Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Predicción de los niveles de polución atmosférica en Tenerife mediante técnicas de Machine Learning

Forecasting air pollution in Tenerife with Machine Learning models

Carlos Domínguez García

La Laguna, 10 de septiembre de 2019

D. **Jesús Manuel Jorge Santiso**, con N.I.F. 42.097.398-S profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

”*Predicción de los niveles de polución atmosférica en Tenerife mediante técnicas de Machine Learning*”

ha sido realizada bajo su dirección por D. **Carlos Domínguez García**, con N.I.F. 42.238.865-D.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 10 de septiembre de 2019

Agradecimientos

Agradezco a todas las personas que me han apoyado.

Y agradezco especialmente a las personas que comparten su conocimiento en Internet pues sin ellos la educación que tendría ahora sería bastante distinta.

Licencia

Resumen

El objetivo de este trabajo ha sido realizar predicciones de la concentración del contaminante conocido como PM2.5, se trata de polvo en suspensión de un diámetro menor a 2.5 micrómetros. El proyecto se centra en la zona de Santa Cruz de Tenerife usando datos históricos de contaminantes de la atmósfera y de variables meteorológicas para entrenar modelos de aprendizaje automático. Así se pretende contribuir al desarrollo de un plan preventivo de la calidad del aire para avisar a la población en caso de que se prevea un nivel de contaminación potencialmente dañino a la salud.

Como parte del proyecto se ha desarrollado un software para la extracción de datos relativos a la contaminación en Tenerife. Se ha usado PostgreSQL como base de datos relacional para almacenar estos datos. Se han construido modelos de aprendizaje automático usando Python como lenguaje de programación. Y se ha desarrollado una aplicación web para mostrar a los usuarios las últimas medidas en el sistema y predicciones de valores futuros.

Palabras clave: Contaminación, Tenerife, Aprendizaje Automático

Abstract

The goal of this project was to forecast the concentration of the pollutant known as PM2.5, which are airborne particles with a diameter of 2.5 micrometers or lower. The project is focused on the area of Santa Cruz de Tenerife using past data of air pollutants and weather variables to train machine learning models. This is meant to contribute to the development of an air quality prevention plan to warn the population if the pollution level is predicted to be dangerous.

As part of the project a software was developed for the automated extraction of data related to Tenerife air. The relational database PostgreSQL was used to store this data. Machine learning models were trained using the programming language Python. And a web application was developed to show users the most recent data in the system and forecasts for future values of PM2.5.

Keywords: Pollution, Tenerife, Machine Learning

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Motivación

La presencia de materia intrusa en la atmósfera puede resultar dañina para los seres vivos: cuando una persona inhala partículas de monóxido de carbono se disminuye la capacidad del cuerpo de transportar oxígeno y en caso de estar expuesto a una fuente continua puede producir la muerte en minutos [1]. Otro ejemplo son las partículas en suspensión que pueden perforar el sistema respiratorio y cardiovascular siendo responsable de 7 millones de muertes al año según la Organización Mundial de la Salud (OMS) [2].

Aunque las legislaciones difieren entre países, la contaminación de la atmósfera es un peligro para todos porque, una vez en la atmósfera, los contaminantes pueden viajar por acción del viento a otras regiones. Por ello, la OMS provee unas directrices generales acerca de los límites de concentración de contaminantes clave en el riesgo de la salud pública [3]. Actualmente los contaminantes propuestos por la OMS son: las partículas en suspensión, que son una mezcla de pequeñas partículas en estado sólido o líquido como hollín, polvo o agua que se encuentran suspendidas en el aire y que en Canarias se conoce bien debido a las constantes intrusiones de polvo africano. El ozono, dióxido de nitrógeno y dióxido de azufre que son gases que irritan las vías respiratorias y pueden provocar problemas respiratorios como asma, además los dos últimos son los principales causantes de la lluvia ácida [4].

Para asegurarse que las concentraciones de los contaminantes en la atmósfera no superan los límites legales se usan datos de sensores que miden estas concentraciones para llevar a cabo procesos de control y de mitigación si fuera necesario. Por tanto, poder predecir las concentraciones de los contaminantes supondría poder reaccionar antes de que la calidad del aire sea peligrosa.

1.2. Objetivos

El trabajo tiene tres grandes objetivos:

- Extraer y almacenar datos sobre la concentración de contaminantes en la atmósfera y datos meteorológicos de Tenerife.
- Desarrollar modelos de predicción de la concentración media de PM2.5 en Tenerife en un período de 24 y 48 horas.
- Desarrollar una aplicación web que permita consultar los últimos valores registrados y los resultados de las predicciones.

