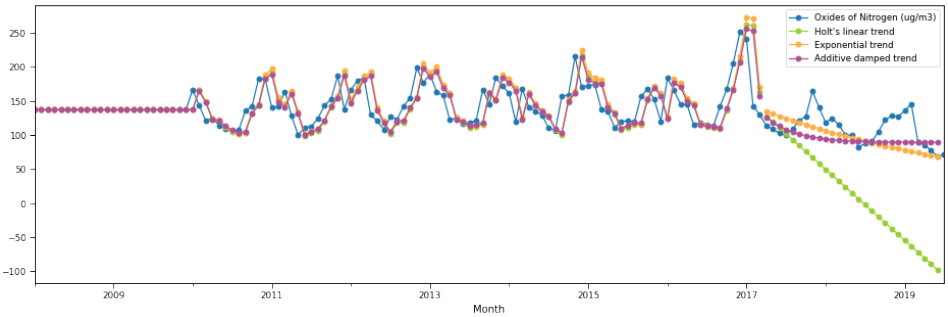

Figura 5.33: Método de Holt lineal para las partículas de óxidos de nitrógeno en la zona de Roadside. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Ozono: Podemos ver en la figura 5.34 que el método lineal toma una tendencia positiva, por lo que se intenta ajustar a los datos con una recta creciente. El error que obtenemos es de 40,5.
  El método exponencial de Holt nos da unos resultados terribles para esta partícula, obtenemos un error de 715,32. Si vemos la predicción y los valores reales no tienen nada que ver.
  Por último el método lineal de Holt amortiguado, nos da un error de 15,05, un error mucho menor que los obtenidos con las otras funciones.

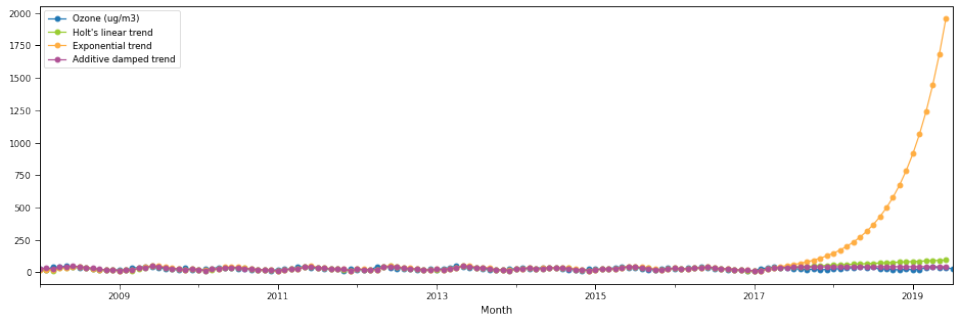

Figura 5.34: Método de Holt lineal para las partículas de ozono en la zona de Roadside. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Partícula PM10: Podemos ver la función obtenida en la figura 5.35, al igual que ocurre con las demás partículas el método lineal nos da una línea recta, para este caso es decreciente, por lo que los valores se alejan de los reales de forma negativa. Su error es de 6,6.
  Para los métodos exponenciales y el lineal amortiguado, sus gráficos se encuentran entre los valores reales de la función. El error del amortiguado es mejor siendo de 4,06, estando muy próximo el del exponencial con un error de 4,18.

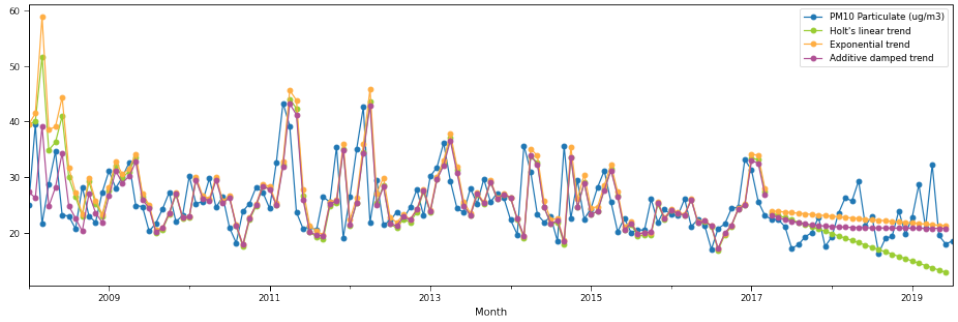

Figura 5.35: Método de Holt lineal para las partículas PM10 en la zona de Roadside. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Partícula PM2.5: Podemos ver el comportamiento de los distintos pruebas en la figura 5.36. El método lineal da como resultado una línea recta decreciente, cun una pendiente no muy grande. Su error cuadrático medio es de 8,22.
  Para los métodos exponenciales y el lineal amortiguado, sus gráficos

se encuentran entre los valores reales de la función. El error del amortiguado es mejor siendo de 3,71, mientras que el error del exponencial es de 3,86.

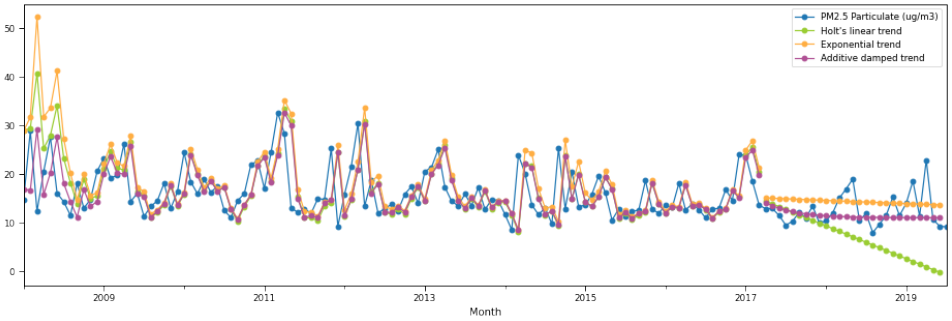

Figura 5.36: Método de Holt lineal para las partículas PM2.5 en la zona de Roadside. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Dióxido de azufre: Podemos observar como se comportan las distintas aproximaciones en la figura 5.37. La forma lineal de Holt, es una recta decreciente que no se ajusta bien a los datos reales, su error es de 7,92. El método eponencial de Holt es el que mejor error consigue con un valor de 3,21. Su aproximación termina siendo una recta creciente.
  El método lineal amortiguado obtiene un error de 3,73, realizando su predicción hacia los valores más pequeños de los datos reales.

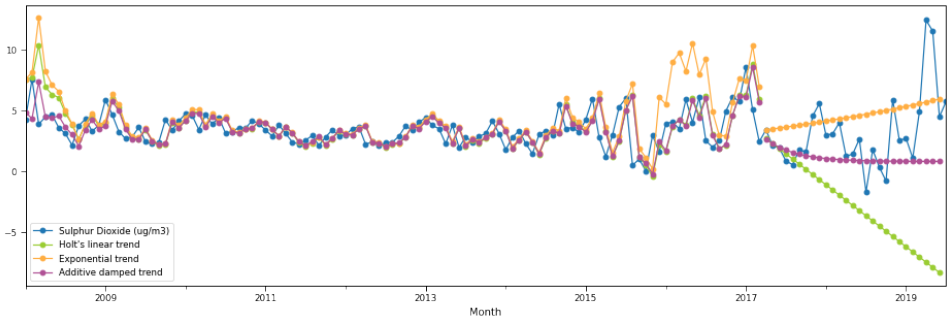

Figura 5.37: Método de Holt lineal para las partículas de dióxido de azufre en la zona de Roadside. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

A continuación, vamos a ver las gráficas de la zona de Background para cada partícula. Vamos a usar a la vez la tabla 5.12 en la que vemos los

errores de predicción.

- Óxido nítrico: Podemos ver los resultados de la gráfica para el óxido nítrico en la figura 5.38. Vemos que la tendencia lineal es muy pronunciada y da como resultado una recta decreciente con bastante pendiente, cuyo error cuadrático medio es muy grande, exactamente 76,86.
  El método exponencial nos da el mejor error cuadrático siendo 6,32. Esta aproximación se ajusta a la media de los valores reales.
  El modelo aditivo amortiguado da un error de 20,97 y vemos que se encuentra bastante por debajo de los valores reales.

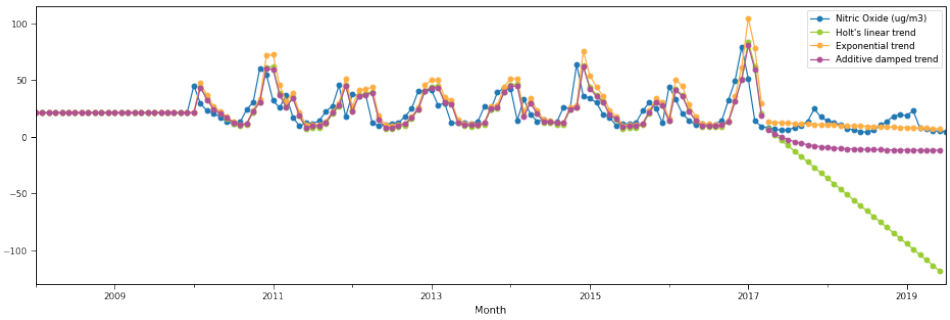

Figura 5.38: Método de Holt lineal para las partículas de óxido nítrico en la zona de Background. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Dióxido de nitrógeno: Podemos ver su gráfica en la figura 5.39.Al igual que nos ocurre con la partícula anterior, el método lineal de Holt nos da una recta decreciente aunque en este caso con una pendiente menor. El error que tenemos con esta primera aproximación es de 12,33. La tendencia exponencial y el método amortiguado linean se ajustan un poco mejor a los datos. Con el exponencial enemos un error de 6,98. El método de Holt amortiguado es el que obtiene un mejor error cuadrático medio, siendo este de 6,47.

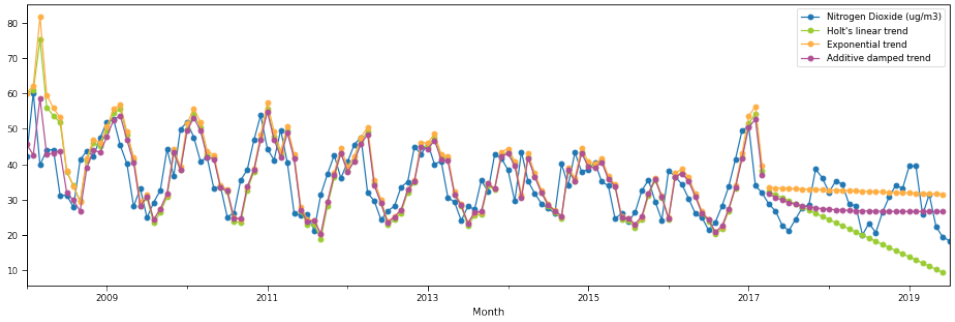

Figura 5.39: Método de Holt lineal para las partículas de dióxido de nitrógeno en la zona de Background. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Óxidos de nitrógeno: Podemos ver su gráfica en la figura 5.40.Con el método líneal de Holt ocurre como en el caso del óxido nítrico, obtenemos una recta decreciente con bastante pendiente. El error que obtenemos es muy grande, exactamente de 83,36.
  El método eponencial consigue un error de 16,35, siendo el mejor para este tipo de partícula en esta zona. Su aproximación parece una recta con una pendiente muy pequeña y negativa, con el valor de la media de los valores.
  El método de Holt amortiguado obtiene un error de 26,29. Este método realiza una pequeña curva por debajo de los datos.

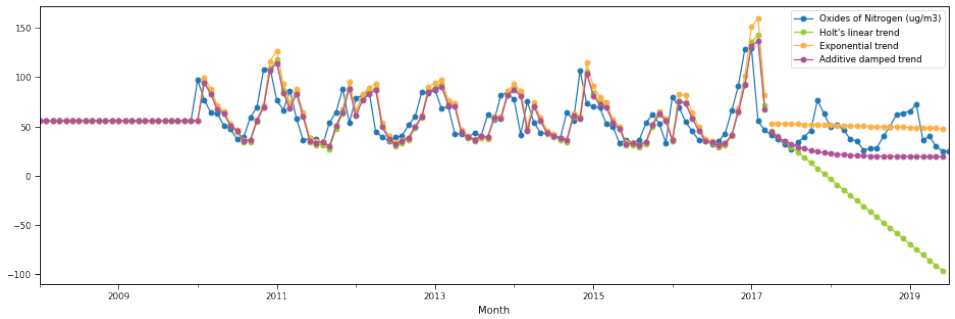

Figura 5.40: Método de Holt lineal para las partículas de óxidos de nitrógeno en la zona de Background. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Ozono: Podemos ver en la figura 5.41 que el método lineal toma una tendencia positiva, por lo que se intenta ajustar a los datos con una recta creciente. El error que obtenemos es de 45,32.

El método exponencial de Holt nos da unos resultados realmente malos para esta partícula, obtenemos un error de 948,77. La predicción toma valores muy altos a comparación con los reales.
El método lineal de Holt amortiguado, nos da un error de 16,52, siendo el mejor de los tres.

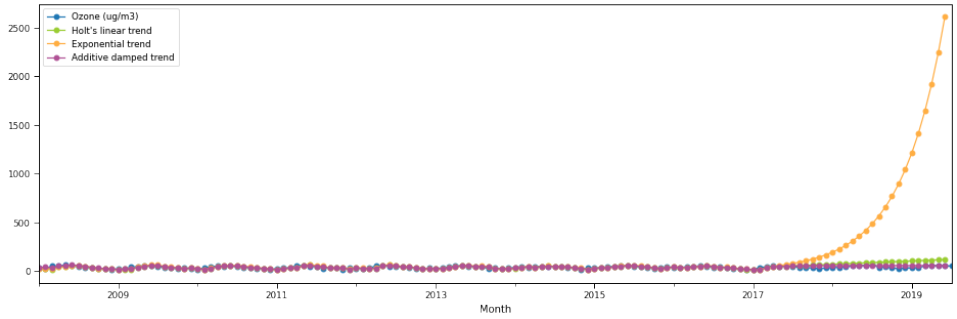

Figura 5.41: Método de Holt lineal para las partículas de ozono en la zona de Background. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Partículas PM10: Podemos ver la función obtenida en la figura 5.42, al igual que ocurre con las demás partículas el método lineal nos da una línea recta. Esta línea es decreciente y con poca pendiente.. Su error es de 6,44.
  Las gráficas de los métodos exponenciales y el lineal amortiguado, muestran sus valores predichos entre los valores reales de la función. El error del amortiguado es de 4,03, mejorandolo el exponencial por una centesima, siendo este de 4,02.

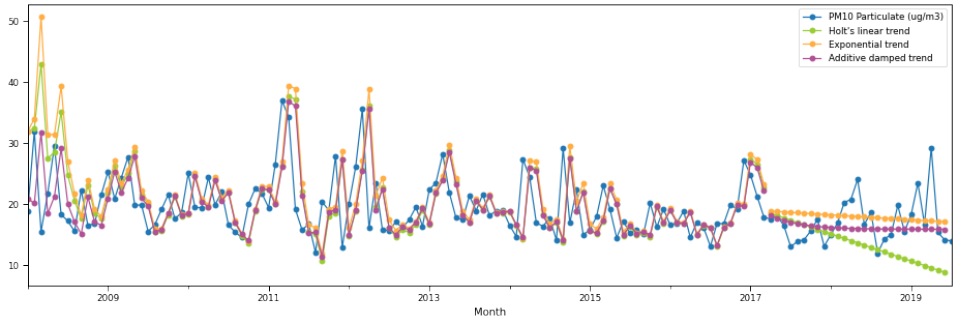

Figura 5.42: Método de Holt lineal para las partículas PM10 en la zona de Background. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Partículas PM2.5: Podemos ver el comportamiento de los distintos pruebas en la figura 5.43. El método lineal da como resultado una línea recta decreciente, cun una pendiente no muy grande y que en la mayoría de su trayectoria cruza los datos reales. Su error cuadrático medio es de 4,5.
  El método exponencial es el que peor resultados nos da, alejandose sus predicciones por encima de los valores reales. El error obtenido para esta aproximación es de 10,36.
  La forma lineal amortiguada, es la que nos da una mejor aproximación, teniendo un error de 3,59.

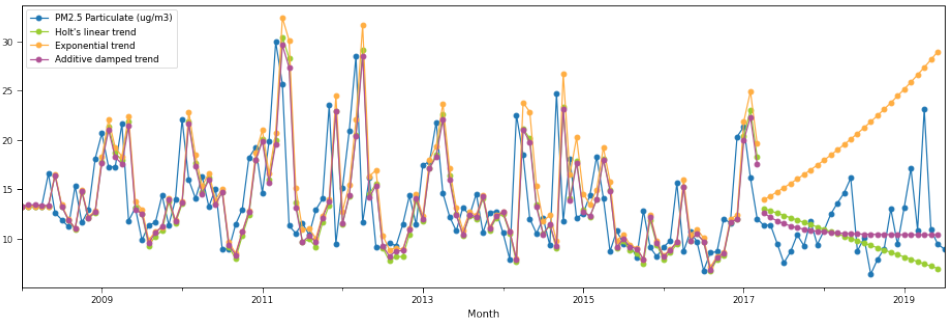

Figura 5.43: Método de Holt lineal para las partículas PM2.5 en la zona de Background. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

- Dióxido de azufre: Podemos observar como se comportan las distintas aproximaciones en la figura 5.44. La forma lineal de Holt, es una recta creciente que se situa por encima de los valores reales. Su error es de 2,04.
  El método eponencial de Holt es el que peor error consigue siendo este de 4,27.
  El método lineal amortiguado obtiene un error de 1,7, siendo el mejor para esta partícula.

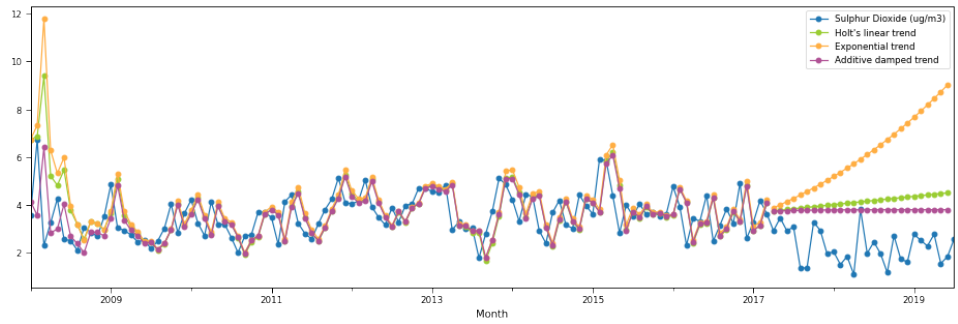

Figura 5.44: Método de Holt lineal para las partículas de dióxido de azufre en la zona de Background. Los valores reales están en azul y la predicción con el modelo aditivio de Holt es en verde, con el modelo exponencial es en naranja y con el modelo aditivo amortiguado es en rosa.

	F.lineal	F.exponencial	F.lineal amortiguada
Óxido nítrico	154,9	24,9	29,51
Dióxido de nitrógeno	14,93	6,97	5,33
Óxidos de nitrógeno	115,58	28,38	27,49
Ozono	40,5	715,32	15,05
Partículas PM10	6.6	4,18	4,06
Partículas PM2.5	8,22	3,86	3,71
Dióxido de azufre	7,92	3,21	3,73

Tabla 5.11: Tabla con el error cuadrático medio generado por cada aproximación con el método de Holt para la zona de Roadisde.

	F.lineal	F.exponencial	F.lineal amortiguada
Óxido nítrico	76,86	6,32	20,97
Dióxido de nitrógeno	12,33	6,98	6,47
Óxidos de nitrógeno	83,36	16,35	26,29
Ozono	45,32	948,77	16,52
Partículas PM10	6.44	4,02	4,03
Partículas PM2.5	4,5	10,36	3,59
Dióxido de azufre	2,04	4,27	1,7

Tabla 5.12: Tabla con el error cuadrático medio generado por cada aproximación con el método de Holt para la zona de Background.

Finalmente para esta sección, vamos a probar el modelo de Holt y Winter en sus dos versiones. Primero vamos a adentrarnos en la versión aditiva y después en la multiplicativa.

Con estos modelos, tenemos en cuenta la estacionalidad de la serie y por ello añadimos un nuevo parámetro, que es de suavización de la estacionalidad γ.

Para cada una de las versiones, aditiva y multiplicatica, vamos a ver dos aproximaciones.

```
1  print('Holt-Winters Seasonal Method Additive - London Mean Roadside')
2  for name, metric in zip(names, metrics):
3    series = pr_LMR[metric]
4    train_pr = series.iloc[:int(len(series) * 0.8)]
5    test_pr = series.iloc[int(len(series) * 0.8):]
6
7    fit1 = ExponentialSmoothing(train_pr, seasonal_periods = 12, trend='
         add', seasonal='add').fit()
8    fcast1 = fit1.forecast(int(len(series) * 0.2)).rename('Additive')
9    mse1 = ((fcast1 - test_pr) ** 2).mean()
10   print('The Root Mean Squared Error of additive trend, additive
         seasonal of '+ 'period season_length={} and a Box-Cox
         transformation {}'.format(4,round(np.sqrt(mse1), 2)))
11
12   fit2 = ExponentialSmoothing(train_pr, seasonal_periods = 12, trend='
         add', seasonal='add', damped=True).fit()
13   fcast2 = fit2.forecast(int(len(series) * 0.2)).rename('Additive+
         damped')
14   mse2 = ((fcast2 - test_pr) ** 2).mean()
15   print('The Root Mean Squared Error of additive damped trend,
         additive seasonal of '+ 'period season_length={} and a Box-Cox
         transformation {}'.format(4,round(np.sqrt(mse2), 2)))
16
17   series.plot(marker='o', color='black', legend=True, figsize=(14, 7))
18   fit1.fittedvalues.plot(style='--', color='red')
19   fcast1.plot(style='--', marker='o', color='red', legend=True)
20   fit2.fittedvalues.plot(style='--', color='green')
21   fcast2.plot(style='--', marker='o', color='green', legend=True)
22
23   plt.show()
```

Entre las líneas 3 y 5 hacemos una copia de los valores de la métrica y al igual que hemos hecho en los demás casos, separamos el conjunto en dos.

Esta vez vamos a hacer solo dos tipos de predicciones, para ambas vamos a usar la función `ExponentialSmoothing`.

En la línea 7 entrenamos nuestro modelo con el conjunto de datos de entrenamiento y establecemos un periodo aditivo de longitud 12 y una tendencia aditiva. Seguido en la línea 8 hacemos la predicción con el conjunto de datos de testeo y calculamos en la línea 9 el error cuadrático medio mostrandolo en la línea 10.

En la línea 12 hacemos nuestro segundo entrenamiento que será igual que la anterior pero usando la versión amortiguada. En la línea 13 realizamos la predicción y calculamos su error en la línea 14.

Finalmente, entre las líneas 17 y 23 mostramos las gráficas de los valores reales junto a estas dos predicciones.

Los resultados obtenidos nos muestran unas gráficas que contienen los valores reales en azul, el modelo aditivo en en verde y el aditivo amoriguado en naranja.

Vamos a ver como se comporta cada una de las partículas con estas aproximaciones y cuales son sus errores, los cuales podemos verlos en la tabla 5.13.
Empezamos estudiando el comportamiento en la zona de Roadside. Antes de ello tenemos que indicar que una vez vistos los gráficos, nos damos cuenta que el método aditivo y el método amortiguado nos dan una aproximación muy parecida de los valores. Por tanto, vamos a explicar como se comportan las predicciones, ya que en ambas tienen el mismo comportamiento, y vemos cual obtiene un mejor error.

- Óxido nítrico: Podemos ver en la figura 5.45 que los resultados de la predicción son superiores a los reales aunque estos tienen un comportamiento muy parecido. Esto es debido a que a partir del año 2.017 hay una tendencia decreciente de los valores que no había ocurrido hasta el momento.
  La forma amortiguada tiene un mejor error cuadrático medio siendo este de 50,99, con respecto al aditivo que asciende a 54,12.

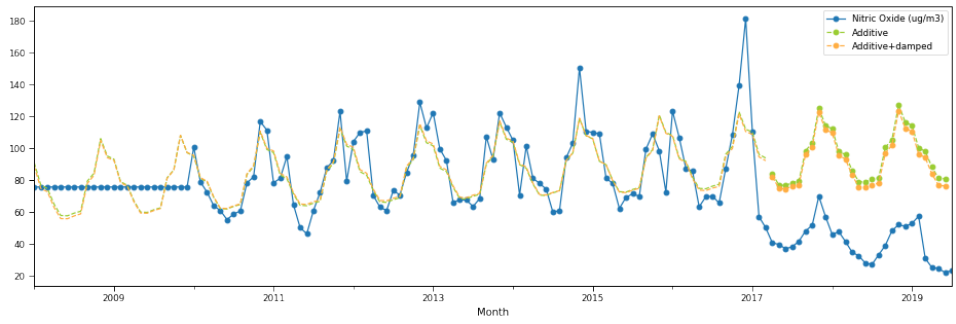

Figura 5.45: Método de Holt-Winter aditivo para las partículas de óxido nítrico en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Dióxido de nitrógeno: Podemos ver en la figura 5.46 que los resultados de la predicción son un poco superiores a los reales y no consiguen ajustarse a los picos de la función.
  El error obtenido con la forma amortiguada es de 9,47 siendo un poco peor que la forma lineal que consigue un error de 9,19.