1.3. Antecedentes

- En la competición KDD Cup [1] de 2018 se planteó un problema similar al que se trata en este proyecto. En esta competición los participantes tuvieron acceso a datos históricos de Londres y Pekín de variables meteorológicas y de concentraciones de contaminantes con el objetivo de predecir la concentración de ciertos contaminantes a las 48 horas siguientes [5]. El equipo ganador usó de técnicas de aprendizaje automático para lograr dicho objetivo, en concreto de *gradient boosting* y de árboles de regresión [6]. El segundo equipo también usó técnicas de aprendizaje automático, en su caso distintas arquitecturas de redes neuronales artificiales [7].
- En [8] se ha hecho uso de una variante de red neuronal para la predicción del índice de calidad del aire de las partículas en suspensión de 2.5 micrómetros o menos. Esto se hizo usando datos de contaminantes y de valores atmosféricos de ciudades de Corea del Sur para unos períodos mayores o iguales a 8 horas (8, 12, 16, 20 y 24 horas) pues vieron que con períodos menores a 5 horas los valores no fluctuaban mucho. Como resultado obtuvieron un modelo capaz de adaptarse bastante bien a la realidad mostrando como prueba de ello gráficas comparando las predicciones con los valores reales entre otras pruebas.
- En [9] se ha hecho uso de otra variante de red neuronal para la predicción de la concentración de las partículas en suspensión de 2.5 micrómetros o menos. Aquí se usaron datos de Pekín y se consiguió predecir la concentración del contaminante para la siguiente hora usando sólo datos históricos del mismo, de la velocidad del viento y de las precipitaciones durante las últimas 24 horas.

[1] La KDD Cup (Knowledge Discovery in Databases Cup) es una competición anual organizado por la ACM (Association for Computing Machinery) donde los participantes tienen acceso a un conjunto de datos y tienen que extraer conocimiento de ellos

Capítulo 2

Fundamentos teóricos

2.1. Almacenamiento de datos

Sin bases de datos para almacenar datos de manera persistente se ha de interactuar directamente con un sistema de almacenamiento. Esto implica que las aplicaciones, para poder usar los datos, tengan que implementar operaciones CRUD (Create, Read, Update, Delete) sobre el sistema de almacenamiento. Por ejemplo, interactuando a través de un sistema de archivos, las aplicaciones tienen que encargarse de abrir o crear los archivos, entender el formato del contenido, actualizar el contenido escribiendo en el fichero y cerrarlo.

Esta complejidad añadida de las aplicaciones abre la posibilidad a una mayor cantidad errores en su código. Pero también, debido la falta de una manera estándar de acceder a los datos, cada aplicación podría implementar el manejo de los datos de manera distinta provocando redundancia si aplicaciones que necesitaban los mismos datos los almacenaban en distintos ficheros, inconsistencia si estos datos duplicados unos se actualizaban y otros no, más problemas de inconsistencia si no se garantiza que las transacciones sean atómicas, es decir, o tienen lugar todas las operaciones asociadas a la transacción en el orden previsto o no tiene lugar ninguna, etc. [10].

Para resolver esta situación surgieron los sistemas gestores de bases de datos. Este tipo de sistema se sitúa entre los usuarios y el sistema de almacenamiento, proporcionando una estructura lógica de los datos y una interfaz que permite a los usuarios referirse a los datos según este modelo de datos. De esta manera pueden usar una interfaz con un alto nivel de abstracción sin tener que preocuparse de los problemas relacionados con interactuar directamente con los ficheros ya que el sistema también se encarga de ello.

En 1970 Edgar F. Codd introdujo el modelo relacional de datos, uno de los más conocidos actualmente. Este modelo permite una representación de los datos más flexibles que los que había hasta la fecha. En palabras de Codd: “It provides a means of describing data with its natural structure only–that is, without superimposing any additional structure for machine representation purposes.” [11].

Es habitual ver bases de datos relacionales usadas para procesar transacciones [1]. El modelo de datos en un sistema con este fin ha de tener una redundancia mínima de datos para así necesitar menos espacio de almacenamiento, tener una mayor robustez a la inconsistencia y que solo haya que actualizar valores en un sitio concreto, haciendo que una base de datos relacional con un modelo normalizado sea bastante adecuada para este fin.