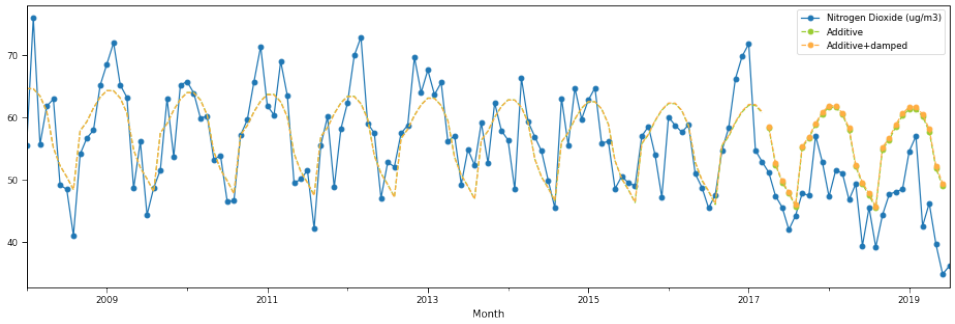

Figura 5.46: Método de Holt-Winter aditivo para las partículas de dióxido de nitrógeno en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Óxidos de nitrógeno: Podemos ver en la figura 5.47 que los resultados de la predicción son superiores a los reales aunque su comportamiento es muy similar.
  La forma amortiguada tiene un mejor error cuadrático medio siendo este de 43,68, con respecto al aditivo que asciende a 45,34.

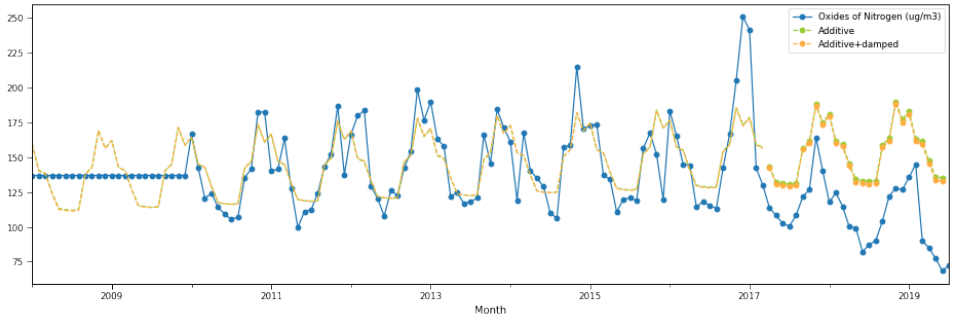

Figura 5.47: Método de Holt-Winter aditivo para las partículas de óxidos de nitrógeno en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Ozono: Podemos ver en la figura 5.48 que los valores de la predicción son bastantes buenos, se ajustan muy bien a los valores reales. El error

que obtenemos con la forma aditiva es de 3,52, al igual superior con respecto a su forma amortiguada que desciende a 3,46.

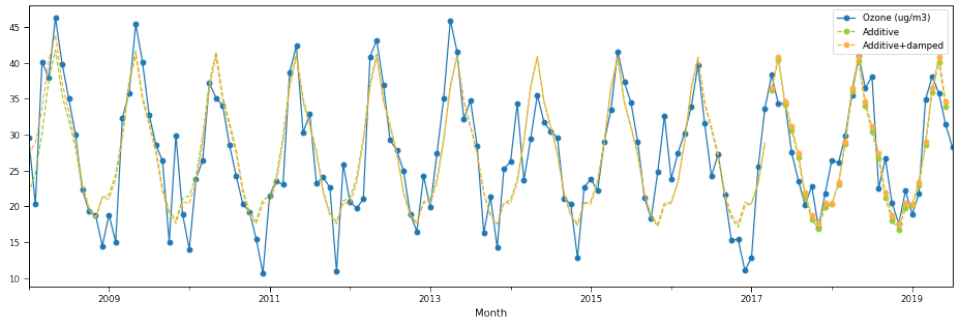

Figura 5.48: Método de Holt-Winter aditivo para las partículas de ozono en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Partículas PM10: Podemos ver en la figura 5.49 que la predicción no es mala, pero tampoco demasiado buena. Le cuesta predecir los valores máximos y mínimos.
  La forma amortiguada tiene un peor error cuadrático medio siendo este de 4,05, con respecto al aditivo que desciende a 3,87.

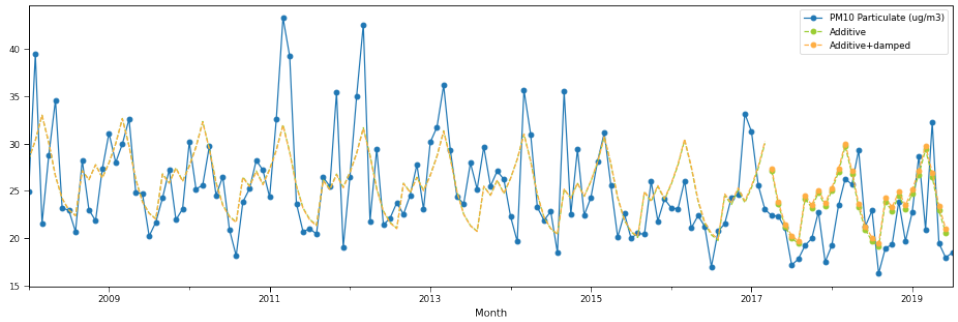

Figura 5.49: Método de Holt-Winter aditivo para las partículas de PM10 en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Partículas PM2.5: Podemos ver en la figura 5.50 que en general la predicción sigue una trazabilidad similar a los valores reales, pero no se ajusta bien a las subidas y bajadas.
  La forma amortiguada tiene un peor error cuadrático medio siendo

este de 3,33, con respecto al aditivo que desciende a 3,2.

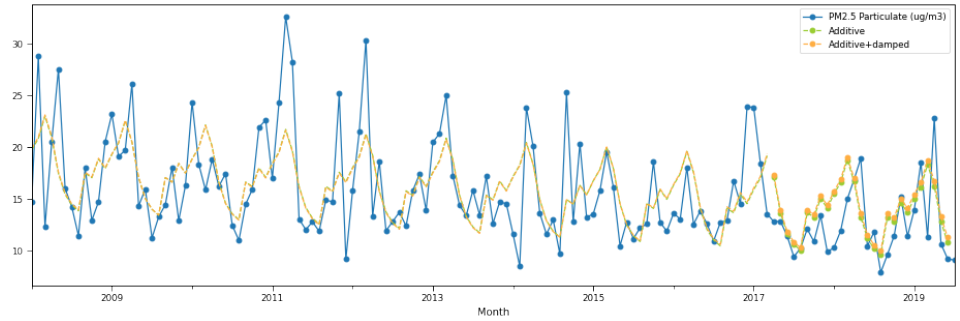

Figura 5.50: Método de Holt-Winter aditivo para las partículas de PM2.5 en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Dióxido de azufre: Podemos ver en la figura 5.51 que este método no se ajusta muy bien a los valores que toma esta partícula. Tenemos que tener en cuenta, que a partir de 2.015 los valores que toma la serie empiezan poco a poco a salirse del rango de valores que estaba tomando esta partícula.
  El error para la forma amortiguada es mejor que la obtenida para la lineal, siendo para el primero de 3,32 y de 3,36 para el segundo.

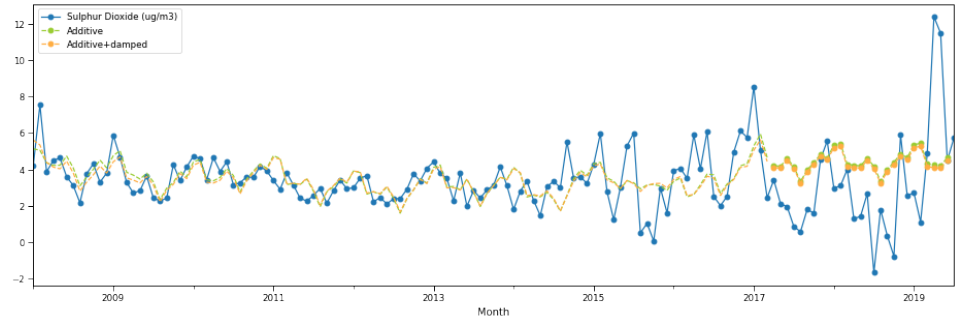

Figura 5.51: Método de Holt-Winter aditivo para las partículas de dióxido de azufre en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

Seguimos con la zona de Background. Antes de ello tenemos que indicar que una vez vistos los gráficos, nos damos cuenta que el método aditivo y el método amortiguado nos dan una aproximación muy parecida de los valores. Por tanto, vamos a explicar como se comportan las predicciones, ya

que en ambas tienen el mismo comportamiento, y vemos cual obtiene un mejor error. Los errores para esta zona los podemos ver en la tabla 5.14

- Óxido nítrico: Podemos ver en la figura 5.52 que los resultados de la predicción son un poco superiores a los reales aunque estos tienen un comportamiento muy parecido.
  La forma amortiguada tiene un mejor error cuadrático medio siendo este de 13,41, con respecto al aditivo que asciende a 14,04.

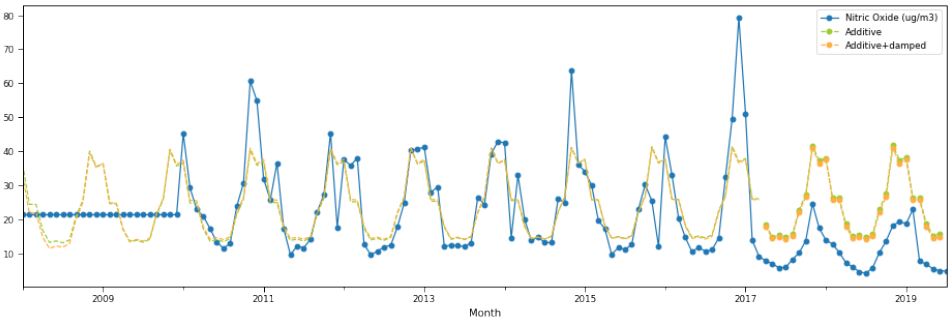

Figura 5.52: Método de Holt-Winter aditivo para las partículas de óxido nítrico en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Dióxido de nitrógeno: Podemos ver en la figura 5.53 que los resultados de la predicción se ajustan al movimiento de los valores reales, aunque varían un poco en los valores concretos.
  El error obtenido con la forma amortiguada es de 3,34 siendo un poco superior al obtenidocon la forma lineal que consigue un error de 2,91.

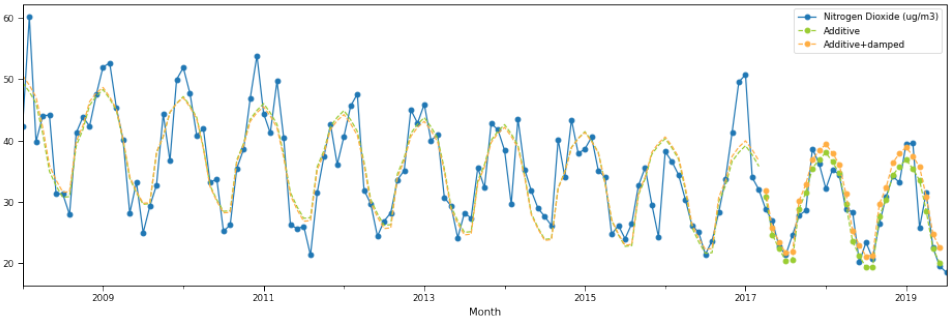

Figura 5.53: Método de Holt-Winter aditivo para las partículas de dióxido de nitrógeno en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Óxidos de nitrógeno: Podemos ver en la figura 5.54 que los resultados de la predicción son un poco superiores a los reales aunque su trazabilidad es muy similar.
  La forma amortiguada tiene un peor error cuadrático medio siendo este de 15,15, con respecto al aditivo que desciende a 14,33.

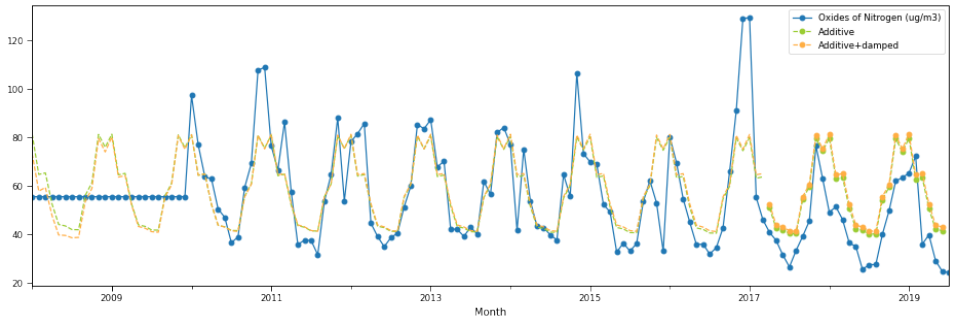

Figura 5.54: Método de Holt-Winter aditivo para las partículas de óxidos de nitrógeno en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Ozono: Podemos ver en la figura 5.55 que los valores de la predicción son bastantes buenos, se ajustan muy bien a la trazabilidad y se asemejan a los valores reales. El error que obtenemos con la forma aditiva es de 5,89 siendo esta superior con respecto a su forma amortiguada que asscience a 6,39.

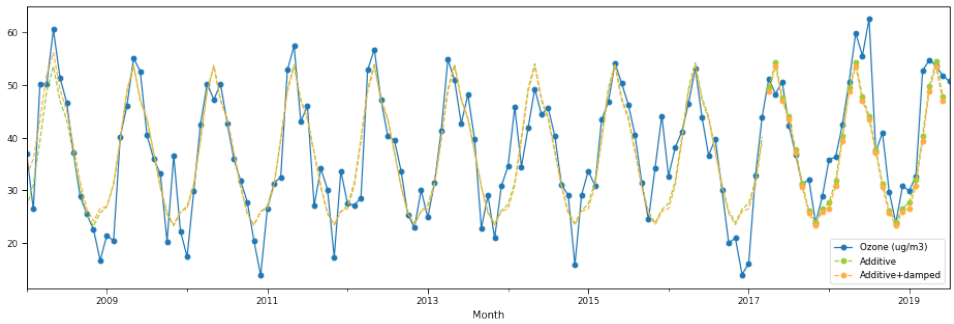

Figura 5.55: Método de Holt-Winter aditivo para las partículas de ozono en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Partículas PM10: Podemos ver en la figura 5.56 que la predicción no es demasiado buena. Le cuesta predecir los valores, aunque intenta

ajustarse al movimiento de los datos reales.
La forma amortiguada tiene un peor error cuadrático medio siendo este de 3,31, con respecto al aditivo que desciende a 3,27.

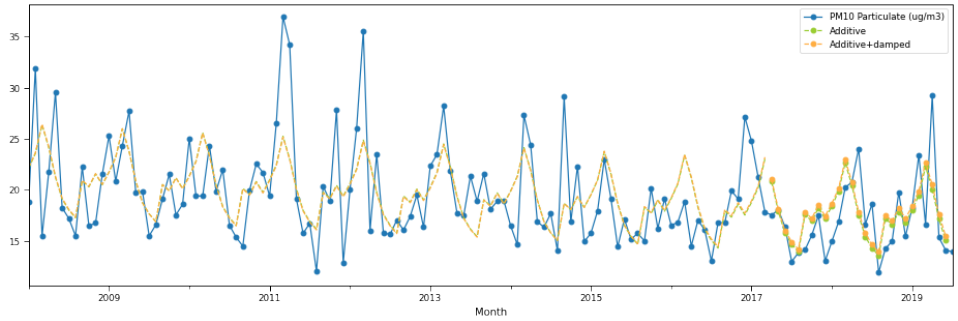

Figura 5.56: Método de Holt-Winter aditivo para las partículas PM10 en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Partículas PM2.5: Podemos ver en la figura 5.57 que en general la predicción sigue una trazabilidad similar a los valores reales, parece la trazabilidad correcta pero con un poco de anticipo.
  La forma amortiguada tiene un peor error cuadrático medio siendo este de 3,06, con respecto al aditivo que desciende a 3,03.

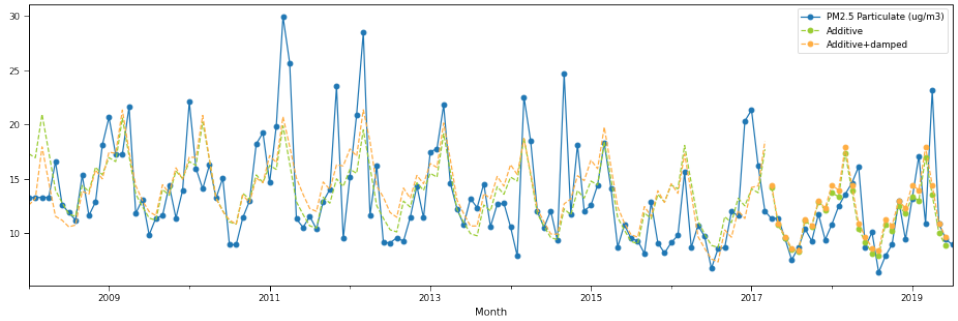

Figura 5.57: Método de Holt-Winter aditivo para las partículas PM2.5 en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

- Dióxido de azufre: Podemos ver en la figura 5.58 que este método predice los valores por encima de los reales.
  El error para la forma amortiguada es mejor que la obtenida para la lineal, siendo para el primero de 1,48 y de 1,57 para el segundo.

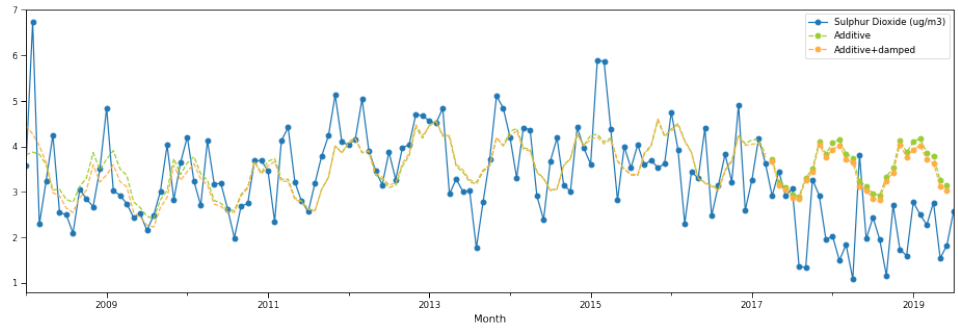

Figura 5.58: Método de Holt-Winter aditivo para las partículas de dióxido de azufre en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en naranja.

	F.aditiva	F.aditiva amortiguada
Óxido nítrico	54,12	50,99
Dióxido de nitrógeno	9,19	9,47
Óxidos de nitrógeno	45,34	43,68
Ozono	3,52	3,46
Partículas PM10	3,87	4,05
Partículas PM2.5	3,2	3,33
Dióxido de azufre	3,36	3,32

Tabla 5.13: Tabla con el error cuadrático medio generado por cada aproximación con el método de Holt-Winter lineal para la zona de Roadside.

	F.aditiva	F.aditiva amortiguada
Óxido nítrico	14,04	13,41
Dióxido de nitrógeno	2,91	3,34
Óxidos de nitrógeno	14,33	15,56
Ozono	5,89	6,39
Partículas PM10	3,27	3,31
Partículas PM2.5	3,03	3,06
Dióxido de azufre	1,57	1,48

Tabla 5.14: Tabla con el error cuadrático medio generado por cada aproximación con el método de Holt-Winter lineal para la zona de Background.

A continuación vemos el método multiplicativo, que es exactamente igual al que acabamos de explicar pero cambiando el periodo de longitud 12 a uno multiplicativo.

```
print('Holt-Winters Seasonal Method Multiplicative - London Mean
    Roadside')
for name, metric in zip(names, metrics):
  series = pr_LMR[metric]
  train_pr = series.iloc[:int(len(series) * 0.8)]
  test_pr = series.iloc[int(len(series) * 0.8):]

  fit1 = ExponentialSmoothing(train_pr, seasonal_periods = 12, trend='
      add', seasonal='mul').fit()
  fcast1 = fit31.forecast(int(len(series) * 0.2)).rename('
      Multiplicative')
  mse1 = ((fcast1 - test_pr) ** 2).mean()
  print('The Root Mean Squared Error of additive trend, multiplicative
       seasonal of '+ 'period season_length={}'.format(12,round(np.
      sqrt(mse1), 2)))

  fit2 = ExponentialSmoothing(train_pr, seasonal_periods = 12, trend='
      add', seasonal='mul', damped=True).fit()
  fcast2 = fit2.forecast(int(len(series) * 0.2)).rename('
      Multiplicative+damped')
  mse2 = ((fcast2 - test_pr) ** 2).mean()
  print('The Root Mean Squared Error of additive damped trend,
      multiplicative seasonal of '+ 'period season_length={}'.format
      (12,round(np.sqrt(mse2), 2)))

  series.plot(marker='o', color='black', legend=True, figsize=(14, 7))
  fit1.fittedvalues.plot(style='--', color='red')
  fcast1.plot(style='--', marker='o', color='red', legend=True)
  fit2.fittedvalues.plot(style='--', color='green')
  fcast2.plot(style='--', marker='o', color='green', legend=True)

  plt.show()
```

Los resultados obtenidos nos muestran unas gráficas que contienen los valores reales en azul, el modelo multiplicativo en en verde y el multipicativo amoriguado en rosa.

Vamos a ver como se comporta cada una de las partículas con estas aproximaciones y cuales son sus errores, los cuales podemos verlos en la tabla 5.15.
Empezamos estudiando el comportamiento en la zona de Roadside. Antes de ello tenemos que indicar que una vez vistos los gráficos, nos damos cuenta que el método multiplicativo y el método amortiguado nos dan una aproximación muy parecida de los valores. Por tanto, vamos a explicar como se comportan las predicciones, ya que en ambas tienen el mismo comportamiento, y vemos cual obtiene un mejor error.

- Óxido nítrico: Podemos ver en la figura 5.59 que los resultados de la predicción son superiores a los reales. En general, tiene una trazabilidad similar, aunque no respete mucho los máximos y los mínismos.

La forma amortiguada tiene un mejor error cuadrático medio siendo este de 45,97, con respecto al multiplicativo que asciende a 46,14.

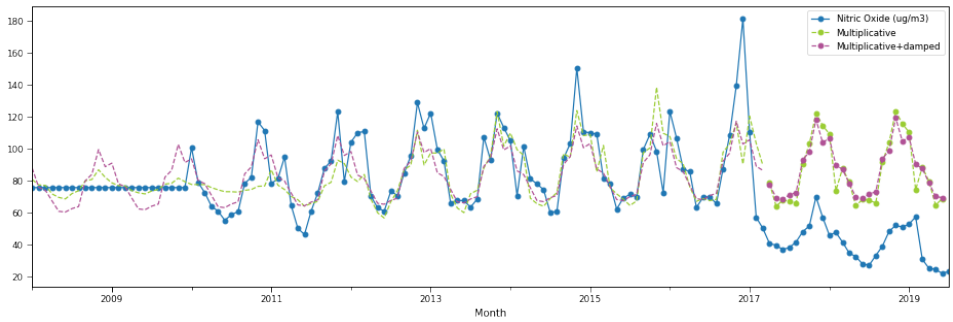

Figura 5.59: Método de Holt-Winter multiplicativo para las partículas de óxido nítrico en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Dióxido de nitrógeno: Podemos ver en la figura 5.60 que los resultados de la predicción son un poco superiores a los reales y no consiguen ajustarse a los picos de la función, ya que la predicción realiza una especie de curvatura.
  El error obtenido con la forma amortiguada es de 9,38 siendo un poco peor que la forma multiplicativa que consigue un error de 9,1.

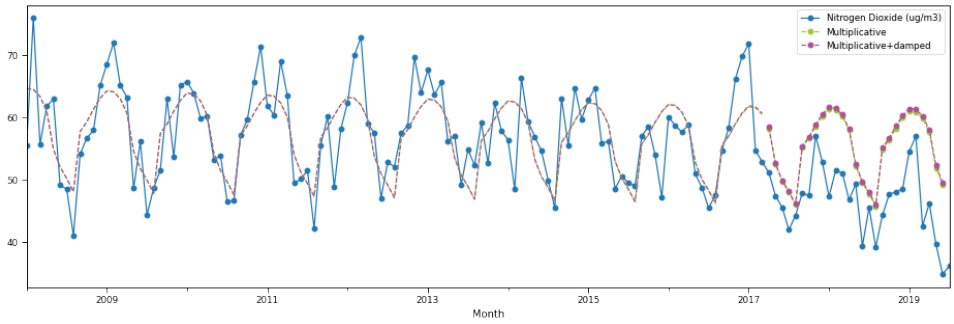

Figura 5.60: Método de Holt-Winter multiplicativo para las partículas de dióxido de nitrógeno en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Óxidos de nitrógeno: Podemos ver en la figura 5.61 que los resultados de la predicción son superiores a los reales aunque su trazabilidad es muy similar.

La forma amortiguada tiene un mejor error cuadrático medio siendo este de 39,99, con respecto al multiplicativo que asciende a 41,36.

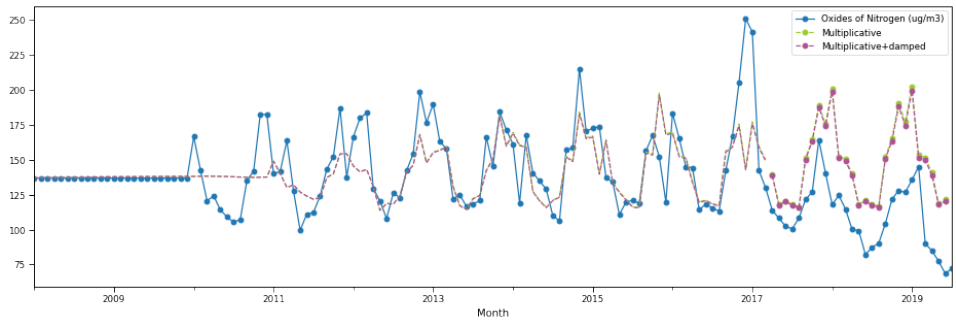

Figura 5.61: Método de Holt-Winter multiplicativo para las partículas de óxidos de nitrógeno en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Ozono: Podemos ver en la figura 5.62 que los valores de la predicción son bastantes buenos, se ajustan muy bien a los valores reales. El error que obtenemos con la forma multiplicativa es de 3,54, un poco superior con respecto a su forma amortiguada que desciende a 3,47.