Estos sistemas son conocidos como sistemas OLTP por las siglas en inglés de procesamiento de transacciones en línea (*Online Transaction Processing*) y suelen mantener el estado actual de un proceso sin guardar datos históricos. Sin embargo no son los más adecuados para realizar análisis de datos porque haría falta tener datos históricos y realizar consultas muy complejas que en un modelo normalizado requeriría unir muchas tablas. Esto implica que haga falta una gran cantidad de tiempo para las consultas, añadiendo una carga importante al sistema y puede que incluso no se pueda realizar en muchos casos ya que las organizaciones suelen tener varias bases de datos en las que guardan información relacionada a distintos procesos y un análisis podría requerir datos de todas ellas e incluso de fuentes externas.

2.1.1. Almacenes de datos

Un almacén de datos es un sistema pensado para realizar análisis en lugar de procesar transacciones, esto también es conocido como un sistema OLAP por las siglas en inglés de procesamiento analítico en linea (*Online Analytical Processing*).

Para implementar un sistema de este tipo también se puede usar una base de datos relacional pero con un modelo de datos desnormalizado. Siguiendo el enfoque de Ralph Kimball, este es un modelo caracterizado por tener tablas de hechos donde se almacenan los datos medidos y tablas de dimensiones que aportan metadatos de los hechos. Este modelo de datos se conoce como modelo en estrella [13] debido a la forma que le confiere tener sólo relaciones entre tablas de hechos y dimensiones como se puede apreciar en el ejemplo presente en la figura 2.1.

Aparte de la velocidad de las consultas en comparación con un modelo normalizado, otra gran ventaja de este modelo es la simplicidad de las consultas al tener menos tablas que en un modelo normalizado y solo necesitar uniones entre tablas de hechos y de dimensiones.

[1] Una transacción es un conjunto de operaciones que respetan las propiedades *ACID (Atomicity, Consistency, Isolation, Durability)*, es decir, se han de realizar de manera que ocurren todas seguidas o no ocurre ninguna, en todo momento respetan las restricciones de la base de dato, en caso de haber varias transacciones de manera concurrente el resultado será como si ocurrieran de manera secuencial (los cambios por una no se ve por la otra) y una vez termina una transacción todos los cambios son registrados, no se pierden. [12]

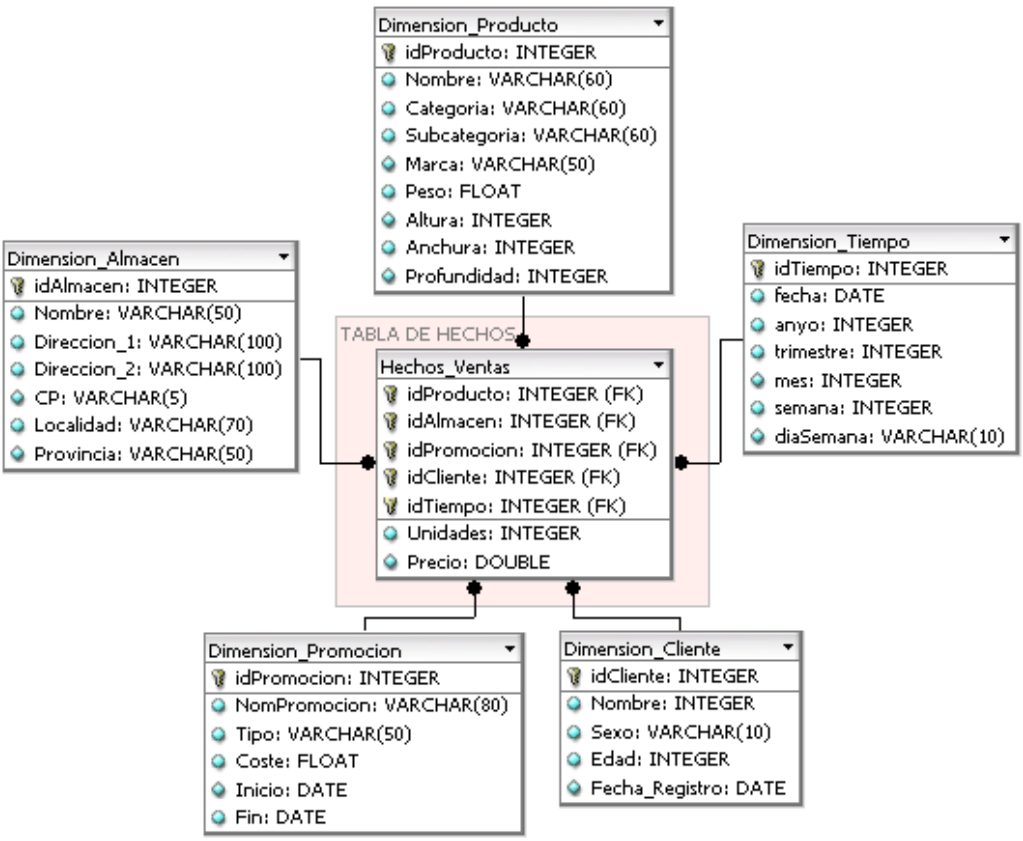


Figura 2.1: Ejemplo de un esquema en estrella

Se puede observar un modelo de datos con una tabla *Hechos_Ventas* donde se almacenan las unidades y precio de cada venta de una organización y que está relacionada con cinco tablas de dimensiones que aportan datos extras para describir las ventas. Estos son el tipo de producto vendido, la fecha de la venta, el cliente que lo compró, el tipo de promoción aplicado y el almacén donde se localizaba el producto.

Jesuja [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)]

2.2. Aprendizaje automático

El aprendizaje automático (más conocido por su nombre en inglés *Machine Learning*) es un campo que se centra en algoritmos capaces de aprender a realizar una tarea dado un conjunto de datos. Dos grandes áreas de este campo son:

- **El aprendizaje supervisado** que se centra en modelos capaces de predecir el valor de una variable objetivo a partir de datos históricos.
- **El aprendizaje no supervisado** que se centra en modelos capaces de encontrar patrones en el conjunto de datos dado.

En este proyecto nos centramos en el aprendizaje supervisado aplicado a tareas de regresión (predecir valores en el dominio de los números reales) y dentro del aprendizaje no supervisado nos centraremos en la tarea de reducción de la dimensionalidad para el preprocesamiento de los datos con el algoritmo que se presenta a continuación.

2.2.1. Análisis de componentes principales

El análisis de componentes principales, conocido como *PCA* por sus siglas en inglés es una técnica de reducción de la dimensionalidad de un conjunto de datos. Como se puede apreciar en la figura 2.2, esta técnica busca representar los datos en un nuevo sistema de coordenadas de manera que se elimine la redundancia que supone tener varios atributos correlacionados. Y, una vez en este sistema de coordenadas, se puede reducir la dimensionalidad proyectando los datos al subespacio generado por los vectores que retienen la mayor parte de la información de los datos originales.

Matemáticamente se puede comenzar a plantear este problema de encontrar los componentes principales por encontrar el vector que minimicen la distancia entre los puntos y la proyección de los mismos. Para el segundo componente sería encontrar el vector perpendicular al primer componente principal que minimice la distancia entre los puntos y su proyección en el segundo vector, etc.

De manera equivalente se puede plantear buscar los vectores de manera que se maximice la dispersión de la proyección de los datos en los correspondientes vectores. Se puede apreciar intuitivamente que estas dos formulaciones son equivalentes viendo el ejemplo en la figura 2.2, donde el primer componente principal es el que se encuentra en la dirección de mayor dispersión y además es el de mínima distancia entre los puntos originales y sus proyecciones en este vector.

Sea $X \in \mathbb{R}^{n \times d}$ el conjunto de n datos con d atributos con media cero (si alguna columna de atributos no tiene una media de cero siempre se puede conseguir restándole la media) y $v \in \mathbb{R}^{d \times 1}$ el primer componente principal que se quiere encontrar. Entonces la varianza de los datos proyectados en v es

$$\frac{1}{n}(Xv)^\top (Xv) = v^\top \frac{X^\top X}{n} v = v^\top C v$$

Donde C es la matriz de covarianza de X. Si incluimos como restricción que v sea un vector unitario mediante un multiplicador de Lagrange y derivamos obtenemos

$$\begin{aligned} L(v, \lambda) &= v^\top C v - \lambda(v^\top v - 1) \\ \frac{\partial L}{\partial \lambda} &= v^\top v - 1 \\ \frac{\partial L}{\partial v} &= 2Cv - 2\lambda v \end{aligned}$$

Finalmente, igualando las derivadas a cero para obtener el máximo tenemos

$$\begin{aligned} v^\top v &= 1 \\ Cv &= \lambda v \end{aligned}$$

Esto último es precisamente la definición de *autovector*: vector que al aplicarle una una transformación resulta en el mismo vector escalado por un factor. Por lo que encontrar