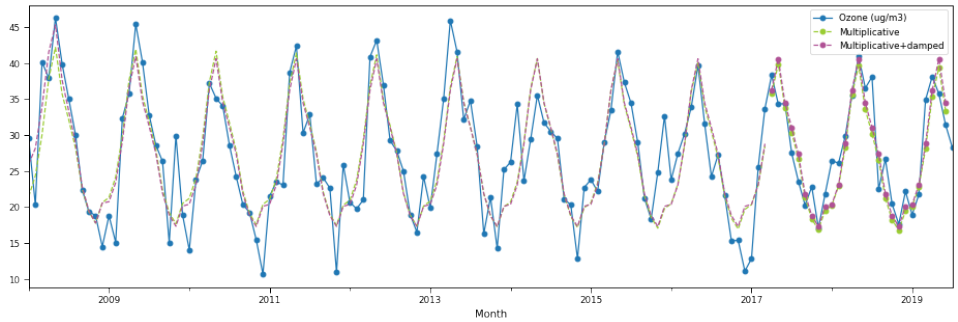

Figura 5.62: Método de Holt-Winter multiplicativo para las partículas de ozono en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Partículas PM10: Podemos ver en la figura 5.63 que la predicción no es mala, pero tampoco demasiado buena. Le cuesta predecir los valores máximos y mínimos.
  La forma amortiguada tiene un peor error cuadrático medio siendo este de 4,44, con respecto al aditivo que desciende a 4,27.

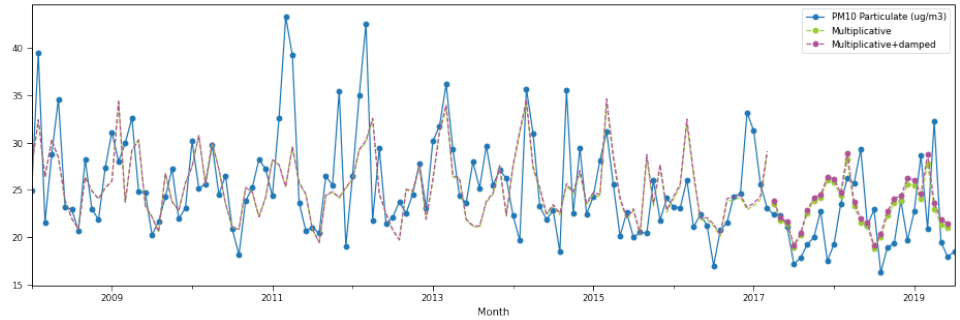

Figura 5.63: Método de Holt-Winter multiplicativo para las partículas PM10 en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Partículas PM2.5: Podemos ver en la figura 5.64 que en general la predicción sigue una trazabilidad parecida a los valores reales, pero no se ajusta bien a las subidas y bajadas.
  La forma amortiguada tiene un peor error cuadrático medio siendo este de 3,09, con respecto al aditivo que desciende a 3,02.

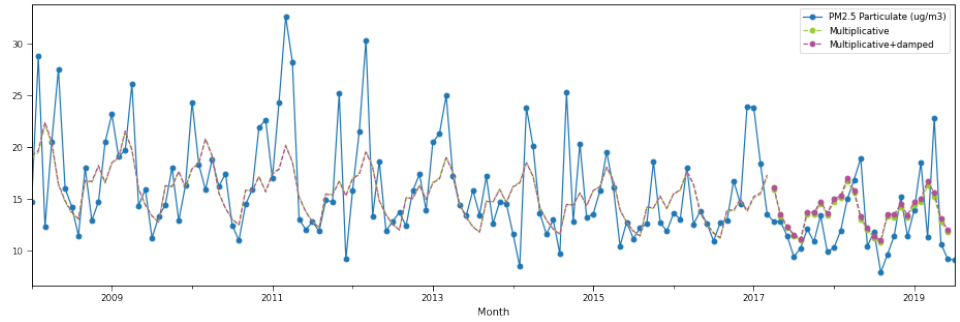

Figura 5.64: Método de Holt-Winter multiplicativo para las partículas PM2.5 en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Dióxido de azufre: Podemos ver en la figura 5.65 que este método no se ajusta muy bien a los valores que toma esta partícula.
  Los errores para ambas formas son iguales, siendo ambos de 3,15.

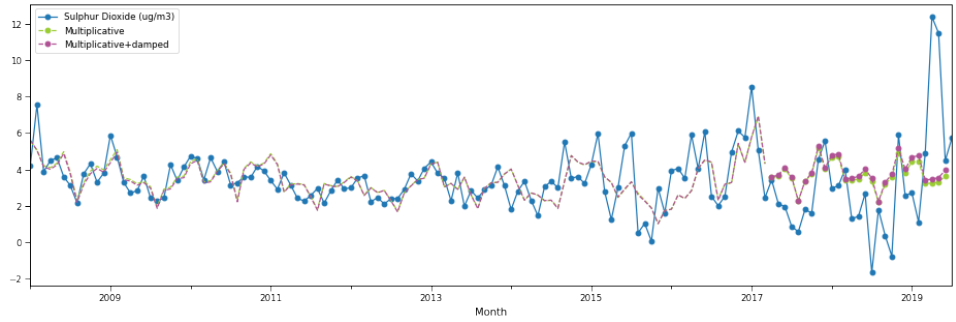

Figura 5.65: Método de Holt-Winter multiplicativo para las partículas de dióxido de azufre en la zona de Roadside. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

Seguimos con la zona de Background. Antes de ello tenemos que indicar que una vez vistos los gráficos, nos damos cuenta que el método multiplicativo y el método amortiguado nos dan una aproximación muy parecida de los valores. Por tanto, vamos a explicar como se comportan las predicciones, ya que en ambas tienen el mismo comportamiento, y vemos cual obtiene un mejor error. Los errores para esta zona los podemos ver en la tabla 5.16

- Óxido nítrico: Podemos ver en la figura 5.66 que los resultados de la predicción son un poco superiores a los reales aunque estos tienen un comportamiento muy parecido.
  La forma amortiguada tiene un peor error cuadrático medio siendo este de 9,43, con respecto al multiplicativo que desciende a 9,08.

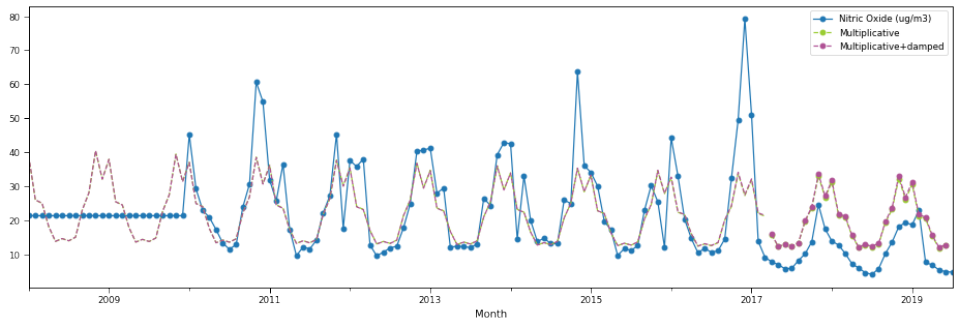

Figura 5.66: Método de Holt-Winter multiplicativo para las partículas de óxido nítrico en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Dióxido de nitrógeno: Podemos ver en la figura 5.67 que los resultados

de la predicción se ajustan al movimiento de los valores reales y su error de predicción no es demasiado grande.
El error obtenido con la forma amortiguada es de 3,11 siendo un poco superior al obtenidoc on la forma multiplicativa que consigue un error de 2,78.

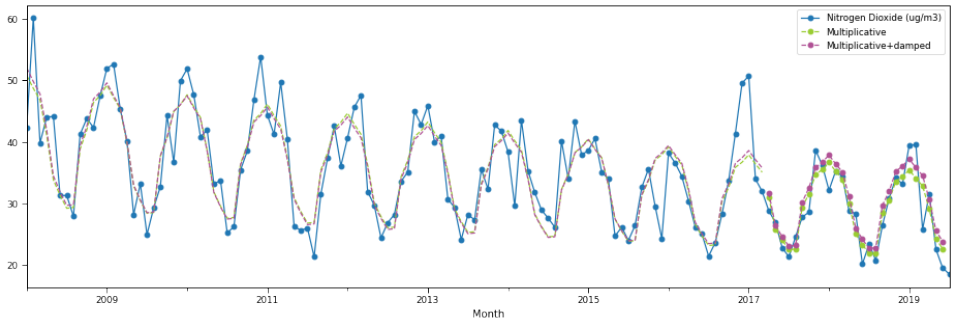

Figura 5.67: Método de Holt-Winter multiplicativo para las partículas de dióxido de nitrógeno en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Óxidos de nitrógeno: Podemos ver en la figura 5.68 que los resultados de la predicción se ajustan a la trazabilidad de los reales aunque son en general, un poco superiores.
  La forma amortiguada tiene un peor error cuadrático medio siendo este de 10,65, con respecto al multiplicativo que desciende a 14,33.

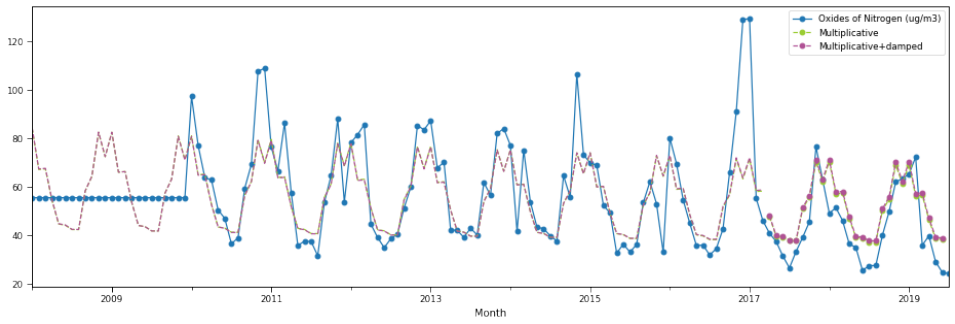

Figura 5.68: Método de Holt-Winter multiplicativo para las partículas de óxidod de nitrógeno en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Ozono: Podemos ver en la figura 5.69 que los valores de la predicción

son bastantes buenos, se ajustan bien a la trazabilidad y se asemejan a los valores reales. El error que obtenemos con la forma multiplicativa es de 6,18 siendo esta mejor con respecto a su forma amortiguada que asciende a 6,44.

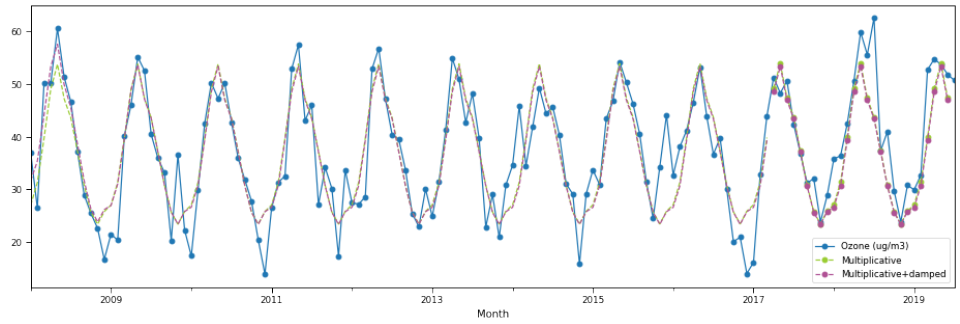

Figura 5.69: Método de Holt-Winter multiplicativo para las partículas de ozono en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Partículas PM10: Podemos ver en la figura 5.70 que la predicción no es demasiado buena. Le cuesta predecir los valores, aunque intenta ajustarse al movimiento de los datos reales.
  La forma amortiguada tiene un mejor error cuadrático medio siendo este de 3,23, con respecto al multiplicativo que asciende a 3,25.

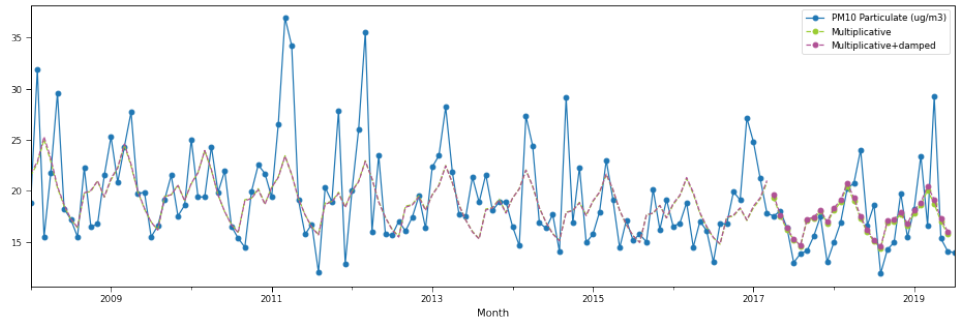

Figura 5.70: Método de Holt-Winter multiplicativo para las partículas PM10 en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Partículas PM2.5: Podemos ver en la figura 5.71 que en general la predicción no es muy buena, aunque los valores predichos se encuentran

dentro del rango de valores reales.
La forma amortiguada tiene un peor error cuadrático medio siendo este de 2,89, con respecto al multiplicativo que desciende a 2,9.

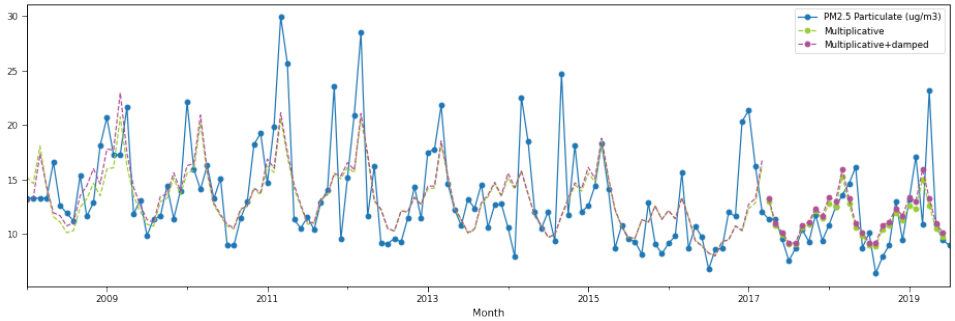

Figura 5.71: Método de Holt-Winter multiplicativo para las partículas PM2.5 en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

- Dióxido de azufre: Podemos ver en la figura 5.72 que este método predice los valores por encima de los reales.
  El error para la forma amortiguada es mejor que la obtenida para la lineal, siendo para el primero de 1,47 y de 1,56 para el segundo.

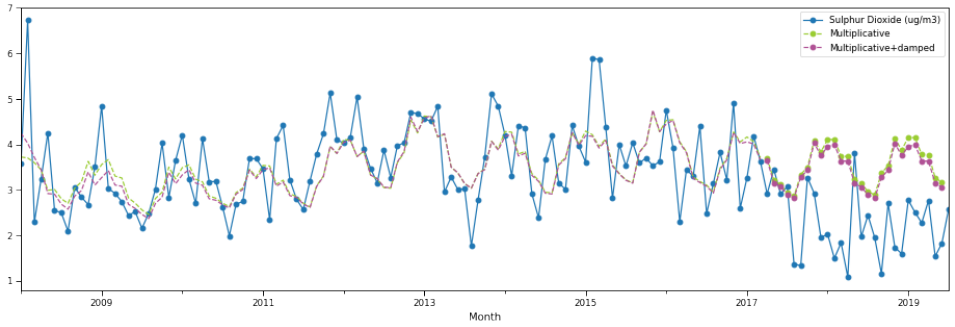

Figura 5.72: Método de Holt-Winter multiplicativo para las partículas de dióxido de azufre en la zona de Background. Los valores reales están en azul, la predicción con el modelo aditivio es en verde y la predicción con el aditivo amortiguado es en rosa.

	F.multiplicativa	F.multiplicativa amortiguada
Óxido nítrico	46,14	45,97
Dióxido de nitrógeno	9,1	9,39
Óxidos de nitrógeno	41,36	39,9
Ozono	3,54	3,47
Partículas PM10	4,27	4,44
Partículas PM2.5	3,02	3,09
Dióxido de azufre	3,15	3,15

Tabla 5.15: Tabla con el error cuadrático medio generado por cada aproximación con el método de Holt-Winter multiplicativo para la zona de Roadside.

	F.multiplicativa	F.multiplicativa amortiguada
Óxido nítrico	9,08	9,43
Dióxido de nitrógeno	2,78	3,11
Óxidos de nitrógeno	10,2	10,65
Ozono	6,18	6,44
Partículas PM10	3,25	3,23
Partículas PM2.5	2,9	2,89
Dióxido de azufre	1,56	1,47

Tabla 5.16: Tabla con el error cuadrático medio generado por cada aproximación con el método de Holt-Winter multiplicativo para la zona de Background.

5.3. ARIMA

Una vez vista la descomposición estacional y el alisamiento exponencial, pasamos a utilizar el método de predicción clásico ARIMA.

En este caso no vamos a hacer un bucle que recorra todas las métricas y obtengamos los resutlados de todos con una sola ejecución. Vamos a hacer un estudio separado para cada métrica. Vamos a ver como lo haríamos para dióxido de nitrógeno.

```
series= pr_LMR['Nitrogen Dioxide (ug/m3)']
stepwise_fit = auto_arima(series, start_p = 1, start_q = 1,
                          max_p = 5, max_q = 5, m = 12,
                          start_P = 0, seasonal = True,
                          d = None, D = 1, trace = True,
                          error_action ='ignore',
```

```
7                               suppress_warnings = True,
8                               stepwise = True)
9 stepwise_fit.summary()
```

Lo primero que haremos es guardar en la variable `series` los valores de la métrica con la que vamos a trabajar.
A continuación, vamos a usar la función `auto_arima` a la cual le vamos a pasar la serie y vamos a pasarle los valores que vienen por defecto para la función.
Esta función se utiliza para encontrar los valores óptimos para usar en la función ARIMA.

Los resultados que se obtienen al ejecutar la función los podemos visualizar ejecutando la línea 9.

```
1  train_pr = series.iloc[:int(len(series) * 0.8)]
2  test_pr = series.iloc[int(len(series) * 0.8):]
3
4  model = SARIMAX(train_pr,
5                  order = (1, 0, 1),
6                  seasonal_order =(0, 1, 1, 12))
7
8  result = model.fit()
9  rm = result.summary()
10
11 print(rm)
12
13 start = len(train_pr)
14 end = len(train_pr) + len(test_pr) -1
15
16 predictions = result.predict(start, end, typ = 'levels').rename("
       Predictions")
17
18 predictions.plot(legend = True)
19 test_pr.plot(legend = True)
20
21 print('MSE:')
22 mean_squared_error(test_pr, predictions)
```

Como hemos hecho anteriormente vamos a dividir la serie en dos. Uno de entrenamiento con el 80 % de los datos y otro de prueba con el 20 % restante de los datos.

En la línea 4, vamos a usar la función `SARIMAX` pasandole como parámetros el conjunto de datos de entrenamiento, el orden (p, d, q) y el orden estacional (P, D, Q, s). Vamos a usar para estos parámetros los óptimos que nos han salido en la ejecución anterior.

A continuación, en la línea 8 entrenamos el modelo con la función `fit` y realizamos la predicción en la línea 16 usando la función `predict` del modelo de ARIMA. A esta función le pasamos como parámetros el índice donde se

inicia la previsión, el índice en el que termina la predicción y como tipo le pasamos "levels"para que prediga los níveles de las variables.

Finalmente, dibujamos en las líneas 18 y 19 las gráficas de los valores reales y los predichos por el modelo. En las líneas 21 y 22 calculamos el error cuadrático medio y lo mostramos.

Ahora vamos a ver que resultados nos da este método. Como ya hemos hecho anteriormente, vamos a ver por separado ambas zonas y partículas. Primero vemos la zona de Roadside, cuya tabla con los errores cuadráticos medios podemos verla en la tabla 5.17 junto con los errores para la zona de Background.

- Óxido nítrico: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,1,1)(2,1,0)[12].
  Si observamos la figura 5.73 vemos que los valores que predice el modelo no se corresponden con los valores reales.
  Podemos observar como la trazabilidad de ambas gráficas es similar, sin embargo los valores predichos se encuentran muy por encima de los reales.
  El error que genera esta predicción es de 17,66.

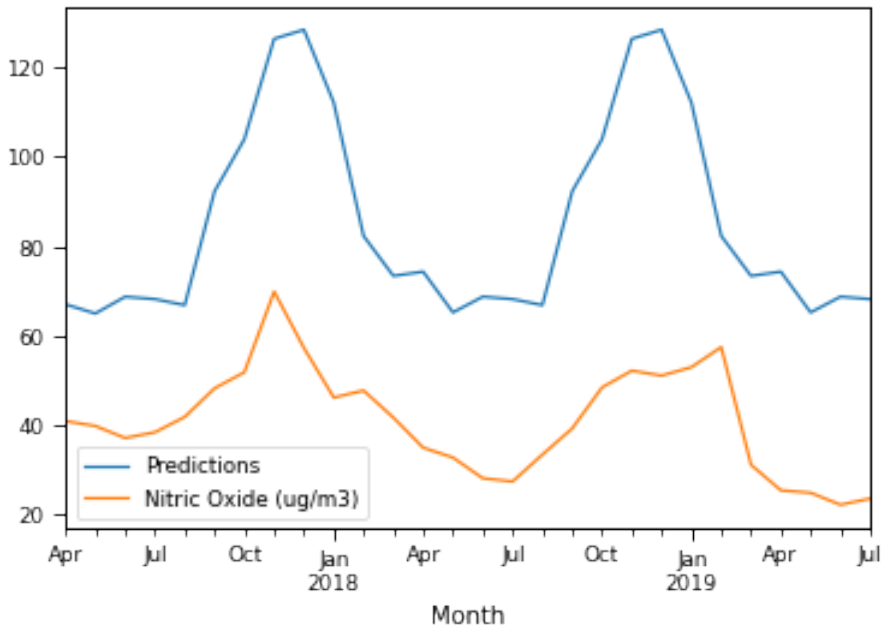

Figura 5.73: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de óxido nítrico y la zona de Roadside. Los valores reales son en naranja y la predicción en azul.

- Dióxido de nitrógeno: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,1)(0,1,1)[12].
  Si observamos la figura 5.74 vemos que los valores que predice el modelo no se corresponden con los valores reales.
  La trazabilidad de ambas gráficas no se parece demasiado y los valores reales se encuentran por debajo de los predichos. Vemos que no es capaz de imitar algunos de los máximos o mínimos, si no todo lo contrario, cuando hay un mínimo lo ve como un máximo. El error que genera esta predicción es de 10,13.

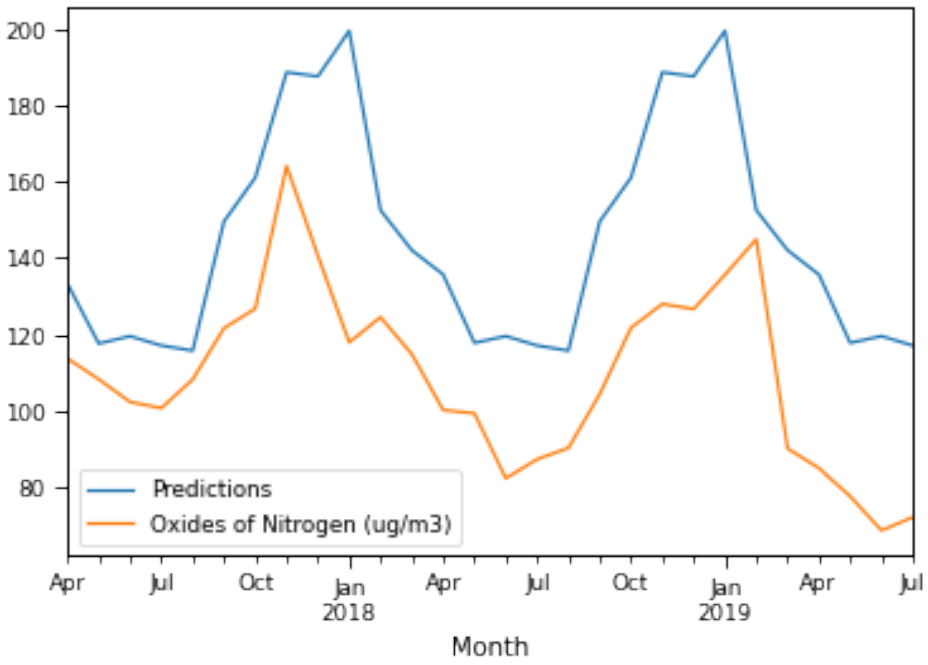

Figura 5.74: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de dióxido de nitrógeno y la zona de Roadside. Los valores reales son en naranja y la predicción en azul.

- Óxidos de nitrógeno: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,2)(0,1,1)[12].
  Si observamos la figura 5.75 vemos que ocurre algo muy parecido a lo ocurrido con la partícula de óxido nítrico.
  Los valores que predice el modelo no se corresponden con los valores reales.
  Podemos observar como la trazabilidad de ambas gráficas es similar, sin embargo los valores predichos se encuentran muy por encima de los reales. El error que genera esta predicción es de 47,07.