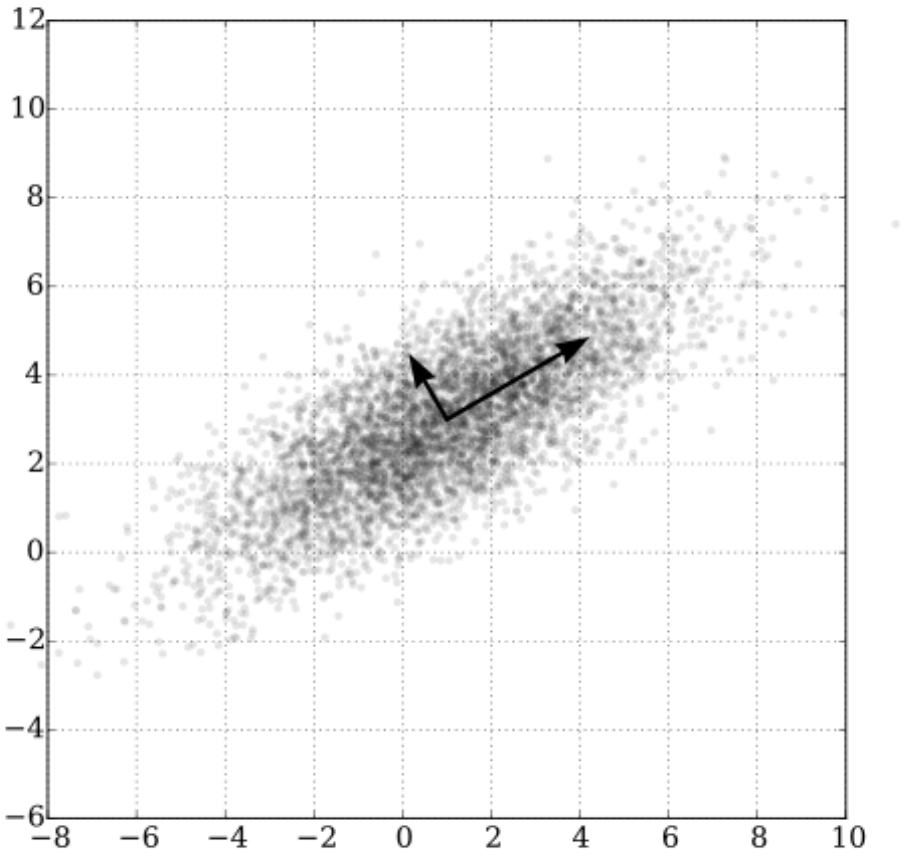


Figura 2.2: Ejemplo de Componentes Principales

Se puede apreciar en el gráfico un conjunto de datos con dos atributos que presentan una alta correlación positiva. Además se pueden ver las dos componentes principales representadas como dos flechas negras, el primer componente principal es el que cubre la dirección con mayor dispersión (y por tanto, el que retiene la mayor parte de la información original de los datos) mientras que el segundo componente principal es el que retiene la menor parte de la información original de los datos y, por tanto, sería el más sensato de descartar para reducir la dimensionalidad de los datos

las componentes principales se reduce a encontrar los autovectores de la matriz de covarianza.

2.2.2. Regresión lineal

La regresión lineal es un modelo sencillo que, dada una entrada con d atributos, el resultado se obtiene de una suma de cada valor (x_i) multiplicado por un factor (β_i) mas un peso extra independiente (β_0) de la entrada que sirve para permitir que el modelo generalize mejor al no tener que obligar al hiperplano que representa a pasar por el origen del sistema de coordenadas.

$$\hat{y} = \beta_0 + \sum_{j=1}^{d} \beta_j x_j$$

También podemos representarlo de manera más compacta en forma matricial añadiendo una dimensión extra para representar el término independiente (y que ha de ser siempre uno en x)

$$\hat{y} = x\beta \qquad x \in \mathbb{R}^{1\times(d+1)} \qquad \beta \in \mathbb{R}^{(d+1)\times 1}$$

Para entrenar un modelo de regresión lineal es común usar como función de error la suma de errores cuadrados

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

Donde n es el número de datos de entrenamiento. Se puede obtener una solución analítica a la minimización del error:

$$\beta = (X^\top X)^{-1} X^\top Y \qquad \beta \in \mathbb{R}^{(d+1)\times 1} \qquad X \in \mathbb{R}^{n\times(d+1)} \qquad Y \in \mathbb{R}^{n\times 1}$$

Note que si dos atributos resultan ser el mismo entonces las filas de $X^\top X$ no serían linealmente independientes y por tanto no se podía invertir. Otro posible problema surge cuando se tiene una gran cantidad de datos y/o atributos pues la multiplicación e inversión de $X^\top X$ es costosa computacionalmente. Por eso, cuando se tiene un número considerable de atributos y/o datos se suelen usar métodos iterativos como el descenso del gradiente para llegar a un buen óptimo local en un tiempo razonable.