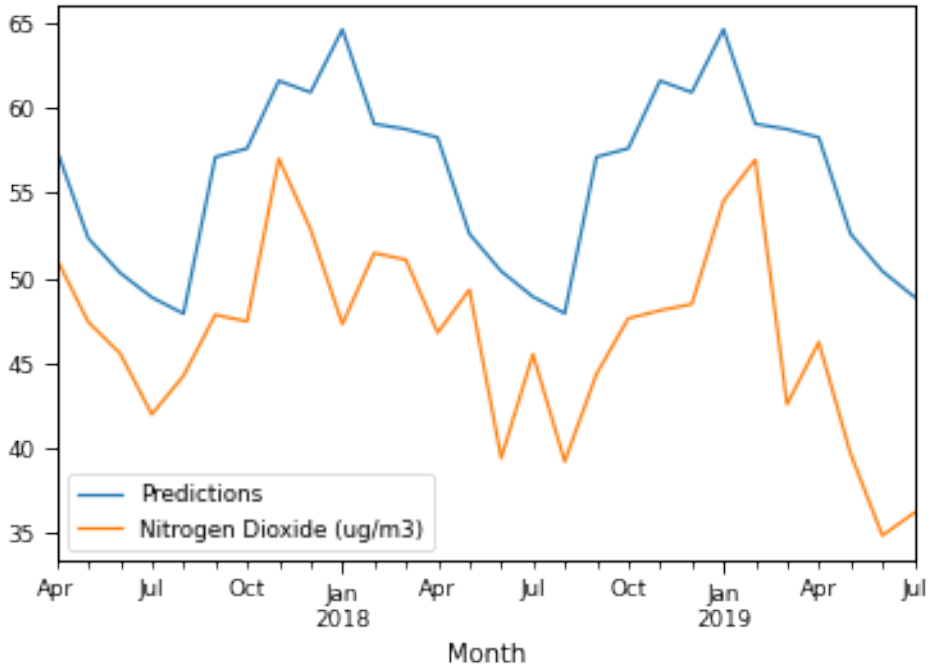

Figura 5.75: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de óxidos de nitrógeno y la zona de Roadside. Los valores reales son en naranja y la predicción en azul.

- Ozono: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,0)(2,1,0)[12].
  Si observamos la figura 5.76 vemos que la predicción se aproxima bastante a la realidad.
  El comportamiento es muy parecido y la diferencia entr la mayoría de los valores no es muy grande. El error obtenido para esta predicción es 3,42.

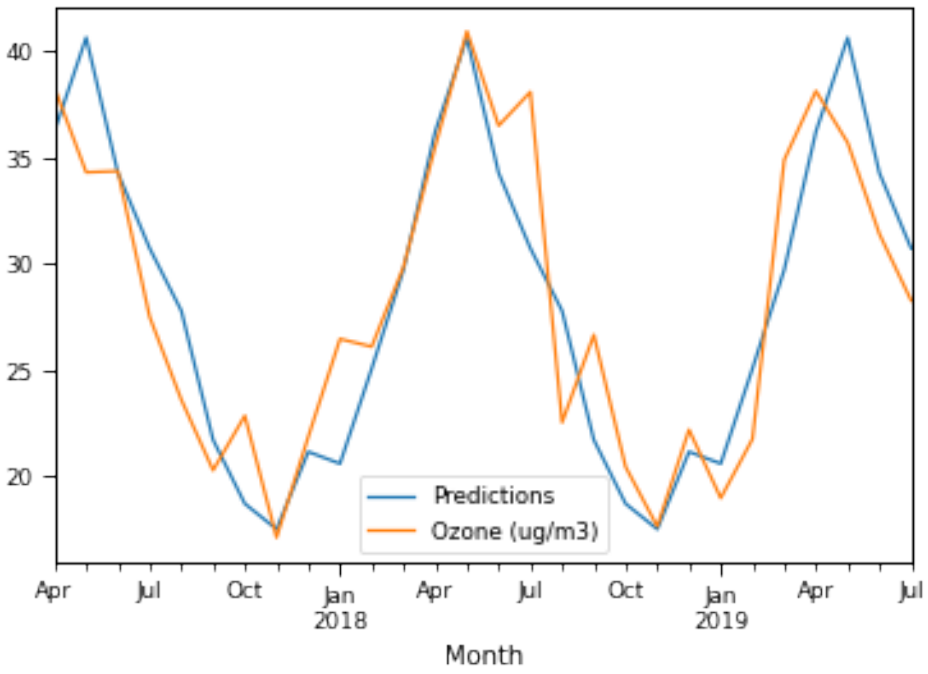

Figura 5.76: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de ozono y la zona de Roadside. Los valores reales son en naranja y la predicción en azul.

- Partículas PM10: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,1)(0,1,1)[12].
  Si observamos la figura 5.77 vemos que la predicción no se parece demasiado a la realidad.
  Ni la trazabilidad ni los valores predichos son buenos. El error obtenido para estra predicción es 4,64.

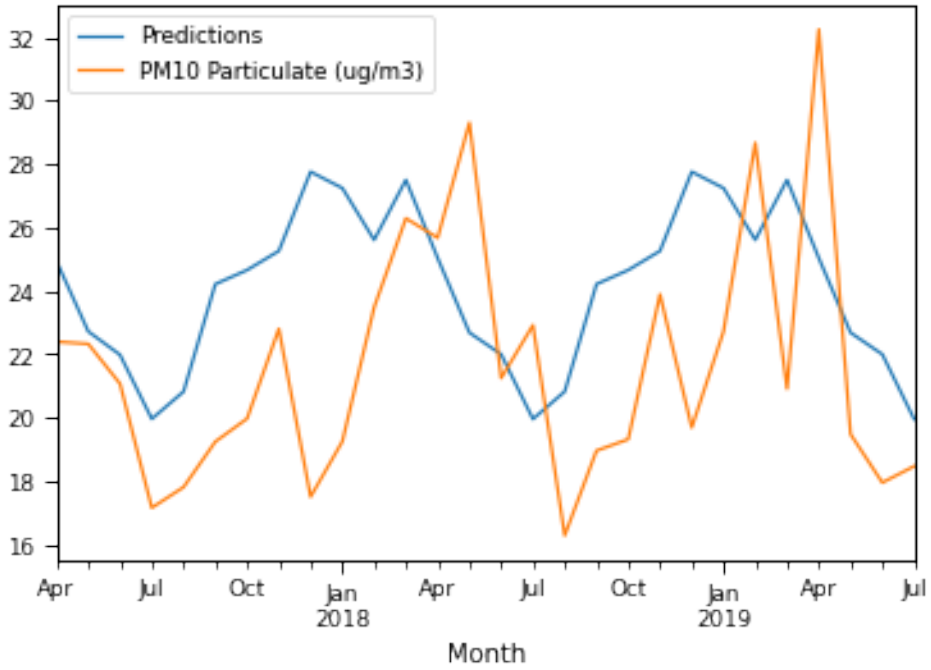

Figura 5.77: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas PM10 y la zona de Roadside. Los valores reales son en naranja y la predicción en azul.

- Partículas PM2.5: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,1)(0,1,1)[12].
  Si observamos la figura 5.78 vemos que los valores que predice el modelo no se corresponden con los valores reales.No se realiza una buena de predicción de los valores ni de la trazabilidad. El error que se ha obtenido para esta gráfica es de 4,41.

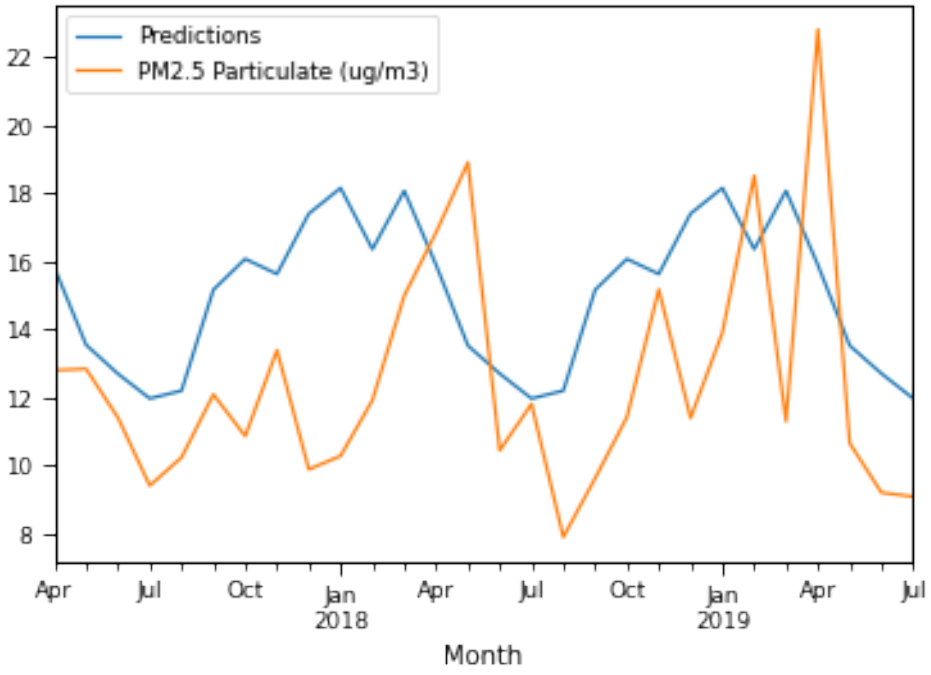

Figura 5.78: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas PM2.5 y la zona de Roadside. Los valores reales son en naranja y la predicción en azul.

- Dióxido de azufre: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,1)(0,1,1)[12].
  Si observamos la figura 5.79 vemos que los valores que predice el modelo tampoco se corresponden con los valores reales.
  El gráfico de predicción no tiene nada que ver con el real. El error que se ha obtenido para esta gráfica es de 3,09.

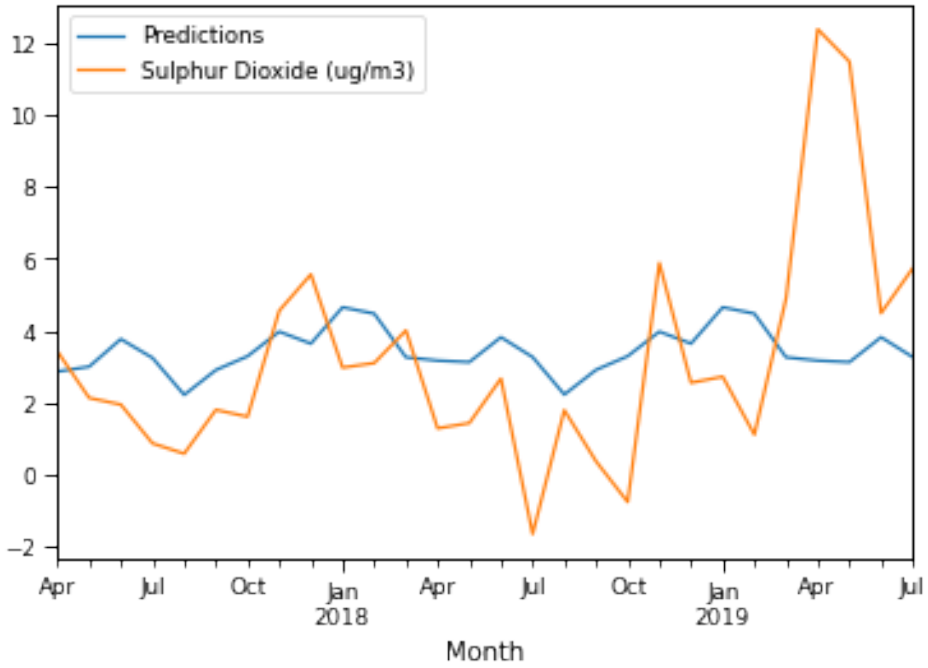

Figura 5.79: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de dióxido de azufre y la zona de Roadside. Los valores reales son en naranja y la predicción en azul.

Vamos a ver ahora que ocurre con la zona de Background. La tabla con el error cuadrático medio podemos verla en la tabla 5.17.

- Óxido nítrico: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,0)(2,1,[1,2])[12].
  Si observamos la figura 5.80 vemos que los valores que predice el modelo no se corresponden con los valores reales.
  Podemos observar como la trazabilidad de ambas gráficas es similar, sin embargo los valores predichos se encuentran muy por encima de los reales. El error que genera esta predicción es de 21,43.

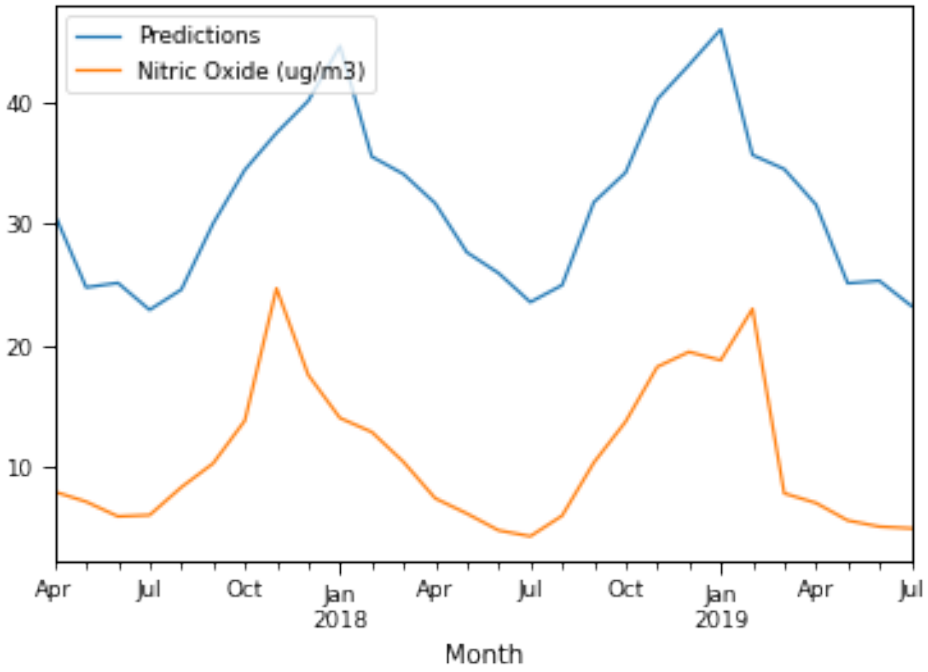

Figura 5.80: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de óxido nítrico y la zona de Background. Los valores reales son en naranja y la predicción en azul.

- Dióxido de nitrógeno: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(0,0,0)(0,1,[1])[12].
  Si observamos la figura 5.81 vemos que la predicción es un poco mejor en esta zona que en la anterior. La trazabilidad es similar, aunque no respete algunos extremos.
  El error que genera esta predicción es de 4,58.

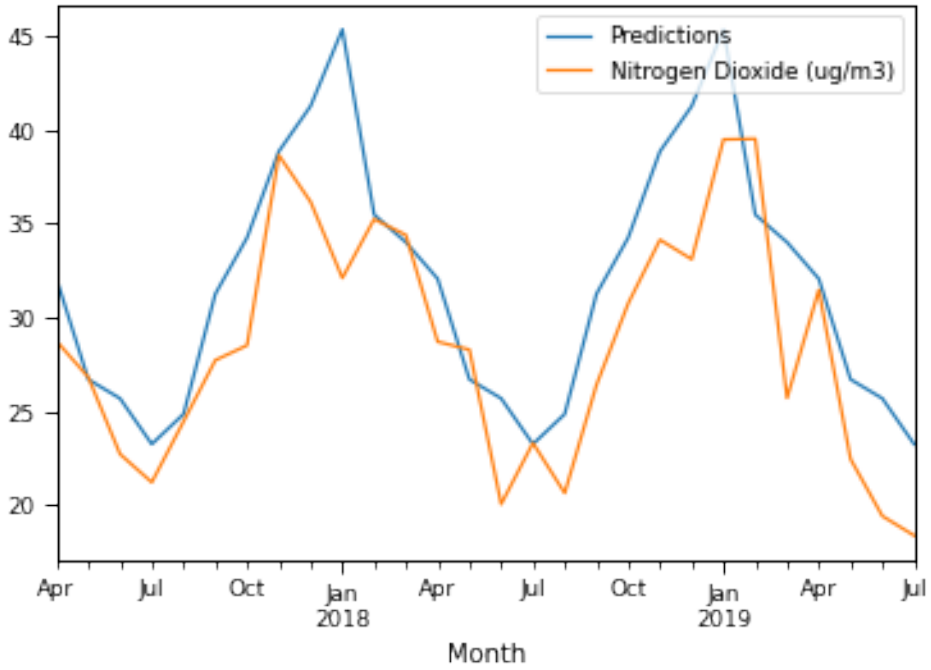

Figura 5.81: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de dióxido de nitrógeno y la zona de Background. Los valores reales son en naranja y la predicción en azul.

- Óxidos de nitrógeno: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,0)(2,1,[1])[12].
  Si observamos la figura 5.82 podemos ver que los valores de la predicción están un poco por encima pero no tanto como con el óxido nítrico. Los valores predichos no se ajusta demasiado bien a la realidad. El error que genera esta predicción es de 13,97.

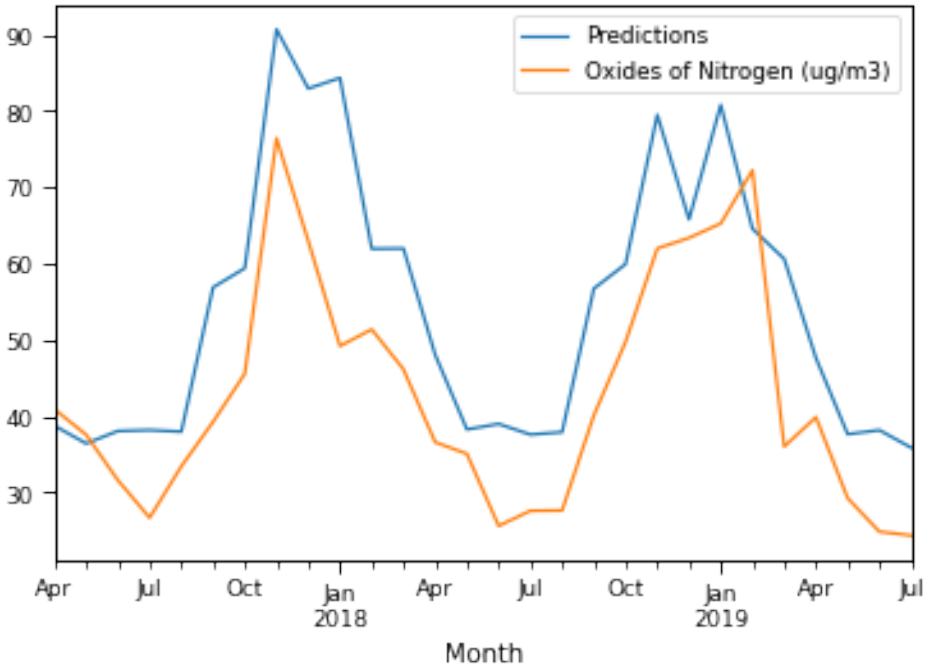

Figura 5.82: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de óxidos de nitrógeno y la zona de Background. Los valores reales son en naranja y la predicción en azul.

- Ozono: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,0)(2,1,0)[12].
  Si observamos la figura 5.83 vemos que la predicción se parece un poco a la realidad. Algunos valores si coinciden o son muy similares, sin embargo en otro hay una gran diferencia.
  El error obtenido para esta predicción es 6,85.

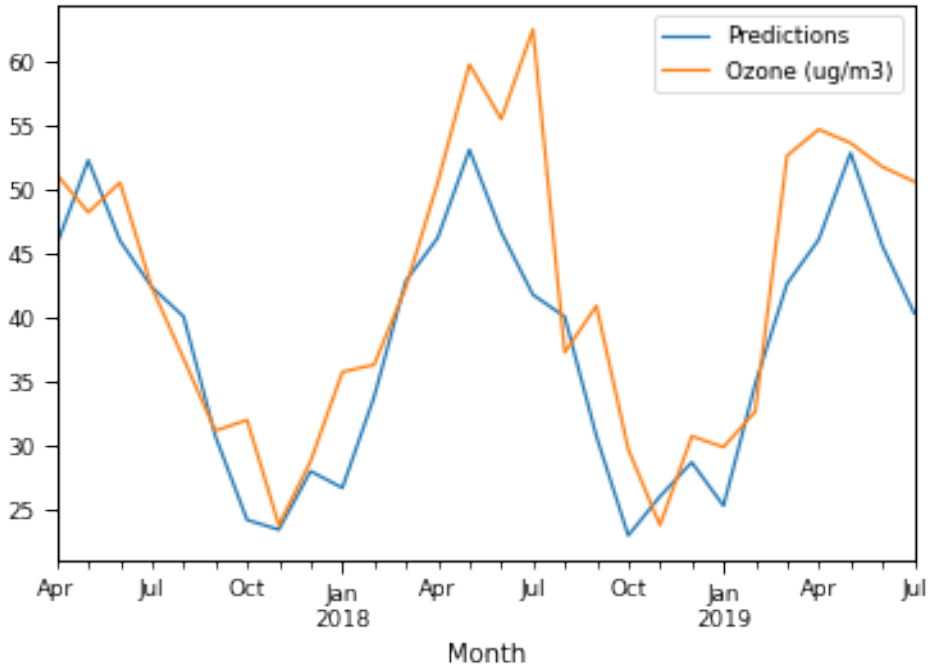

Figura 5.83: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de ozono y la zona de Background. Los valores reales son en naranja y la predicción en azul.

- Partículas PM10: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(0,0,0)(2,1,0)[12].
  Si observamos la figura 5.84 vemos que la predicción se aleja bastante de la realidad.
  La trazabilidad es muy diferente y hay pocos valores que coincidan. El error obtenido para estra predicción es 4,56.

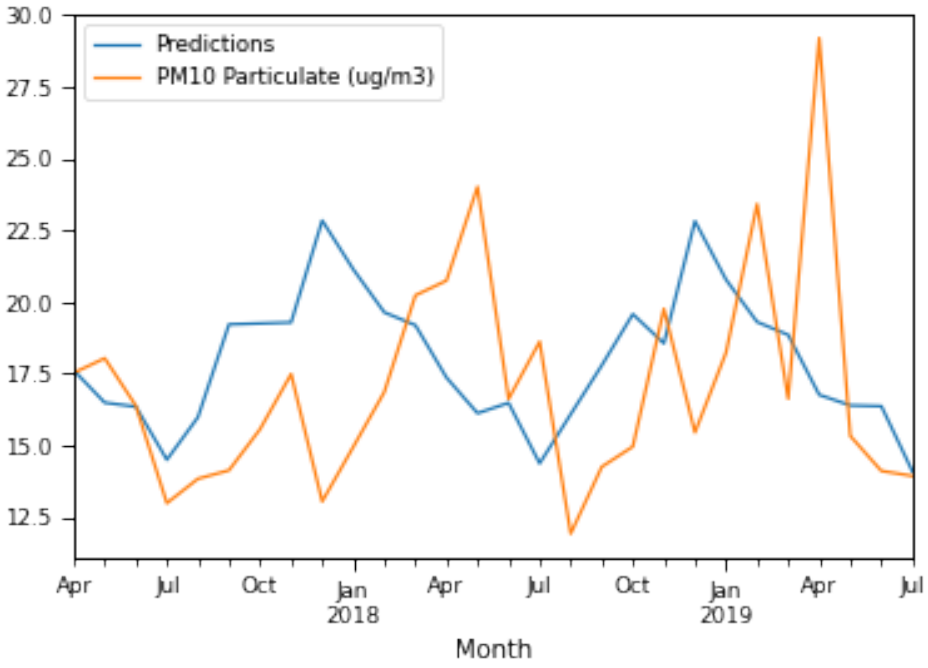

Figura 5.84: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas PM10 y la zona de Background. Los valores reales son en naranja y la predicción en azul.

- Partículas PM2.5: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,0)(2,1,0)[12].
  Si observamos la figura 5.85 vemos que el comportamiento de los valores reales difieren mucho de los que se han predicho.
  No se realiza una buena de predicción de los valores ni de la trazabilidad.
  El error que se obtiene es de 3,65.

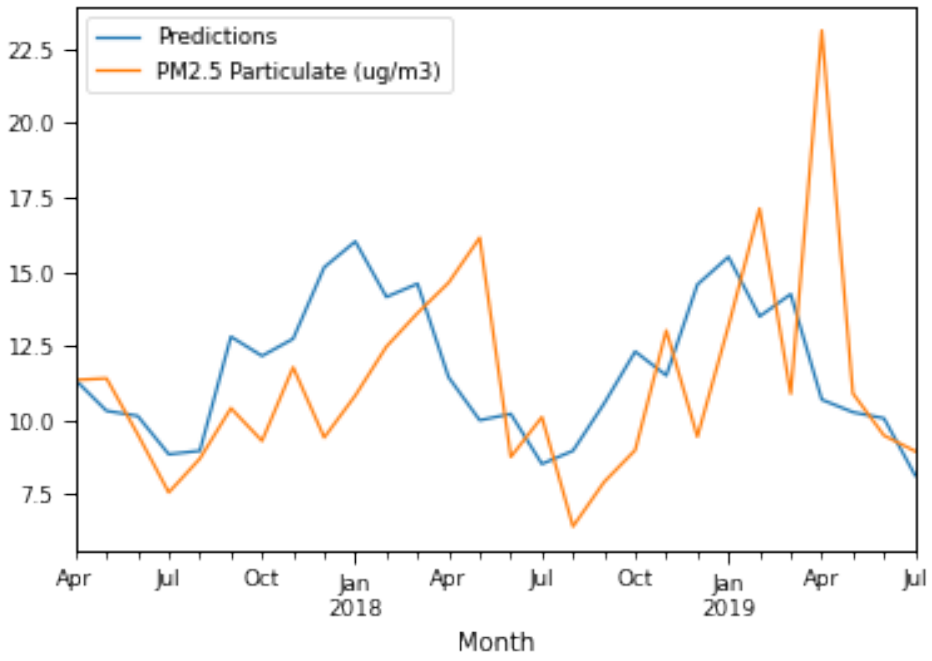

Figura 5.85: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas PM2.5 y la zona de Background. Los valores reales son en naranja y la predicción en azul.

- Dióxido de azufre: El modelo óptimo que hemos obtenido para esta partícula es ARIMA(1,0,0)(2,1,0)[12]. Si observamos la figura 5.86 vemos que los valores que predice el modelo tampoco se corresponden con los valores reales.El gráfico de predicción tiene la mayoría de sus valores por encima de los reales y su comportamiento es más lineal que el real. El error que se ha obtenido para esta gráfica es de 1,44.

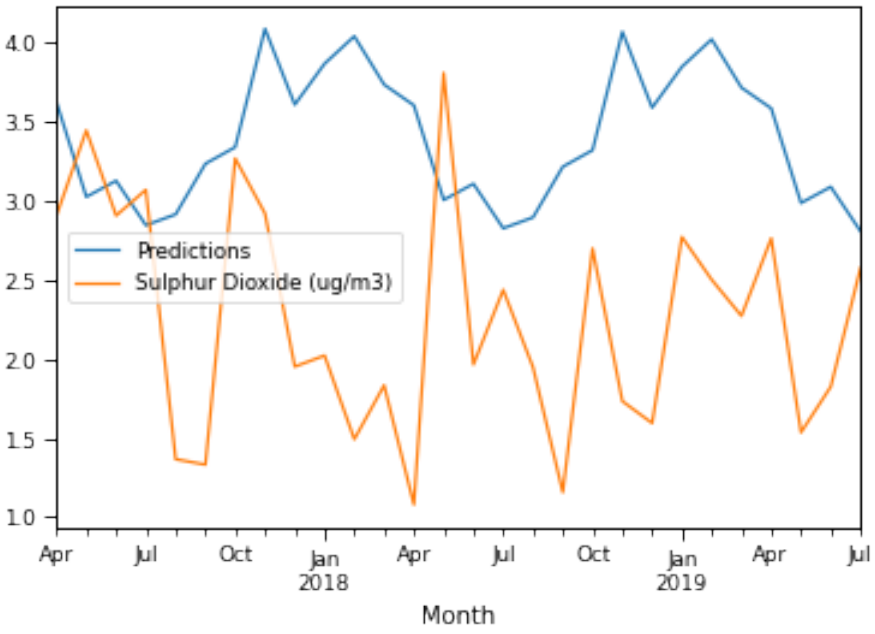

Figura 5.86: Predicción con el modelo de ARIMA y sus valores óptimos para las partículas de dióxido de azufre y la zona de Background. Los valores reales son en naranja y la predicción en azul.

	Zona Roadside	Zona Background
Óxido nítrico	17,66	21,43
Dióxido de nitrógeno	10,13	4,58
Óxidos de nitrógeno	47,07	13,97
Ozono	3,42	6,85
Partículas PM10	4,64	4,56
Partículas PM2.5	4,41	3,65
Dióxido de azufre	3,09	1,44

Tabla 5.17: Tabla con el error cuadrático medio generado por la téxnica de ARIMA para la zona de Roadside y Background.

5.4. LSTM

Finalmente, vamos a proceder a realizar la predicción con redes neuronales.

Vamos a proceder a hacer y explicar dos métodos distintos. El primero de ello será una red LSTM con regresión = 1.
El segundo serán una red LSTM utilizando el método de la ventana. Esto quiere decir que se usarán x movimientos atrás para poder realizar la predicción.

El siguiente código es común para las distintas predicciones que vamos a hacer por lo que vamos a explicarlo detalladamente antes de realizar las predicciones.

```
1  def create_dataset(dataset, look_back=1):
2    dataX, dataY = [], []
3    for i in range(len(dataset)-look_back-1):
4      a = dataset[i:(i+look_back), 0]
5      dataX.append(a)
6      dataY.append(dataset[i + look_back, 0])
7    return np.array(dataX), np.array(dataY)
8
9  # cargamos el conjunto de datos
10 dataframe = dataset_metric
11 dataset = dataframe.values
12 dataset = dataset.astype('float32')
13
14 # normalizamos el conjunto de datos
15 scaler = MinMaxScaler(feature_range=(0, 1))
16 dataset = scaler.fit_transform(dataset)
17
18 # dividimos entre entranmiento y test
19 train_size = int(len(dataset) * 0.8)
20 test_size = int(len(dataset) * 0.2)
21 train, test = dataset[0:train_size,:], dataset[train_size:len(dataset)
       ,:]
```

Lo primero que vemos en el código es la función `create_dataset` a la que le pasamos como parámetro el conjunto de datos y una variable denominada `look_back`.
Esta función se utiliza para crear un nuevo conjunto de datos. El conjunto de datos que le pasamos como parámetro es el que queremos transformar. En nuestro caso, `look_back` es el número de meses previos que usaremos como variable de entrada para la siguiente predicción. Por defecto le hemos puesto como valor 1.
Como salida creará un conjunto de datos donde `X` será la cantidad de particulas para cierto momento `t` e `Y` será la cantidad para el momento `t+1`.

Desde la línea 11 a la 13 cargamos los valores de la métrica y la conver-

timos a un tipo float 32.
Ya que las LSTMs son sensibles a la escala de los datos de entrada, en la línea 15 asignamos a la variable `scaler` la función `MinMaxScaler` para que ajuste los valores a una escala entre 0 y 1.
A continuación, en la línea 16 usamos la función `fit_transform` junto a `scaler` en nuestro conjunto de datos, ya que queremos usar los datos normalizados para entrenar nuestro modelo.

Una vez tenemos nuestro modelo normalizado, dividimos nuestro conjunto de datos en dos. El primero de ellos de entrenamiento con el 80 % de los datos y el segundo de ellos con el 20 % de los datos restantes.

LSTM para regresión

Expresamos el problema de predicción como un problema de regresión, es decir, si tenemos x cantidad de una partícula este mes, ¿cuál será la cantidad el próximo mes?

```
1  look_back = 1
2  trainX, trainY = create_dataset(train, look_back)
3  testX, testY = create_dataset(test, look_back)
4
5  trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
6  testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
7
8  #creamos la LSTM
9  model = Sequential()
10 model.add(LSTM(4, input_shape=(1, look_back)))
11 model.add(Dense(1))
12 model.compile(loss='mean_squared_error')
13 model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
14
15 # hacemos las predicciones
16 trainPredict = model.predict(trainX)
17 testPredict = model.predict(testX)
18
19 # invertimos las predicciones
20 trainPredict = scaler.inverse_transform(trainPredict)
21 trainY = scaler.inverse_transform([trainY])
22 testPredict = scaler.inverse_transform(testPredict)
23 testY = scaler.inverse_transform([testY])
24
25 # calculamos el error
26 trainScore = math.sqrt(mean_squared_error(trainY[0], trainPredict
       [:,0]))
27 print('Resultado del entrenamiento: %.2f RMSE' % (trainScore))
28 testScore = math.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
29 print('Resultado del test: %.2f RMSE' % (testScore))
30
31 trainPredictPlot = np.empty_like(dataset)
32 trainPredictPlot[:, :] = np.nan
33 trainPredictPlot[look_back:len(trainPredict)+look_back, :] =
       trainPredict
34
35 testPredictPlot = np.empty_like(dataset)
36 testPredictPlot[:, :] = np.nan
```

```
37 testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] =
       testPredict
38
39 plt.plot(scaler.inverse_transform(dataset))
40
41 plt.plot(trainPredictPlot,'r', linewidth = 2)
42 plt.plot(testPredictPlot,'m', linewidth = 2)
43 plt.legend( ('Datos', 'Prediccion datos entramiento', 'Prediccion
       sobre los datos de test'), loc = 'upper left')
44 plt.grid(True)
45 plt.title("Prediccion", fontsize = 15)
46 plt.xlabel("Meses", fontsize = 10)
47 plt.ylabel("Valor", fontsize = 10)
48 plt.show()
```

Ahora vamos a usar la función que hemos creado al principio con el valor de `look_back` igual a 1. En las líneas 2 y 3 creamos los conjuntos de datos `trainX` y `trainY` que contendrán los valores de entrenamiento para los momentos `t` y `t+1` respectivamente. De igual forma creamos los conjuntos de datos `testX` y `testY` que contendrán los valores de testeo para los momentos `t` y `t+1`.

Tenemos que recordar que las redes LSTM esperan tener como entrada unos datos que contienen las muestras, los pasos de tiempo y las características. Ya que nuestros datos no tienen esos pasos de tiempo, usamos en las líneas 5 y 6 la función `reshape` para poner con la estructura que queremos a los conjuntos `trainX` y `testX`

Una vez tenemos los datos con la estructura que deseamos, desde la línea 9 creamos la red neuronal. Esta red tiene una capa visible de 1 capa, una capa oculta con 4 neuronas y una capa de salida que realiza una predicción. La función de activación que se utiliza es sigmoidal, ya que esta es la que sua por defecto. La red está entrenada por 100 épocas y se usa un tamaño de lote igual a 1. En la línea 13 entrenamos el modelo con los datos de entrenamiento.

A continaución, realizamos las predicciones tanto del conjunto de datos de entrenamiento como el de testeo en las líneas 16 y 17.

Volvemos a transformar nuestro conjunto desde la línea 20 a la 23 para tenerlo como al principio. Para ello usamos la función `inverse_transform`. Desde la línea 25 a la a la 29 calculamos y mostramos el error cuadrático medio tanto de la parte de entrenamiento como de testeo.

Finalmente desde la línea 31 a la 44, mostramos los datos reales, los datos de predicción para los datos de entrenamiento y la predicción de los datos sobre la parte de testeo.

Una vez explicado el procedimiento, vamos a ver que resultados hemos obtenido con nuestros datos.

Empezamos viendo las zona de Roadside para cada una de las partículas, podemos ver sus errores en la tabla 5.18.

- Óxido nítrico: Podemos ver en la figura 5.87 que la trazabilidad es casi idéntica, salvo que algunos de los valores reales son un poco menores que los predichos.
  Sin embargo, hay que indicar lo bien que se ajusta la predicción al cambio de tendencia repentino.
  El error de predicción para los datos de entrenamiento es de 17,79 y el error para los datos de prueba es de 20,20.

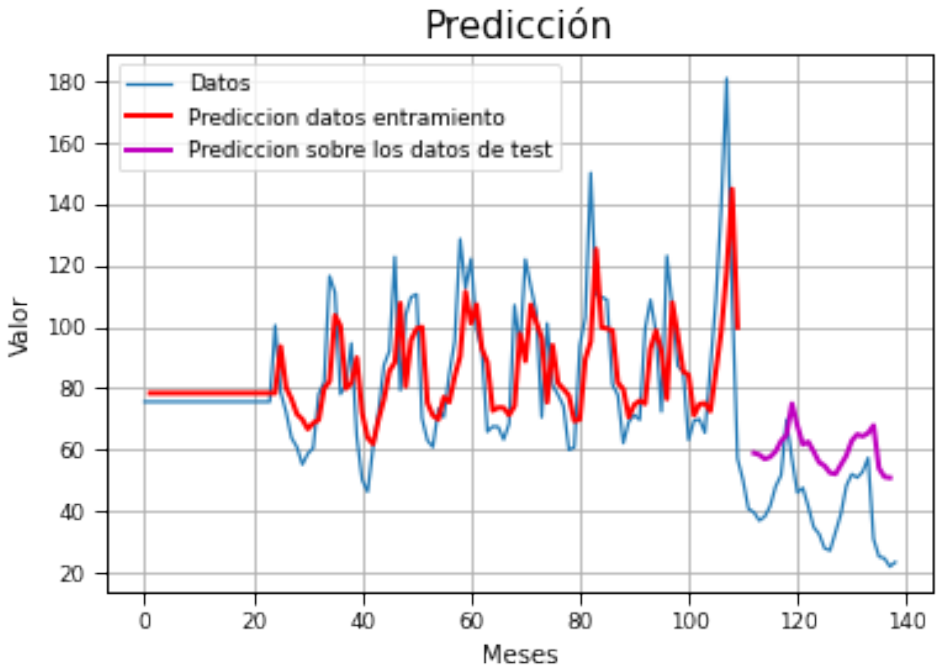

Figura 5.87: Predicción con LSTM y regresión 1 para las partículas de óxido nítrico en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de nitrógeno: Podemos ver en la figura 5.88 que la predicción no es tan buena como podríamos esperar, la trazabilidad en los datos de entrenamiento es bastante parecida, aunque no consigue acercarse a los extremos de los valores reales.
  Si nos centramos en los datos de testeo, vemos que la mayoría de los valores se encuentran bastante por encima de los reales, aunque su movimiento sea muy parecido. El error de predicción para los datos de entrenamiento es de 6,55 y el error para los datos de prueba es de 7,22.

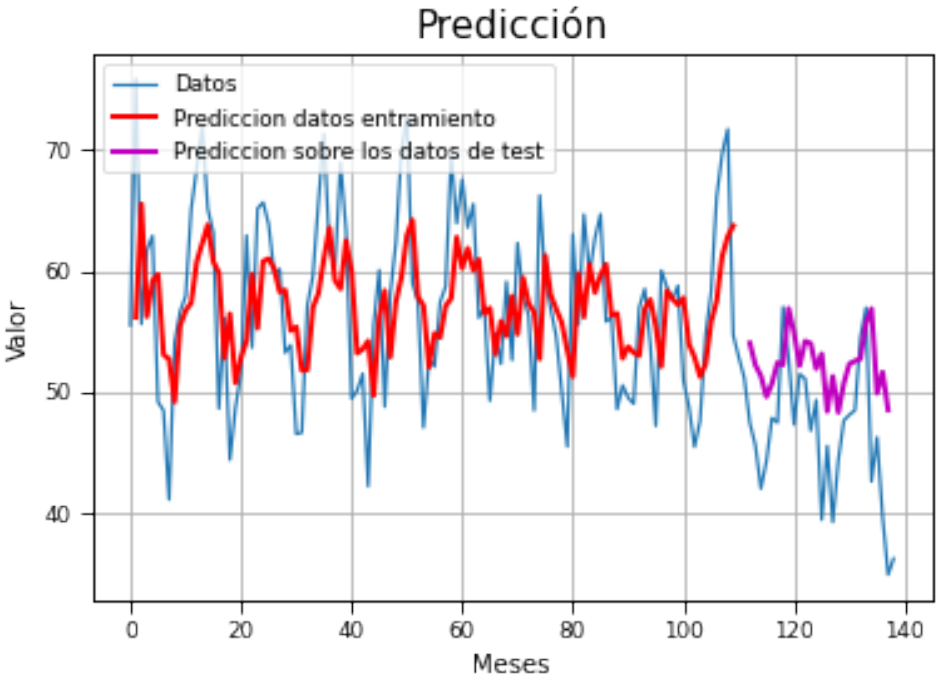

Figura 5.88: Predicción con LSTM y regresión 1 para las partículas de dióxido de nitrógeno en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Óxidos de nitrógeno: Podemos ver en la figura 5.89 que la predicción de los datos tanto los de entrenamiento como lo de testeo son bastante buenos.
  Podemos ver como la trazabilidad entre ambas es muy similar y se acerca mucho a los extremenos relativos y absolutos de la gráfica real. El error de predicción para los datos de entrenamiento es de 22,07 y el error para los datos de prueba es de 22,07.

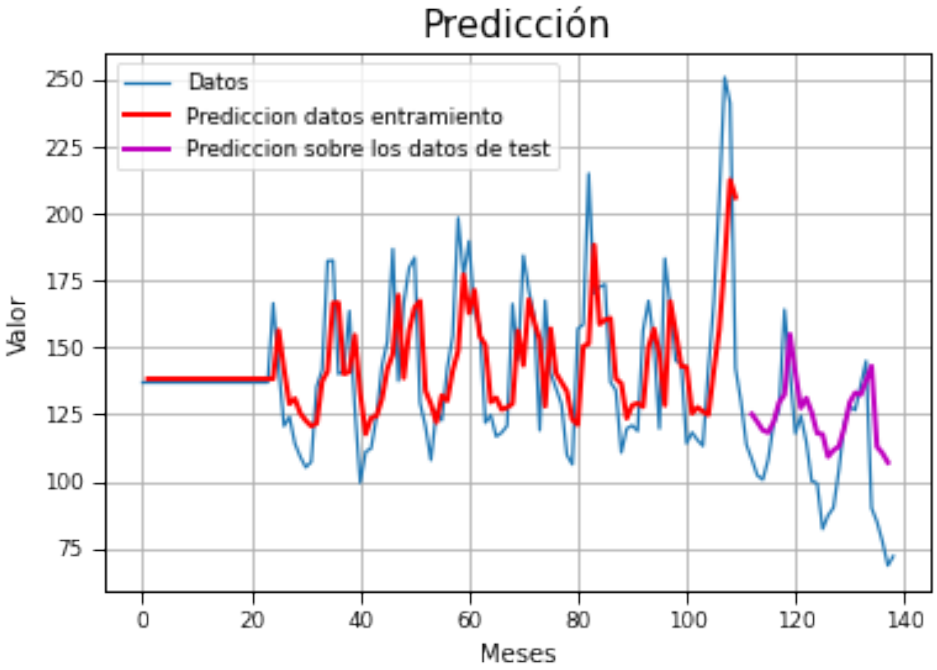

Figura 5.89: Predicción con LSTM y regresión 1 para las partículas de óxidos de nitrógeno en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Ozono: Podemos ver en la figura 5.90 que la predicción obtenida para el ozono es bastante buena.
  El comportamiento de la predicción es prácticamente idéntico al real, aunque este no consiga llegar a los valores máximos y mínimos que alcanzan los valores reales.
  El error de predicción para los datos de entrenamiento es de 6,6 y el error para los datos de prueba es de 5,1.

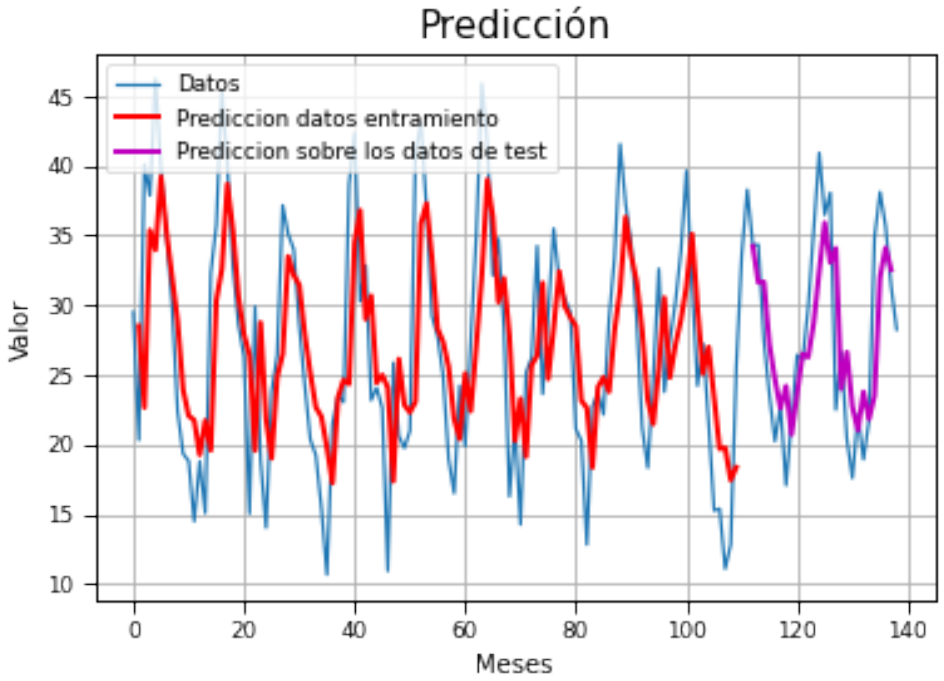

Figura 5.90: Predicción con LSTM y regresión 1 para las partículas de ozono en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM10: Podemos ver en la figura 5.91 que la predicción obtenida para este tipo de partículas no es buena.
  Los valores predichos se encuentra un poco por debajo del centro de la gráfica y no se acercan a los extremos de los valores reales. El error de predicción para los datos de entrenamiento es de 4,8 y el error para los datos de prueba es de 4,87.

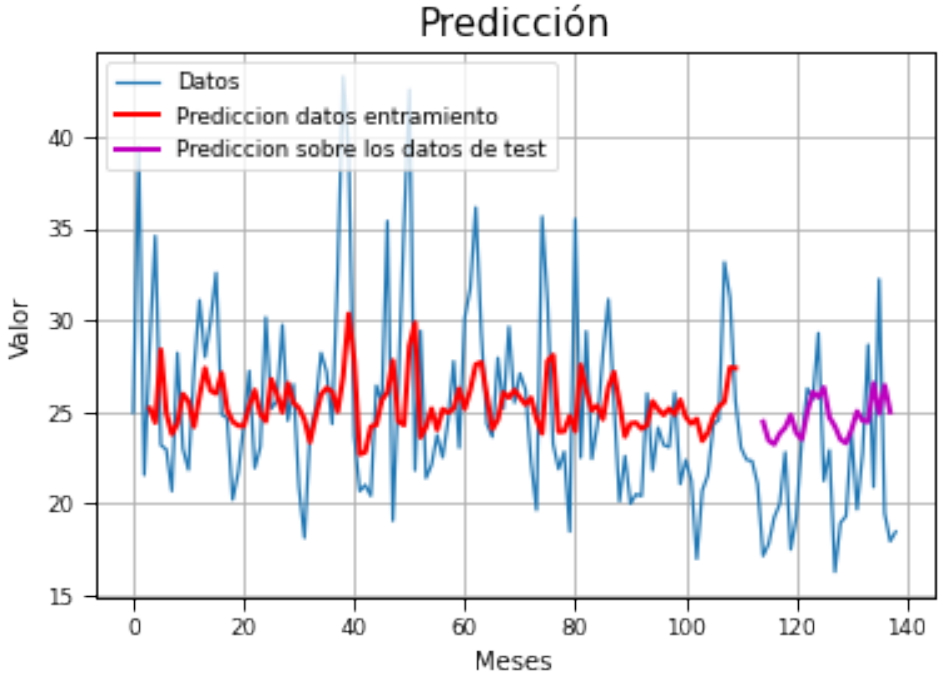

Figura 5.91: Predicción con LSTM y regresión 1 para las partículas PM10 en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM2.5: Podemos ver en la figura 5.92 que la predicción obtenida no se asemeja a los valores reales que toma la partícula.
  Los valores predichos se encuentra por el centro de la gráfica y no se acercan a los extremos de los valores reales. El error de predicción para los datos de entrenamiento es de 4,78 y el error para los datos de prueba es de 4,44.

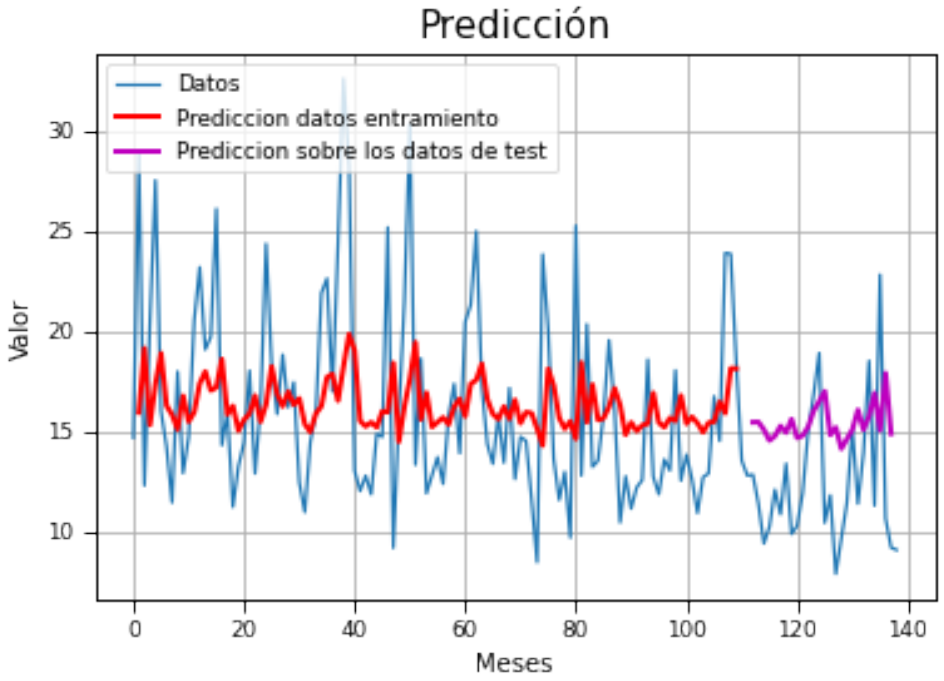

Figura 5.92: Predicción con LSTM y regresión 1 para las partículas PM2.5 en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de azufre: Podemos ver en la figura 5.93 que la predicción obtenida intenta ajustarse a la realidad y aunque no sea demasiado buena, se ajusta a las subidas y bajadas.
  Los valores predichos se encuentra por el centro de la gráfica y difiere en algunas partes bastante de la realidad. El error de predicción para los datos de entrenamiento es de 1,2 y el error para los datos de prueba es de 2,71.