2.2.3. Árbol de Decisión

La representación de este modelo es un árbol binario donde el nodo raíz y cada nodo intermedio representan un atributo y un valor por el que dividir el conjunto de datos en

dos. Y los nodos hojas contienen los posibles resultados del modelo (los cuales son valores constantes en cada región).

Gráficamente los árboles dividen el espacio de entrada en rectángulos (o hiperrectángulos si el número de atributos es mayor a dos) y cada región tiene asociada una constante como valor de salida.

Se puede observar un ejemplo genérico de árbol de decisión en la figura 2.3 además de un ejemplo de división del espacio de entrada.

En este tipo de modelo se tiene que aprender es la topología del árbol, lo que implica aprender los atributos y valores sobre los que dividir el espacio de entrada. Para ello existen diversos algoritmos pero nos vamos a centrar en *CART (Classification and Regression Tree)* pues es el implementado en la librería que se ha usado para el proyecto, *scikit-learn* [15]

Para entrenar el modelo se elige la función de error que, en el caso de regresión, es común la suma de errores cuadrados ($\sum_{i=1}^{n} (y_i - f(x_i))^2$) y para elegir el atributo y valor de cada nodo se realiza de manera iterativa el procedimiento que se detalla a continuación.

Dado un atributo se busca el mejor valor que divida el espacio de entrada. Para ello se ordenan los valores del conjunto de entrenamiento y se usan como candidatos las medias de cada dos observaciones (así se asegura que el valor siempre va a separar el espacio en dos, teniendo al menos una observación en cada región). Para cada candidato se calcula el error de partición: se asignaría como constante de región la media de los datos de entrenamiento encontrados en la región (porque es el valor que minimiza la suma de errores cuadrados) y el error de la partición sería la suma de errores cuadrados entre los valores y las medias de su región correspondiente. El menor de estos sería el error del atributo. Se realizaría lo mismo con el resto y para el nodo actual se elegiría el atributo y valor de menor error.

Esto se repite para cada nueva partición de datos y se puede dejar terminar cuando en cada partición haya un sólo dato de entrenamiento pero esto lleva a un sobreajuste del modelo a los datos de entrenamiento. Por tanto, es habitual especificar como hiperparámetros el máximo número de nodos hojas, número de observaciones mínimo para considerar un nodo como hoja, etc.

2.2.4. Random Forest

Bagging (Bootstrap aggregating) es una sencilla técnica usada para mejorar la precisión de modelos de aprendizaje automático. Normalmente se usa con árboles de decisión y consiste en entrenar varios árboles, cada uno usando un subconjunto del conjunto de entrenamiento elegido de manera aleatoria con reemplazamiento y dando como resultado la media de los resultados de los árboles (o la moda si se trabajase con un problema de clasificación).