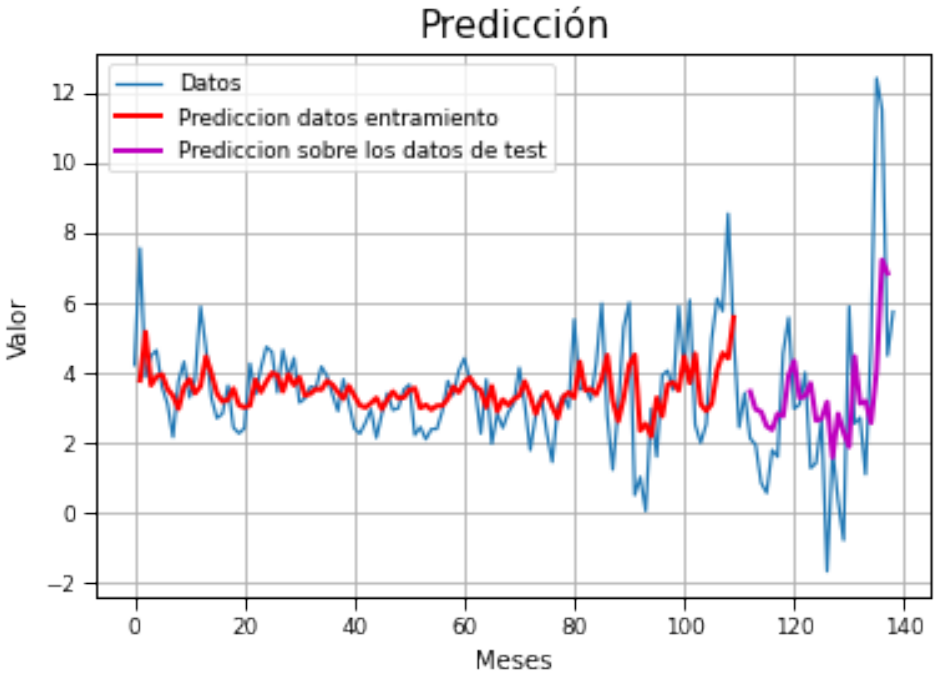

Figura 5.93: Predicción con LSTM y regresión 1 para las partículas de dióxido de azufre en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

A continuación, vamos a ver como se comportan las partículas con este método en la zona de Background. Podemos ver la tabla con los errores cuadráticos medios en la tabla 5.19

- Óxido nítrico: Podemos ver en la figura 5.94 que la trazabilidad es muy similar, aunque para los datos de entrenamiento no consiga llegar a los máximos y mínimos que se observan.
  Podemos ver que para los datos de testeo, ha conseguido una traza similar y se ha acercado bastante a los valores máximos pero se encuentra bastante alejado de los mínimos
  El error de predicción para los datos de entrenamiento es de 10,72 y el error para los datos de prueba es de 7,78.

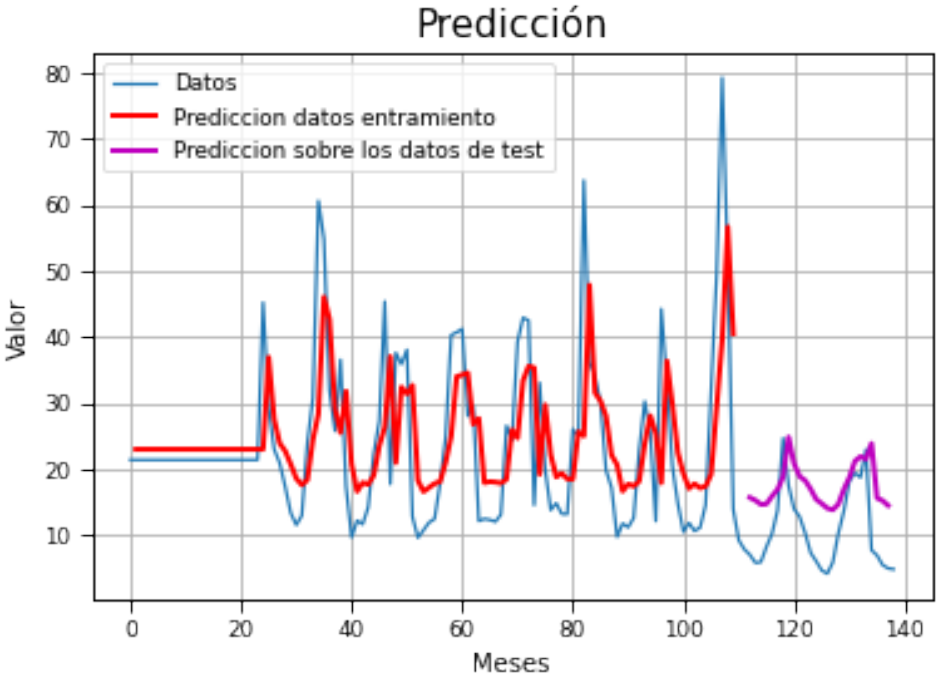

Figura 5.94: Predicción con LSTM y regresión 1 para las partículas de ozono en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de nitrógeno: Podemos ver en la figura 5.95 que la predicción es bastante similar a la realidad, la trazabilidad en los datos de entrenamiento es bastante parecida, aunque no consigue acercarse a los extremos de los valores reales.
  Si nos centramos en los datos de testeo, vemos que la trazabilidad sigue siendo bastante buena y aunque se encuentre un poco alejado de los mínimos, se acerca con bastante precisión a los máximos. El error de predicción para los datos de entrenamiento es de 6,49 y el error para los datos de prueba es de 5,32.

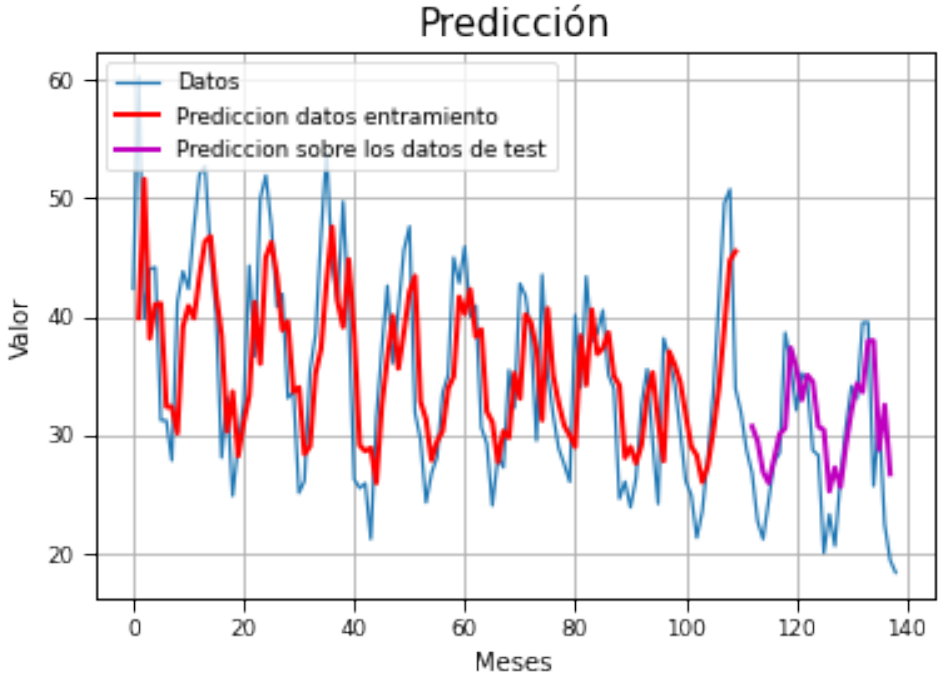

Figura 5.95: Predicción con LSTM y regresión 1 para las partículas de dióxido de nitrógeno en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Óxidos de nitrógeno: Podemos ver en la figura 5.96 que la predicción de los datos tanto los de entrenamiento como lo de testeo son bastante buenos.
  Podemos ver como la trazabilidad entre ambas es muy similar aunque no se acerca mucho a los extremenos relativos y absolutos de la gráfica real.
  El error de predicción para los datos de entrenamiento es de 16,11 y el error para los datos de prueba es de 12,44.

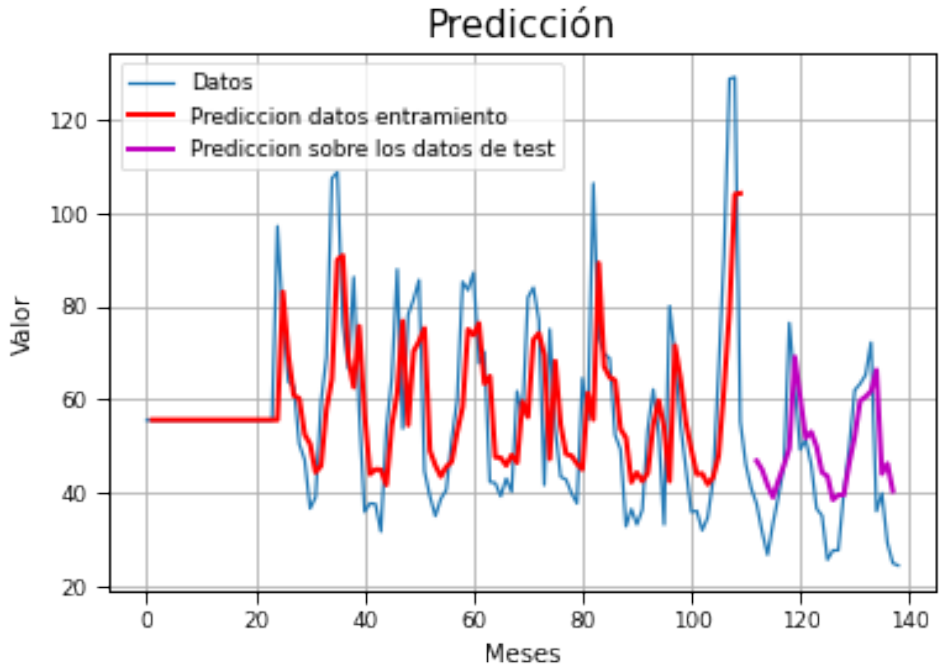

Figura 5.96: Predicción con LSTM y regresión 1 para las partículas de óxidos de nitrógeno en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Ozono: Podemos ver en la figura 5.97 que la predicción obtenida para el ozono es bastante buena.
  El comportamiento de la predicción es prácticamente idéntico al real, aunque este no consiga llegar a los valores máximos y mínimos que alcanzan los valores reales. Se comporta de una forma muy similar con los datos de entrenmiento y los de testeo.
  El error de predicción para los datos de entrenamiento es de 8,36 y el error para los datos de prueba es de 7,99.

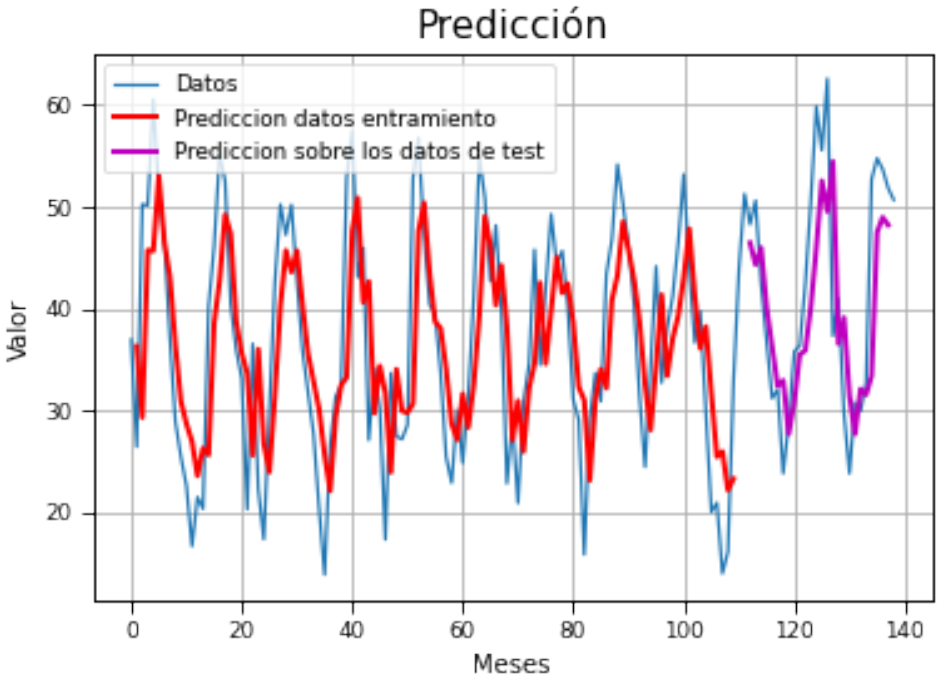

Figura 5.97: Predicción con LSTM y regresión 1 para las partículas de ozono en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM10: Podemos ver en la figura 5.98 que la predicción obtenida para este tipo de partículas no es buena.
  Los valores predichos se encuentran en torno al valor de 20 y no se acercan a los extremos de los valores reales.
  Mientras que la gráfica real tiene mucha diferencia entre valores máximos y mínimos, los de la predicción tiene una diferencia mmuy pequeña.
  El error de predicción para los datos de entrenamiento es de 4,65 y el error para los datos de prueba es de 4,35.

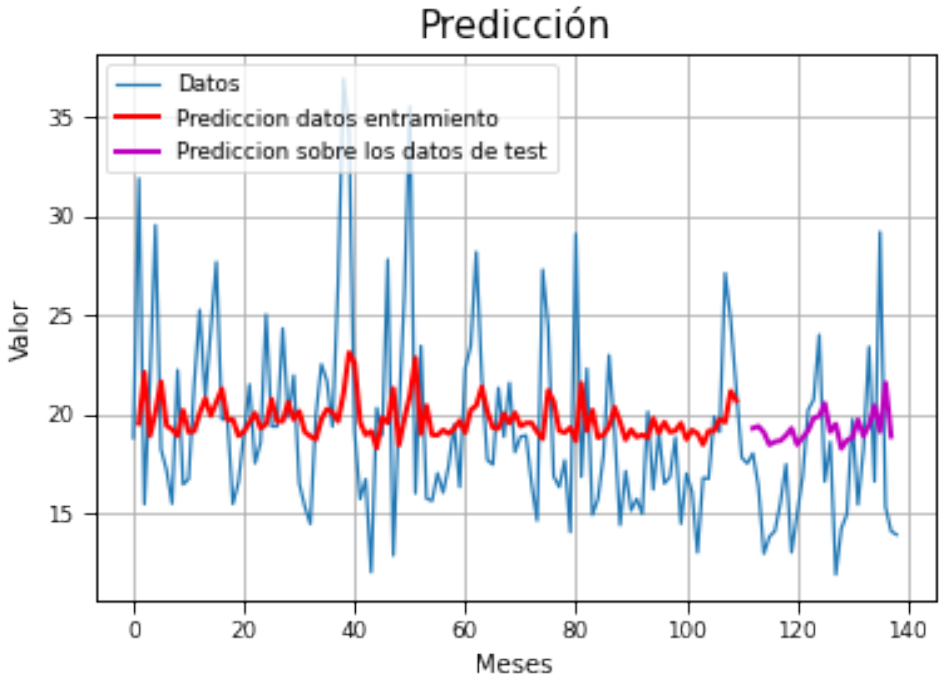

Figura 5.98: Predicción con LSTM y regresión 1 para las partículas PM10 en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM2.5: Podemos ver en la figura 5.99 que la predicción obtenida se asemeja en la trazabilidad a la de los datos reales, sin embargo, no consigue alcanzar los extremos de esta.
  Los valores predichos se encuentra por el centro de la gráfica real. El error de predicción para los datos de entrenamiento es de 4,23 y el error para los datos de prueba es de 3,78.

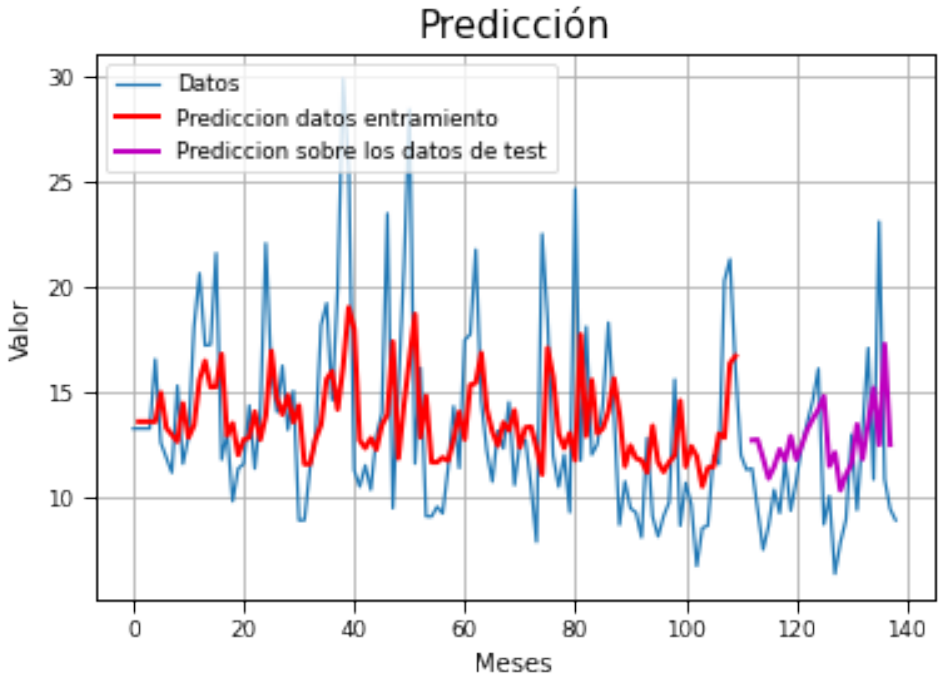

Figura 5.99: Predicción con LSTM y regresión 1 para las partículas PM2.5 en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de azufre: Podemos ver en la figura 5.100 que la predicción obtenida intenta ajustarse a la realidad, pero no es demasiado buena. Los valores predichos no tienen tanta diferencia en sus máximos y mínimos, tal y como pasa en la realida. Aún así, consigue un error cuadrático bastante pequeño.
  Los valores para la parte de testeo son superiores en casi todo su conjunto que los reales. El error de predicción para los datos de entrenamiento es de 0,85 y el error para los datos de prueba es de 1,14.

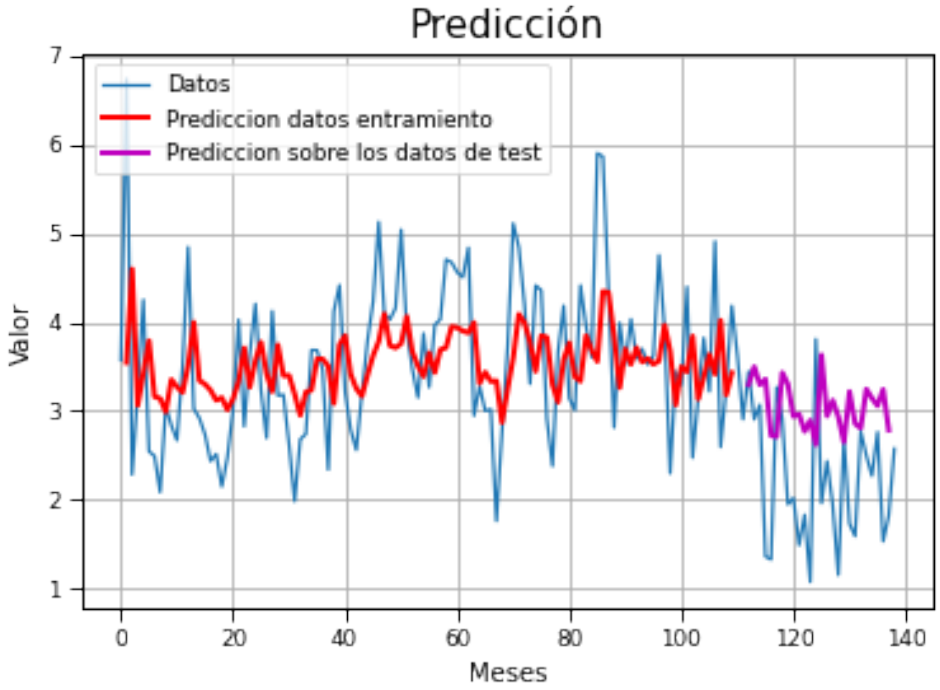

Figura 5.100: Predicción con LSTM y regresión 1 para las partículas de dióxido de azufre en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

LSTM para Regresión Usando el Método de la Ventana

En este caso, en lugar de usar solo un mes para realizar la predicción, podemos usar x meses para hacer la predicción para el siguiente mes. Este método se denomina Ventana y el tamaño es el valor que decidamos para el número de pasos atrás.

```
look_back = 3
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)

trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))

# creamos la LSTM
model = Sequential()
model.add(LSTM(4, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error')
model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)

# hacemos las predicciones
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)

# invertimos las predicciones
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])

# calculamos el error
trainScore = math.sqrt(mean_squared_error(trainY[0], trainPredict
    [:,0]))
print('Resultado del entrenamiento: %.2f RMSE' % (trainScore))
testScore = math.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
print('Resultado del test: %.2f RMSE' % (testScore))

trainPredictPlot = np.empty_like(dataset)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(trainPredict)+look_back, :] =
    trainPredict

testPredictPlot = np.empty_like(dataset)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] =
     testPredict

plt.plot(scaler.inverse_transform(dataset))

plt.plot(trainPredictPlot,'r', linewidth = 2)
plt.plot(testPredictPlot,'m', linewidth = 2)
plt.legend( ('Datos', 'Prediccion datos entramiento', 'Prediccion
    sobre los datos de test'), loc = 'upper left')
plt.grid(True)
plt.title("Prediccion", fontsize = 15)
plt.xlabel("Meses", fontsize = 10)
plt.ylabel("Valor", fontsize = 10)
plt.show()
```

Como podemos observar el código es exactamente igual que el anterior pero cambiando el valor del parámetro `look_back`. En nuestro caso hemos elegido el valor 3.

Una vez vistas las diferencias entre ambos códigos, vamos a ver que resultados hemos obtenido para nuestros datos.

Empezamos viendo las zona de Roadside para cada una de las partículas, podemos ver sus errores en la tabla 5.18.

- Óxido nítrico: Podemos ver en la figura 5.101 que la trazabilidad de la gráfica es muy pareida, salvo que algunos de los valores reales, sin embargo los valores de test predichos son un poco mayores que los reales.
  Se ajusta bastante bien al movimiento que realiza.
  El error de predicción para los datos de entrenamiento es de 17 y el error para los datos de prueba es de 20,68.

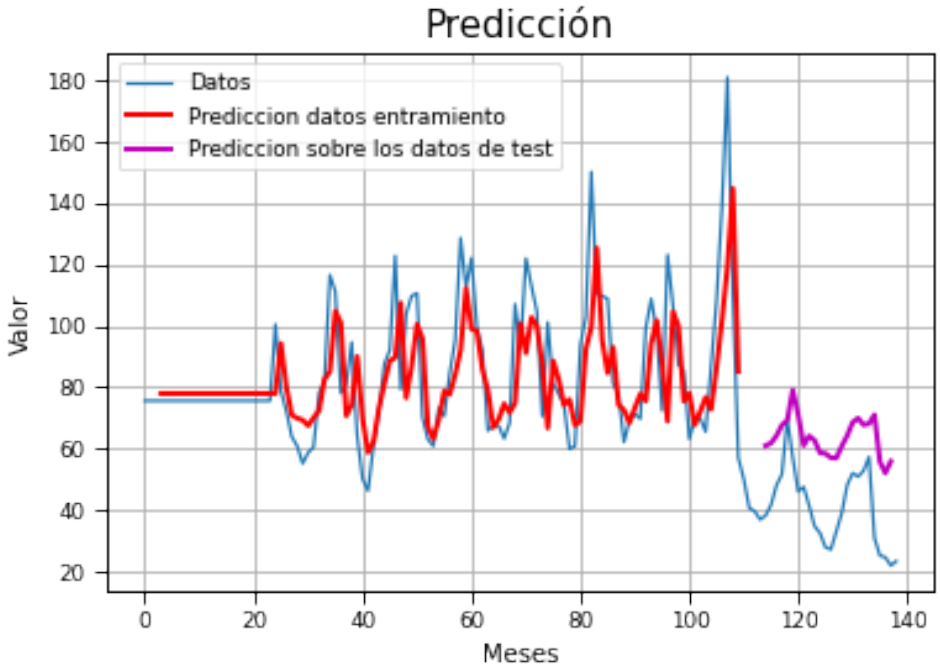

Figura 5.101: Predicción con LSTM usando el método de la ventana para las partículas de óxido nítrico en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de nitrógeno: Podemos ver en la figura 5.102 que la predicción no es demasiado buena, ya que no es capaz de obtener los valores máximos y mínimos a los que llegan los valores reales.
  Si nos centramos en los datos de testeo, vemos que la mayoría de los valores se encuentran bastante por encima de los reales, aunque su movimiento sea bastante parecido. El error de predicción para los datos de entrenamiento es de 5,95 y el error para los datos de prueba es de 8,08.