Sin embargo, puede ocurrir que unos atributos sean bastante importante para la

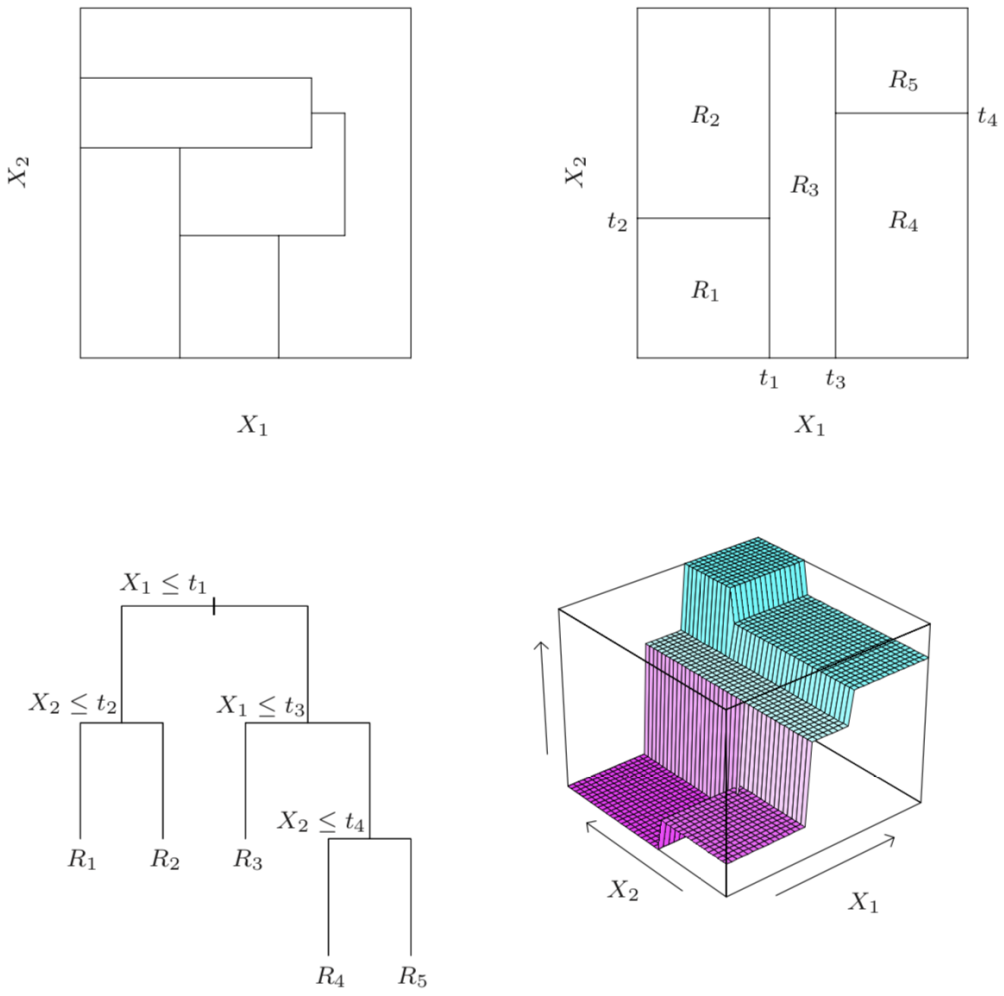


Figura 2.3: Representación de un árbol de decisión

La figura de arriba a la izquierda muestra el espacio de entrada dividido de una manera imposible por un árbol de decisión, mientras que a su derecha se encuentra un ejemplo posible de partición.

La figura de debajo a la izquierda es la representación del árbol como un árbol binario. Finalmente la última figura representa la misma partición del espacio de entrada que la segunda figura pero añadiendo como dimensión el resultado del árbol.

Imagen perteneciente al libro "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" [14]

predicción y formen parte de varios árboles, haciendo que haya una correlación entre ellos. En el caso mas extremo todos los árboles usarían un solo atributo y serían todos iguales, haciendo que el uso de *Bagging* sea indiferente a usar un sólo árbol.

Aquí es donde surge el modelo *Random Forest*, para reducir esta correlación emplea la misma estrategia que *Bagging* pero modificando el algoritmo de aprendizaje de los árboles: para cada nodo no se tendrán en cuenta todos los atributos sino un subconjunto aleatorio de tamaño m, donde m es un hiperparámetro del modelo.

2.2.5. Gradient Boosting

Boosting es otra técnica que se usa para mejorar la precisión de modelos de aprendizaje automático basada en usar múltiples modelos, comúnmente árboles de decisión. En concreto, se basa en construir cada árbol usando información de los anteriores para aprender de sus errores.

De entre los algoritmos de *boosting*, *Gradient Boosting* es uno de los más empleados. Este modelo empieza con la mejor predicción posible sin tener en cuenta ningún otro atributo: la media de la variable objetivo. A continuación se calcula la diferencia entre todas las variables objetivo de entrenamiento y la predicción del modelo. El siguiente paso es entrenar un árbol de decisión usando como variables objetivo las diferencias calculadas. Ahora el modelo está compuesto de una constante mas un árbol de decisión, la predicción del modelo será la suma de la constante mas la salida del árbol multiplicado por un factor para combatir el sobreajuste (este es un hiperparámetro conocido como *learning rate*). Este procedimiento continúa hasta alcanzar un criterio de parada especificado como hiperparámetro del modelo que puede ser, por ejemplo, un número máximo de iteraciones o un número de iteraciones sin mejora.

2.2.6. ARIMA

ARIMA es un modelo que trabaja con los datos en forma de series temporales. Estas son secuencias de observaciones tomadas en intervalos de tiempos regulares y ordenadas cronológicamente. Una serie temporal puede presentar cuatro componentes distintos como se puede apreciar en la figura 2.4:

- **La tendencia** indica un incremento o decremento de la serie a largo plazo.
- **La variación estacional** es un movimiento que ocurre durante un tiempo y en una frecuencia fija. Generalmente debido a factores como la estación del año o el día de la semana.
- **La variación cíclica** es una oscilación de frecuencia no fija que ocurre generalmente durante un período de tiempo mayor a un año.
- **El ruido** es una variación debido a fenómenos aislados y que no presentan ninguna regularidad.