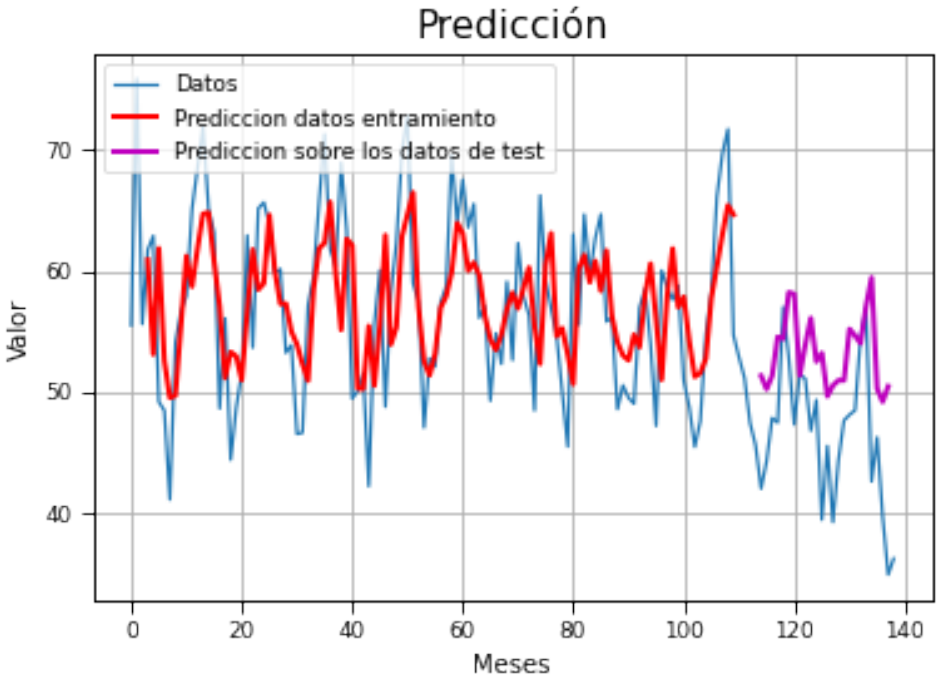

Figura 5.102: Predicción con LSTM usando el método de la ventana para las partículas de dióxido de nitrógeno en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Óxidos de nitrógeno: Podemos ver en la figura 5.103 que la predicción de los valores concuerda bastante con la realidad.
  El comportamiento de ambas gráficas es muy parecido e includo hay momento en los que se consiguen sobreponer.
  El error de predicción para los datos de entrenamiento es de 21,07 y el error para los datos de prueba es de 23,78.

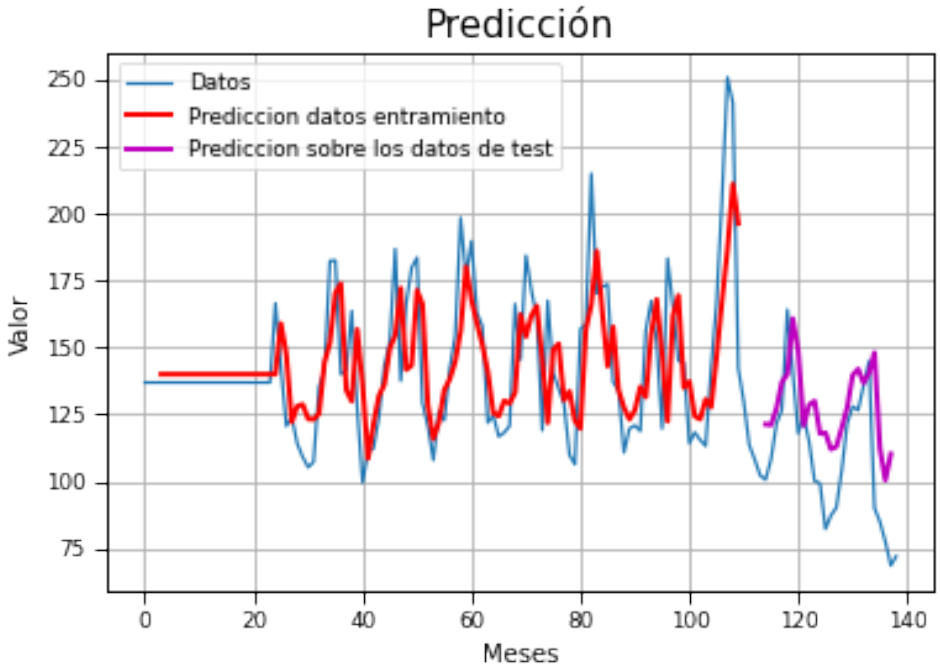

Figura 5.103: Predicción con LSTM usando el método de la ventana para las partículas de óxidos de nitrógeno en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Ozono: Podemos ver en la figura 5.104 que la predicción obtenida para el ozono es bastante buena.
  La trazabilidad de la serie y de la predicción es bastante similar y aunque los valores mínimos no los recoge demasiado bien, se acerca más a los valores máximos.
  El error de predicción para los datos de entrenamiento es de 6,08 y el error para los datos de prueba es de 4,67.

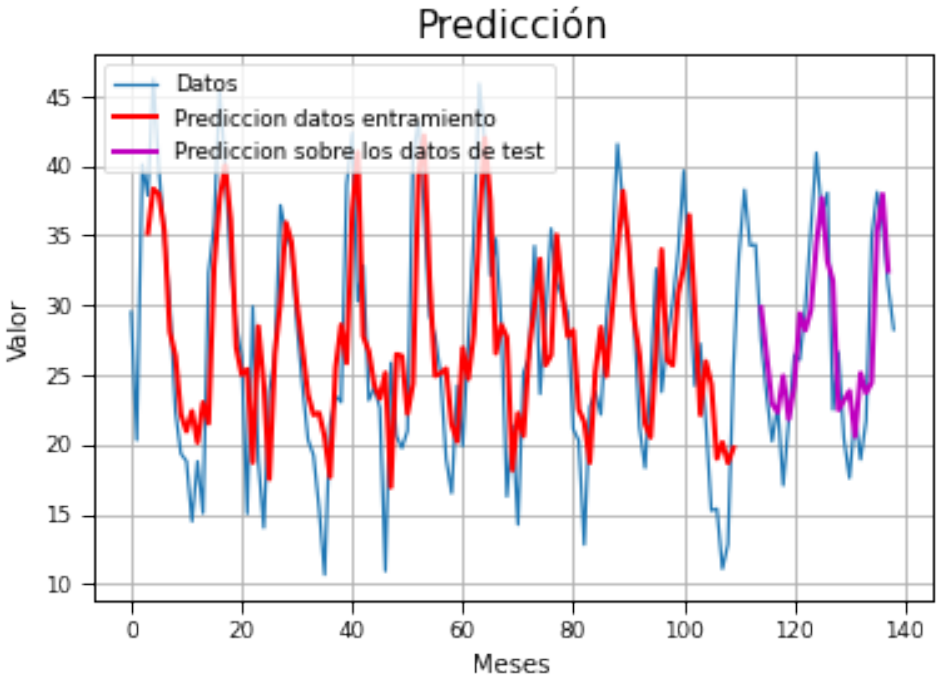

Figura 5.104: Predicción con LSTM usando el método de la ventana para las partículas de ozono en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM10: Podemos ver en la figura 5.105 que la predicción obtenida para este tipo de partículas se asemeja a los reales, por lo que no es una buena predicción.
  Los valores predichos se encuentra un poco por debajo del centro de la gráfica y no se acercan a los extremos de los valores reales ni tiene un comportamiento a lo largo del tiempo similar. El error de predicción para los datos de entrenamiento es de 4,85 y el error para los datos de prueba es de 4,81.

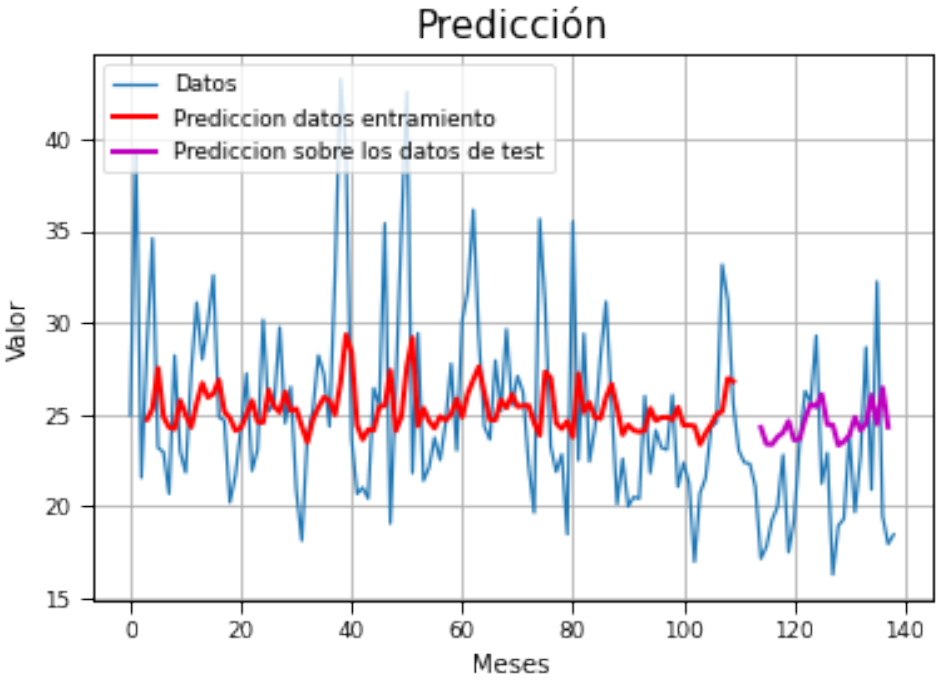

Figura 5.105: Predicción con LSTM usando el método de la ventana para las partículas PM10 en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM2.5: Podemos ver en la figura 5.106 que la predicción obtenida no es buena, ya que no se parece a la real.
  Los valores predichos se encuentra por el centro de la gráfica y no se acercan a los extremos de los valores reales. El error de predicción para los datos de entrenamiento es de 4,58 y el error para los datos de prueba es de 4,46.

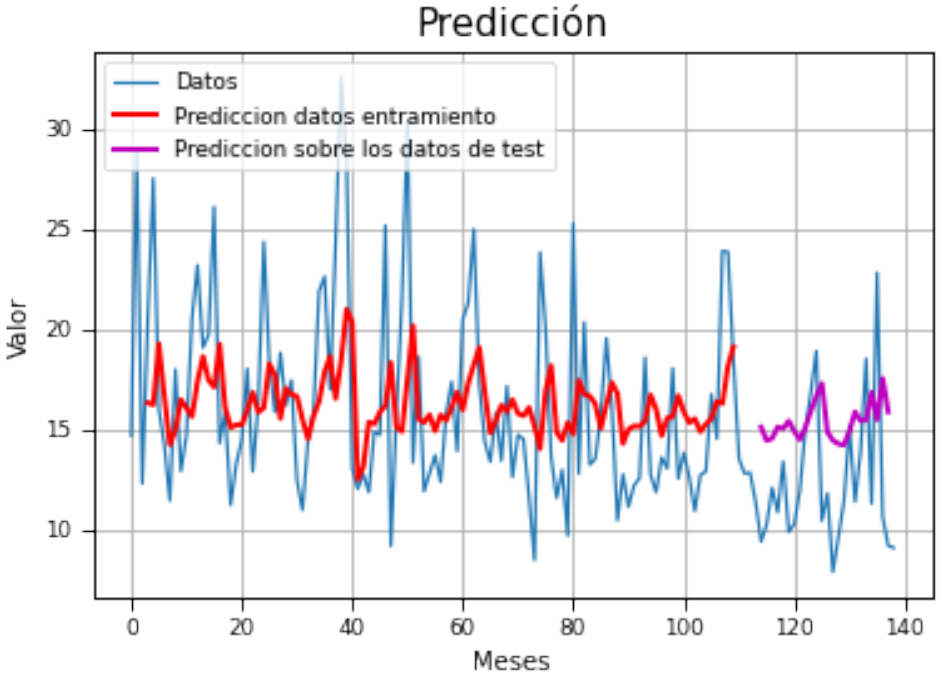

Figura 5.106: Predicción con LSTM usando el método de la ventana para las partículas PM2.5 en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de azufre: Podemos ver en la figura 5.107 que la predicción obtenida se asemeja un poco en la trazabilidad de los valores reales. Los valores predichos se encuentra por el centro de la gráfica y difiere en algunas partes bastante con la realidad, dependiendo de si existen máximos o mínimos muy marcados. El error de predicción para los datos de entrenamiento es de 1,14 y el error para los datos de prueba es de 2,81.

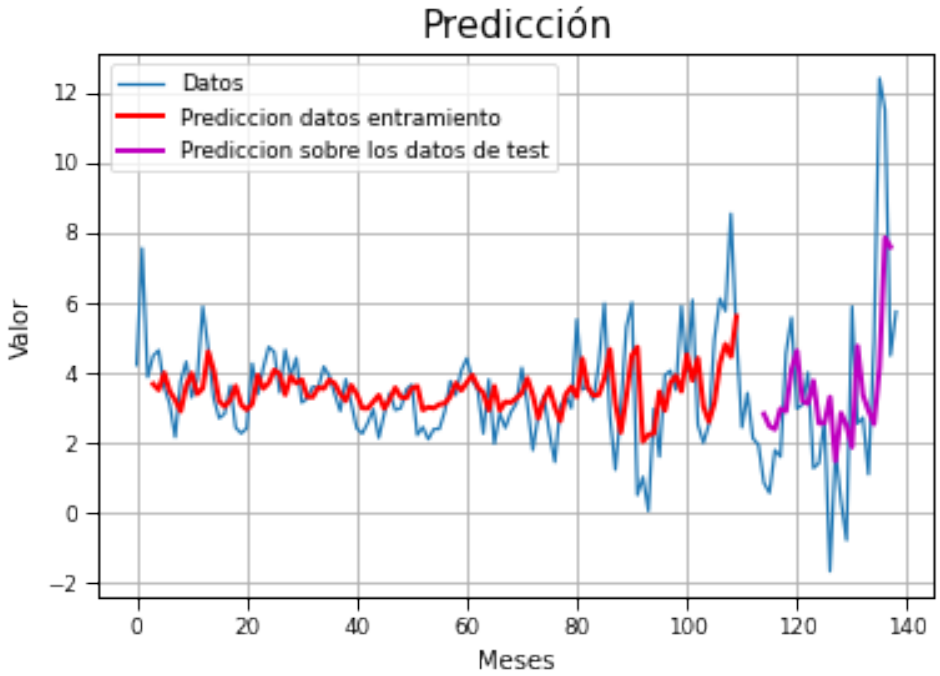

Figura 5.107: Predicción con LSTM usando el método de la ventana para las partículas de dióxido de azufre en la zona de Roadside. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

A continuación, vamos a ver como se comportan las partículas con este método en la zona de Background. Podemos ver la tabla con los errores cuadráticos medios en la tabla 5.19

- Óxido nítrico: Podemos ver en la figura 5.108 que la trazabilidad es muy similar, aunque para los datos de entrenamiento no consiga llegar a los máximos y mínimos que se observan.
  Podemos ver que para los datos de testeo, ha conseguido una traza similar y se ha acercado bastante a los valores máximos pero se encuentra bastante alejado de los mínimos
  El error de predicción para los datos de entrenamiento es de 10,72 y el error para los datos de prueba es de 7,78.

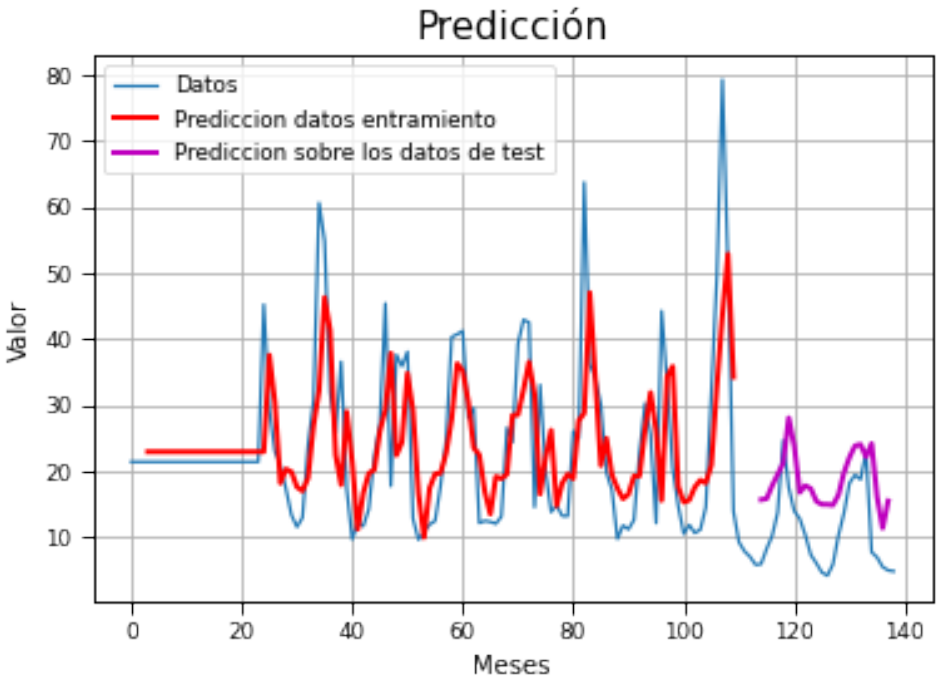

Figura 5.108: Predicción con LSTM usando el método de la ventana para las partículas de óxido nítrico en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de nitrógeno: Podemos ver en la figura 5.109 que la predicción es bastante buena. La trazabilidad en los datos de entrenamiento es bastante parecida y consigue acercarse bastante, e incluso en algunos casos muy cerca, a los extremos de los valores reales.
  Si nos centramos en los datos de testeo, vemos que la trazabilidad sigue siendo bastante buena y aunque se encuentre un poco alejado de los mínimos, se acerca con bastante precisión a los máximos. El error de predicción para los datos de entrenamiento es de 5,96 y el error para los datos de prueba es de 5,18.

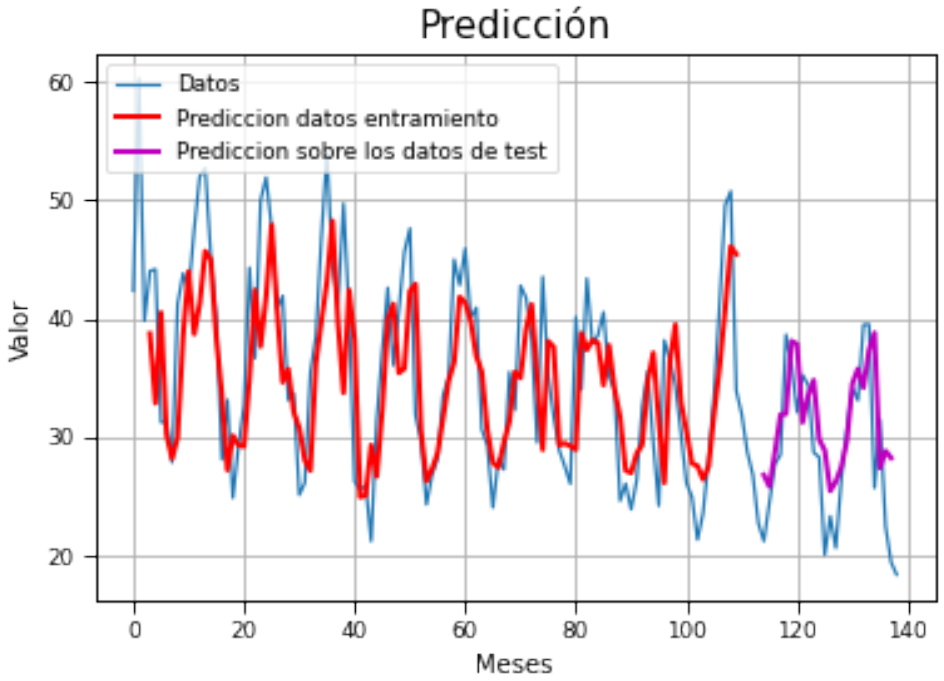

Figura 5.109: Predicción con LSTM usando el método de la ventana para las partículas de dióxido de nitrógeno en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Óxidos de nitrógeno: Podemos ver en la figura 5.110 que la predicción de los datos tanto los de entrenamiento como lo de testeo son bastante buenos.
  Podemos ver como la trazabilidad entre ambas es muy similar, aunque en la parte de entrenamiento consigue acercarse más a los máximos y mínimos, en la parte de testeo solo consigue predecir los máximos.
  El error de predicción para los datos de entrenamiento es de 15,36 y el error para los datos de prueba es de 13,29.

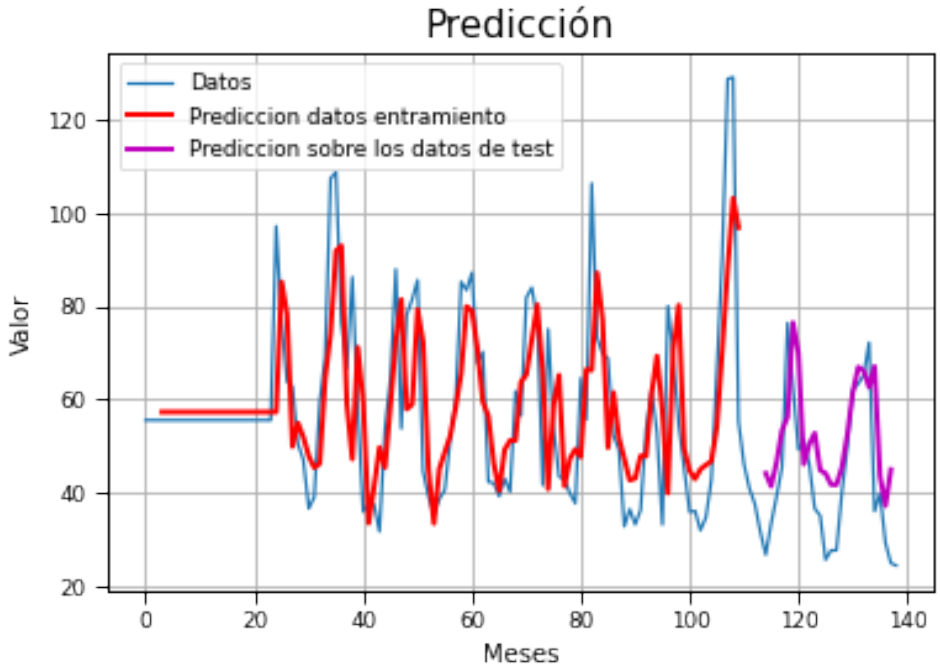

Figura 5.110: Predicción con LSTM usando el método de la ventana para las partículas de óxidos de nitrógeno en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Ozono: Podemos ver en la figura 5.111 que la predicción obtenida para el ozono en general es bastante buena.
  El comportamiento de la predicción es parecido al real, aunque este no consiga llegar a la mayoría de los valores máximos y mínimos que alcanzan los valores reales, se acerca bastante, incluso en alguna ocasiones toma valores muy cercanos. Para la parte de testeo consigue acercarse más a los mínimos que a los máximos.
  El error de predicción para los datos de entrenamiento es de 7,69 y el error para los datos de prueba es de 7,63.

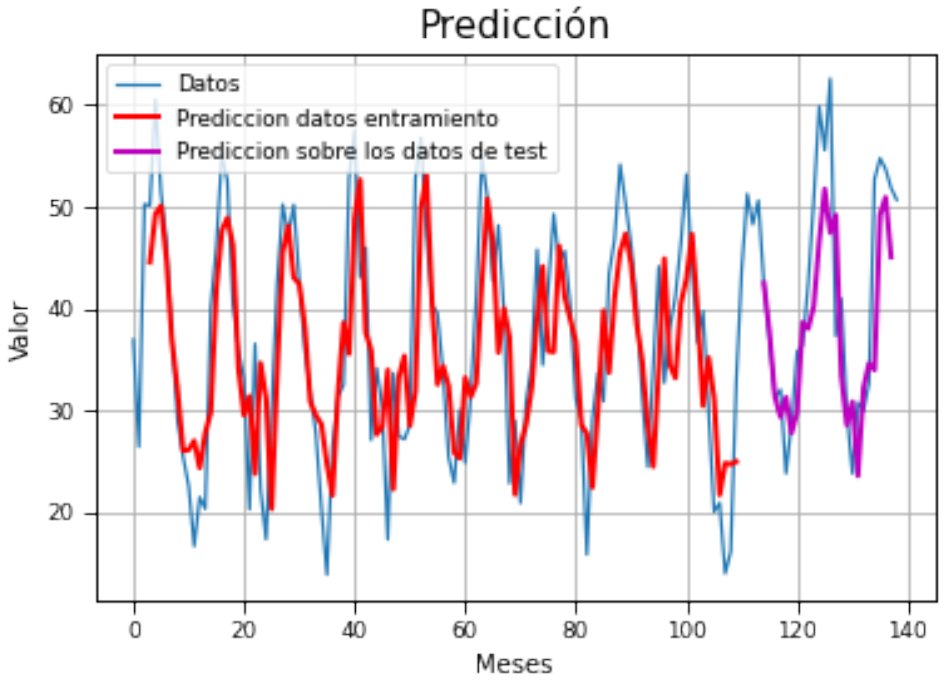

Figura 5.111: Predicción con LSTM usando el método de la ventana para las partículas de ozono en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM10: Podemos ver en la figura 5.112 que la predicción obtenida para este tipo de partículas no es acertada.
  Los valores predichos se encuentran en torno al valor de 20 y no se acercan a los extremos de los valores reales.
  Mientras que la gráfica real tiene mucha diferencia entre valores máximos y mínimos, los de la predicción tiene una diferencia mmuy pequeña.
  El error de predicción para los datos de entrenamiento es de 4,45 y el error para los datos de prueba es de 4,4.

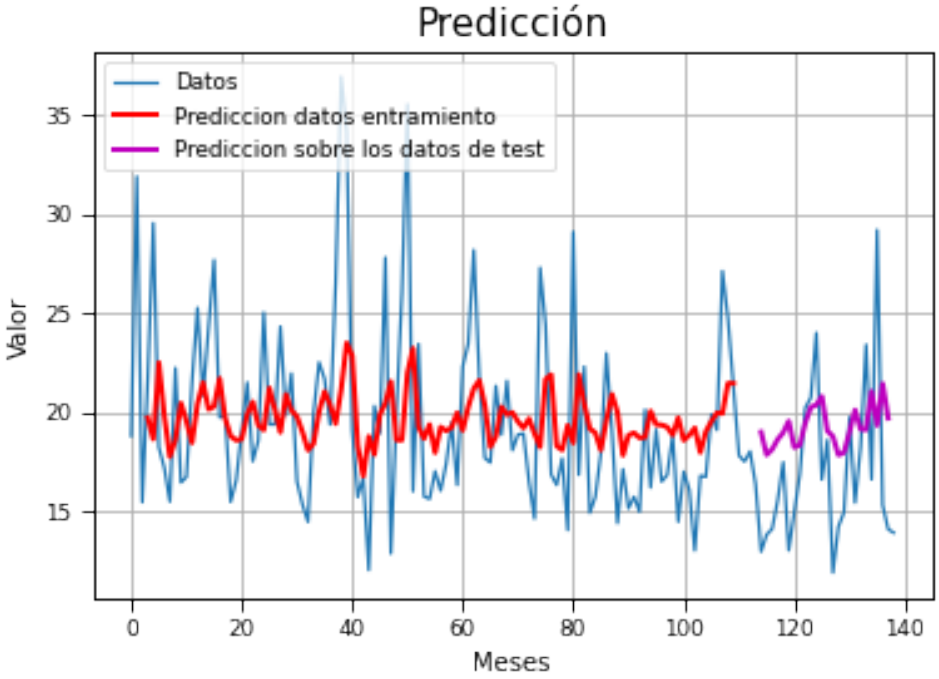

Figura 5.112: Predicción con LSTM usando el método de la ventana para las partículas PM10 en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Partículas PM2.5: Podemos ver en la figura 5.113 que la predicción obtenida se asemeja en la trazabilidad a la de los datos reales, sin embargo, no consigue alcanzar los extremos de esta, aunque si sabe que existen.
  El error de predicción para los datos de entrenamiento es de 4,26 y el error para los datos de prueba es de 3,66.

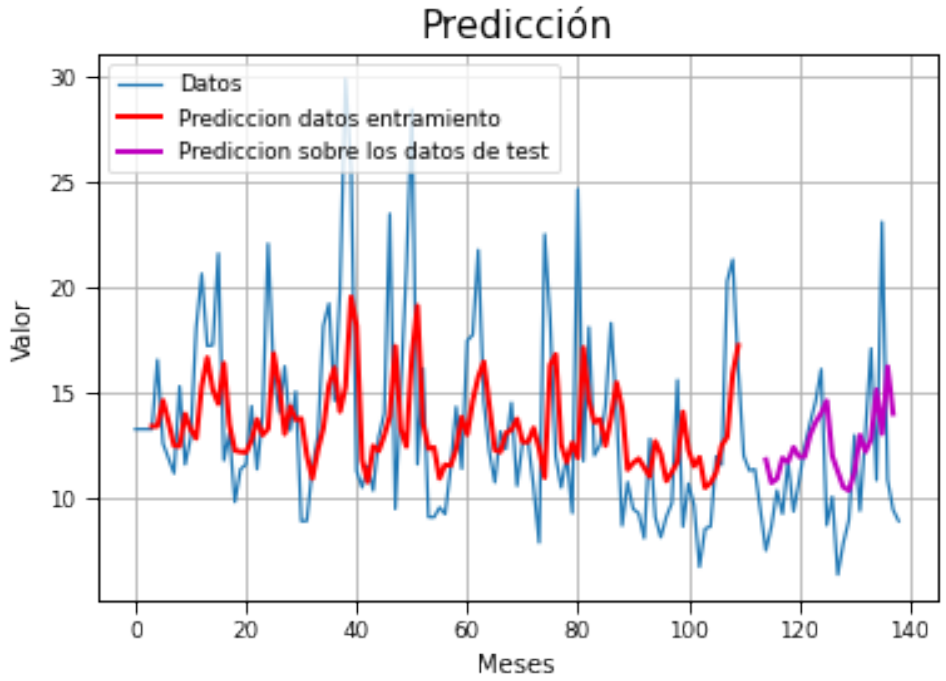

Figura 5.113: Predicción con LSTM usando el método de la ventana para las partículas PM2.5 en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

- Dióxido de azufre: Podemos ver en la figura 5.114 que la predicción obtenida intenta ajustarse a la realidad y tiene una trazabilidad muy parecida.
  Los valores predichos no tienen tanta diferencia en sus máximos y mínimos, tal y como pasa en la realidad. Aún así, consigue un error cuadrático bastante pequeño.
  Los valores para la parte de testeo son superiores en casi todo su conjunto que los reales. El error de predicción para los datos de entrenamiento es de 0,76 y el error para los datos de prueba es de 1,08.

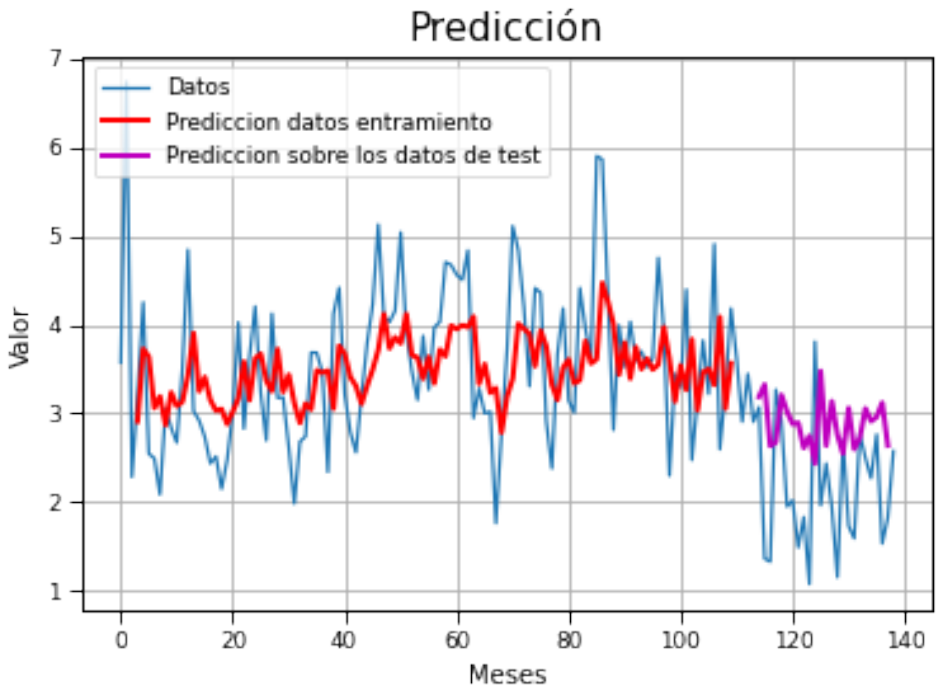

Figura 5.114: Predicción con LSTM usando el método de la ventana para las partículas de dióxido de azufre en la zona de Background. Los valores reales son en azul, la predicción sobre los datos de entrenamiento es en rojo y la predicción sobre los datos test en rosa.

	Regresión	Método de la Ventana
Óxido nítrico	20,20	23,49
Dióxido de nitrógeno	7,22	20,68
Óxidos de nitrógeno	22,07	23,78
Ozono	5,1	4,67
Partículas PM10	4,87	4,81
Partículas PM2.5	4,44	4,46
Dióxido de azufre	2,71	2,81

Tabla 5.18: Tabla con el error cuadrático para los métodos de LSTM en la zona de Roadside.

	Regresión	Método de la Ventana
Óxido nítrico	7,78	8,33
Dióxido de nitrógeno	5,32	5,18
Óxidos de nitrógeno	12,44	13,29
Ozono	7,99	7,63
Partículas PM10	4,35	4,4
Partículas PM2.5	3,73	3,66
Dióxido de azufre	1,14	1,08

Tabla 5.19: Tabla con el error cuadrático para los métodos de LSTM en la zona de Background.

Capítulo 6

Conclusiones

En este último capítulo vamos a ver las conclusiones que nos han aportado el estudio de las distintas partículas.

Vamos a estudiar cuales de ellas nos dan mejores resultados por zona y por partícula. Para ello, vamos a hacer uso de las tablas comparativas del error cuadrático medio.

Los pasos a seguir van a ser estudiar para cada técnica que modelo nos ha dado mejor resultado.
Una vez hayamos visto cual es el mejor modelo de cada técnica, veremos cual es la mejor técnica para esa partícula. De igual forma, vamos a ver que técnica y modelo sería mejor para cada zona.

Vamos a empezar por la zona de Roadside y vamos a ver para cada partícula que técnica nos ha dado un mejor resultado.

Si observamos la tabla 6.1, vemos que para la técnica de Descomposición estacional, la mejor predicción para todas las partículas, y por tanto para la zona de Roadside también, es la que obtuvimos con el método STL.
Si recordamos, lo que hicimos fue descomponer las distintas componentes de la serie para realizar su predicción por separado y tras eso volver a componer la serie predecida.

El segundo método que mejor predice para la zona es con el método HP, seguida del método autorregresivo y por úlitmo el método de persistencia.

	Persistencia	Autorregresivo	HP	STL
Óxido nítrico	8,01	15,99	13,15	6,83
Dióxido de nitrógeno	5,35	7,4	4,15	3,12
Óxidos de nitrógeno	16,51	23,27	16,46	10,77
Ozono	5,56	4,1	4,13	3,11
Partículas PM10	4,93	4,67	4,20	3,2
Partículas PM2.5	4,45	4,26	3,41	2,9
Dióxido de azufre	3,02	2,69	2,48	1,92

Tabla 6.1: Tabla con el error cuadrático medio para todas las partículas en la zona de Roadside, con los métodos de descomposición estacional.

Vamos a ver ahora la técnica de alisamiento exponencial. Como con esta técnica tenemos varios métodos con varias pruebas variando los parámetros, primero vamos a ver que prueba es mejor para cada partícula en cada método.
Una vez tengamos establecido esto, veremos cual es el mejor método para cada partícula. Para esto, usaremos las tablas que ya vimos en la sección 5.
También veremos que método es mejor para la zona completa.

Empezamos por el método de alisamiento exponecial simple. Si observamos la tabla 5.9, vemos que todas las partículas salvo los óxidos de nitrógeno, tienen su mejor aproximación cuando el valor $\alpha = 0.8$.
Los óxidos de nitrógeno la tienen para un α optimizado por la función.
Por tanto, para la zona de Roadside con este método también tendrá una mejor aproximación cuando tomamos $\alpha = 0.8$.

Seguimos por el método de Hölt. Para ello vamos a ver los valores de la tabla 5.11. Observamos que en general para todas las partículas, y por consiguiente para la zona de Roadside, obtenemos los mejores valores con la función lineal amortiguada.
Para el óxido nítrico y el dióxido de azufre se obtiene un menor error con la función exponencial.

Continuamos con el método de Hölt y Winter en la versión aditiva. Podemos ver los resultados que hemos obtenido en la tabla 5.13.
La función aditiva amortiguada obtiene un menor error para la mayoría de las partículas, por tanto también para la zona de Roadside.
La función aditiva nos da unos mejores resultados para el dióxido de nitrógeno y para las partículas PM10 y PM2.5.

Finalmente, vemos los resultados del método de Holt y Winter en la ver-

sión multiplicativa. Los resultados del error los vemos en tabla 5.15.
Como resultado vemos que la función multiplicativa nos da un mejor resultado para la mayoría de las partículas. Si nos fijamos en el óxido nítrico, los óxidos de nitrógeno y el ozono, vemos que tienen un menor error con la función multiplicativa amortiguada.
Por tanto, la zona de Roadside también tendrá una mejor predicción usando la versión multiplicativa.

Finalmente, para el alisamiento exponencial vamos a ver la tabla 6.2 en la que comparamos los mejores resultados de cada modelo y cada partícula.

Podemos ver que no hay una mayoría para la que un método sea mejor que otro.
El método de alisamiento exponencial simple es el mejor para el óxido nítrico y el dióxido de azufre.
Para el dióxido de azufre y los óxidos de nitrógeno el método que tiene un menor error es el de Hölt.
Para el ozono y las partículas PM10, obtenemos la mejor aproximacióna a los valores reales con el método de Hölt y Winder aditivo. Finalmente, las partículas PM2.5 tienen una mejor predicción con el método de Hölt y Winter multiplicativo.

No podemos decir que un método sea mejor que otro para la mayoría de las partículas en la zona de Roadside, pero si vemos el error medio que produce cada método, el mejor sería el del alisamiento exponencial simple.

	Simple	Holt	Holt-Winter A.	Holt-Winter M.
Óxido nítrico	17,12	24,9	50,99	45,97
Dióxido de nitrógeno	8,65	5,33	9,19	9,1
Óxidos de nitrógeno	30,86	27,49	43,68	39,9
Ozono	7,97	15,05	3,46	3,47
Partículas PM10	4,34	4,06	3,87	4,27
Partículas PM2.5	3,96	3,71	3,2	3,02
Dióxido de azufre	3,08	3,21	3,32	3,15

Tabla 6.2: Tabla con el error cuadrático medio para todas las partículas en la zona de Roadside, con los métodos de alisamiento exponencial.

Para la técnica de ARIMA. solo hemos hecho una prueba con la mejor aproximación que nos recomendaba el algoritmo. Por tanto, solo usaremos los valores del error de la tabla 5.17 para la zona de Roadside más adelante.

Nos queda por ver los resultados de la técnica LSTM.
Con esta técnica hemos hecho dos aproximaciones y vamos a ver cual es mejor para partícula.
Nos centramos en la tabla 5.18 y vemos que la mayoría de las partícula obtienen unos mejores resultados con el método de regresión, y por tanto para la zona de Roadside con esta técnica.
Para el ozono y las partículas PM10, se ha obtenido un error menor con el método de la ventana.

Vamos a terminar de ver los resultados para Roadside. Para ello, usamos la tabla 6.3.

Finalmente, vemos que para todas las partículas la técnica que menor error nos ha dado para todas las partículas ha sido la descomposición estacional y todos ellos usando el algoritmo STL para obtener las distintas componentes de la serie.

	Desc.estacional	Alisamiento exp.	ARIMA	LSTM
Óxido nítrico	6,83	17,12	17,66	20,20
Dióxido de nitrógeno	3,12	5,33	10,13	7,22
Óxidos de nitrógeno	10,77	27,49	47,07	22,07
Ozono	3,11	3,46	3,42	4,67
Partículas PM10	3,2	3,87	4,64	4,81
Partículas PM2.5	2,9	3,02	4,41	4,44
Dióxido de azufre	1,92	3,08	3,09	2,71

Tabla 6.3: Tabla con el error cuadrático medio para todas las partículas en la zona de Roadside, con los métodos de LSTM.

Vamos ahora a ver la zona de Background. Vamos a seguir el mismo procedimiento hecho para la otra zona.

Empezamos viendo la tabla 6.4, que nos muestra los errores cuadráticos medio para la técnica de Descomposición estacional.
Al igual que ocurría con la otra zona, la mejor predicción para todas las partículas, y por tanto también para la zona de Background, es la que obtuvimos con el método STL.

Con los demás métodos se comportan también igual ambas zonas. El segundo mejor método para la zona es el método HP, seguida del método autorregresivo y por úlitmo el método de persistencia.

	Persistencia	Autorregresivo	HP	STL
Óxido nítrico	4,46	10,45	7,12	3,81
Dióxido de nitrógeno	5,16	4,17	3,63	2,63
Óxidos de nitrógeno	11,61	14,07	11,81	7,3
Ozono	8,22	7,13	6,12	4,7
Partículas PM10	4,92	4,18	3,72	3,17
Partículas PM2.5	4,33	3,68	3,32	2,65
Dióxido de azufre	1,03	1,06	0,71	0,64

Tabla 6.4: Tabla con el error cuadrático medio para todas las partículas en la zona de Background, con los métodos de descomposición estacional.

Seguimos con la técnica de alisamiento exponencial. Vamos a hacer el mismo proceso que hemos hecho con la zona de Roadside.

Empezamos por el método más simple, el alisamiento exponecial simple. Si observamos la tabla 5.10, vemos que todas la mayoría de las partículas tienen su mejor aproximación cuando el valor α es optimizado.
El ozono y las partículas PM10 tienen una mejor predicción cuando $\alpha = 0.8$. La única partícula que se ajusta mejor su predicción siendo el valor de $\alpha = 0.2$ es el dióxido de azufre.
Por tanto, para la zona de Background en general, también tendrá una mejor aproximación cuando tomamos el valor de α optimizado por la función.

Vamor a ver ahora los resultado obtenidos por el método de Hölt. Para ello vamos a ver los valores de la tabla 5.12. Vemos que para la mayoría de las partículas obtenemos una mejor aproximación a los valores reales con las funciones lineales amortiguadas.
A pesar de ello, para el óxido nítrico, los óxidos de nitrógeno y las partículas PM10, se obtiene un menor error con la función exponencial.

Continuamos con el tercer método que probamos, el método de Hölt y Winter en la versión aditiva.
Podemos ver los resultados que hemos obtenido en la tabla 5.14.La función aditiva es la que nos da un menor error para la mayoría de las partículas, lo que implica que también nos da un menor error para la zona de Background. Sin embargo, la función aditiva amortiguada nos da unos mejores resultados para el óxido nítrico y para el dióxido de azufre.

Vemos los resultados de la versión multiplicativa del método de Holt y Winter en la versión aditiva. Los resultados del error los vemos en tabla 5.16.

Como resultado vemos que la función multiplicativa nos da un mejor resultado para la mayoría de las partículas y por tanto también para la zona de Background.
La función multiplicativa amortiguada da mejores resultados para el dióxido de azufre y las partículas PM10 y PM2.5.

Finalmente, para el alisamiento exponencial vamos a ver la tabla 6.5 en la que comparamos los mejores resultados de cada modelo y cada partícula.

Para la mayoría de las partículas, el método con el que tienen un menor error es el de Hölt y Winter multiplicativo, por tanto también lo será para la zona de Background.
Para el óxido nítrico obtenemos un menor error con el método de alisamiento exponencial simple y para el ozono el mejor método es el de Hölt y Winter aditivo.

	Simple	Holt	Holt-Winter A.	Holt-Winter M.
Óxido nítrico	5,85	6,32	13,41	9,08
Dióxido de nitrógeno	6,97	6,47	2,91	2,78
Óxidos de nitrógeno	15,2	16,35	14,33	10,2
Ozono	11,22	16,52	5,89	6,18
Partículas PM10	4,04	4,02	3,27	3,23
Partículas PM2.5	3,64	3,59	3,03	2,89
Dióxido de azufre	1,54	1,7	1,48	1,47

Tabla 6.5: Tabla con el error cuadrático medio para todas las partículas en la zona de Background, con los métodos de alisamiento exponencial.

Como ya hemos comentado con la otra zona, para la técnica de ARIMA. solo hemos hecho una prueba con la mejor aproximación que nos recomendaba el algoritmo. Por tanto, solo usaremos los valores del error de la tabla 5.17 para la zona de Roadside más adelante.

Vemos los resultados de la última técnica.
Con LSTM hemos hecho dos aproximaciones y vamos a ver cual es mejor para partícula.
Nos centramos en la tabla 5.19 y vemos que la mayoría de las partícula obtienen un menor error cuadrático medio con el método de la ventana. Por tanto, la zona de Background tendrá un menor error con este método.
El óxido nítrico, los óxidos de nitrógeno y las partículas PM10, obtienen mejores resultados con el método de regresión.

Vamos a terminar de ver los resultados para Background. Para ello, usamos la tabla 6.6.

Finalmente, vemos que para todas las partículas la técnica que menor error nos ha dado para todas las partículas ha sido la descomposición estacional y todos ellos usando el algoritmo STL para obtener las distintas componentes de la serie.

	Desc.estacional	Alisamiento exp.	ARIMA	LSTM
Óxido nítrico	3,81	5,85	21,43	7,78
Dióxido de nitrógeno	2,63	2,78	4,58	5,18
Óxidos de nitrógeno	7,3	10,2	13,97	12,44
Ozono	4,7	5,89	6,85	7,63
Partículas PM10	3,17	3,23	4,56	4,35
Partículas PM2.5	2,65	2,89	3,65	3,66
Dióxido de azufre	0,64	1,47	1,44	1,08

Tabla 6.6: Tabla con el error cuadrático medio para todas las partículas en la zona de Background, con los métodos de LSTM.

Bibliografía

[1] Vassilis Assimakopoulos and K. Nikolopoulos. The theta model: A decomposition approach to forecasting. *International Journal of Forecasting*, 16:521–530, 10 2000.

[2] Ricardo José Canales Salinas and Eleonora del Socorro Rodríguez Alonso. Estimaciones alternativas del pib potencial de nicaragua. *Revista Electrónica de Investigación en Ciencias Económicas*, 15(0):6–7, 2013.

[3] Robert B. Cleveland, Wiliam S. Cleveland, Jean E. McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal os Official Statistics*, 31(0):1–9, 1990.

[4] Emily Connolly, Gary Fuller, Timothy Baker, and Paul Willis. Update on implementation of the daily air quality index. *Department for Environment Food & Rural Affairs*, 14, 2013.

[5] Marcel de Matas, Qun Shao, Martyn F.Biddiscombe, Sally Meah, Henry Chrytin, and Omar S.Usman. Predicting the clinical effects of a short acting bronchodilator in individual patients using artificial neural networks. *European Journal of Pharmaceutical Sciences*, pages 707–715, 2010.

[6] Agencia de Protección Ambiental. Ozono. Biblioteca Nacional de Medicina de los EE.UU.

[7] Michel Denuit, Donatien Hainaut, and Julien Trufin. *Effective Statistical Learning Methods for Actuaries III*. Springer, 2019.

[8] José Ángel Gallardo San Salvador. cluster-3.pdf, 2020/2021. Universidad de Granada.

[9] Greater London Authority. Air quality in london 2016-2020, 2020.

[10] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts: Melbourne, Australia., 2018.

[11] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts: Melbourne, Australia., 2018.

[12] Envira IOT. ¿qué son las partículas pm10 y pm2.5?, 2019.

[13] P.G. Kou-llapis, F.S. Stylianou, B. Olsson, and S.C. Kassinos. Towards whole-lung simulations of aerosol deposition: A model of the deep lung. *Journal of Aerosol Science*, pages 1–17, 2020.

[14] Antonios Lalas, Stavros Nousias, Dimis trios Kikidis, Aris Lalos, Gerasimos Arvanitis, Christos Sougles, KonstantinosMoustakas, Kontantinos Votis, Sylvia Verbanck, Omar Usmani, and Dimitros Tzo-varas. Substance deposition assessment in obstructed pulmonary system through numerical characterization of airflow and inhaled particles attributes. *BMC Medical Informatics and Decision Making*, pages 25–44, 2017.

[15] Jerónimo Lorente Pardo. ajuste5.pdf, 2004. Universidad de Granada.

[16] Beatriz E. López Porrero. *Limpieza de datos*. Editorial Feijoo, 2009.

[17] Andrés Mañas Mañas. Notas sobre pronóstico del flujo de tráfico en la ciudad de madrid, 2019.

[18] Francisco Montes Abad. Desestacionalización. Universidad de Granada.

[19] Peter Norving and Stuart J. Russell. *Inteligencia Artificial: Un Enfoque Moderno*. Prentince Hall, 2009.

[20] Instituto para la Salud Geoambiental. El dióxido de azufre so2.

[21] Agencia para Sustancias Tóxicas y el Registro de Enfermedades. Toxfaqs™ - Óxidos de nitrógeno (monóxido de nitrógeno, dióxido de nitrógeno, etc.) (nitrogen oxides).

[22] María del R. Prieto. Contaminación en los siglos xviii y xix.

[23] Germán Pérez Aneiros. Tema1.pdf, 2008/2009. Universidad de Coruña.

[24] María Jesús Ramírez García-Ligero, Aurora Hermoso Carazo, Juan Antonio Maldonado Jurado, Patricia Román Román, and Francisco Ruiz Torres. $P_t04_desigualdadbasica.pdf.UniversidaddeGranada$.

[25] María Jesús Ramírez García-Ligero, Aurora Hermoso Carazo, Juan Antonio Maldonado Jurado, Patricia Román Román, and Francisco Ruiz Torres. $P_t04_desigualdadchebychev.pdf.UniversidaddeGranada$.

[26] Omar S.Usmai, Martyn F.Biddiscombe, and Peter J. Barnes. Regional lung deposition and bronchodilator response as a function of β2-agonist particle size. *American journal of respiratory and critical care medicine*, pages 1497–1504, 2005.

[27] Omar S. Usmani, Martyn F. Biddiscombe, and Peter J. Barnes. No evidence that electric charge increseases inhaled ultrafine particle deposition in human lungs. *National Library of Medicine*, pages 25–44, 2020.

[28] Sylvia Verbanck, Biddiscombe, Martyn F., Usmani, and Omar S. Inhaled aerosol distribution between proximal bronchi and lung periphery. *European Journal of Pharmaceutics and Biopharmaceutics*, pages 18–22, 2020.

[29] Sylvia Verbanck, Ghader Ghorbaniasl, Martyn F.Diddiscombe, Dusica Dragojlovic, Nathan Ricks, Chris Lacor, Bart Ilsen, Johan de Mey, Daniel Schuermans, S Richard Underwood, Peter J Barnes, Walter Vincken, and Omar S Usmani. Inhaled aerosol distribution in human airways: A scintigraphy-guided study in a 3d printed model. *Journal of Aerosol Medicine and Pulmonary Drug Delivery*, 29(0):525–533, 2016.