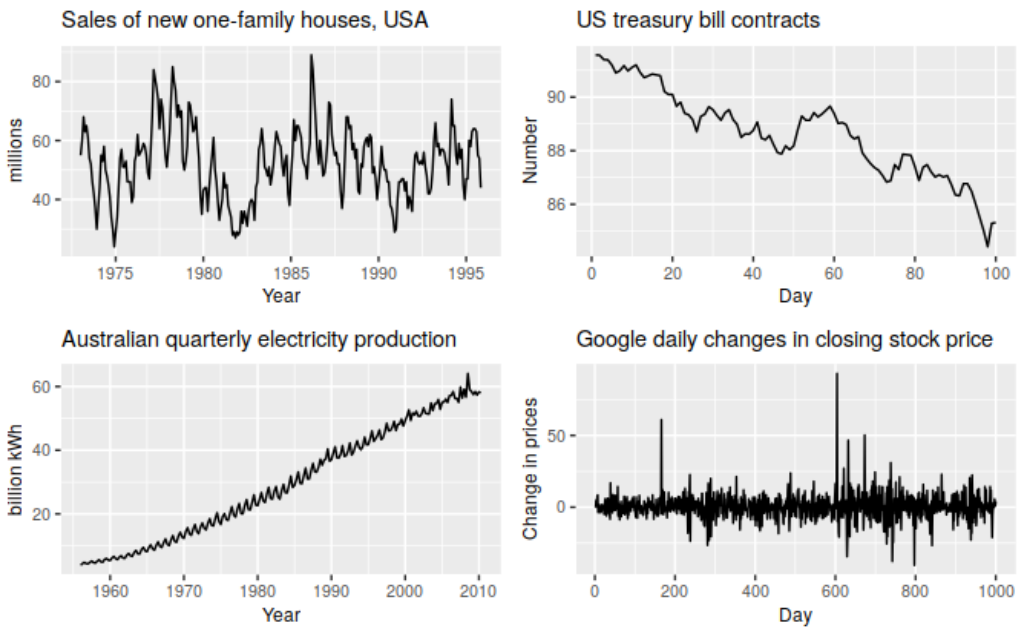


Figura 2.4: Ejemplo de series temporales

- La serie temporal de arriba a la izquierda presenta una variación estacional además de cíclica.
- La serie a su derecha presenta solamente una tendencia decreciente.
- La serie de abajo a la izquierda presenta tanto una variación estacional como una tendencia creciente.
- Finalmente, la última no presenta ningún comportamiento regular.

Imagen perteneciente al libro "Forecasting: Principles and Practice" [16]

ARIMA es un acrónimo de *Autoregressive Integrated Moving Average* que indica precisamente los tres componentes del modelo:

- **Autoregresión:** es un modelo de regresión lineal que usa como entradas los p valores anteriores de la serie temporal

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

  Donde ϵ_t es ruido blanco.

- **Integrado:** es la operación contraria a la diferenciación, la cual es una operación que se realiza para hacer que la serie temporal sea estacionaria [2]. ARIMA espera un hiperparámetro d que es el número de veces que se ha llevado a cabo la diferenciación en la serie temporal.

- **Media móvil:** es un modelo en el que se usa como base la media de la serie temporal y para predecir el siguiente punto se ajusta esta media añadiéndole una combinación lineal de los q errores anteriores de la serie temporal

$$y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}$$

Dados unos valores para los hiperparámetros (p, d, q), ARIMA estima los mejores parámetros mediante máxima verosimilitud.

2.3. Experimentación

El objetivo final de los modelos de predicción es que sean usados para predecir futuros valores en un entorno incierto. Por ello se han de llevar a cabo unas medidas para probar cuán bien los modelos pueden desempeñar su función fuera del entorno de entrenamiento.

2.3.1. Partición del conjunto de datos

Es una buena práctica no usar todo el conjunto de datos disponible para entrenar los modelos sino separarlo en un conjunto de entrenamiento y en otro de testeo. El conjunto de entrenamiento se usará para aprender todo relativo a los datos y al terminar el proyecto se usará el conjunto de testeo para que se evalúen los modelos y así tener una medida fiable de cómo se comportarán con datos nuevos.

Si los modelos a usar tienen hiperparámetros y solo se usara el conjunto de entrenamiento para entrenar y evaluar modelos con distintas configuraciones de hiperparámetros se corre el riesgo de sobreajustarse al conjunto de entrenamiento y, por tanto, que los modelos no desempeñen un trabajo tan bueno con datos nuevos. Por ello es buena

[2] Una serie temporal se dice que es estacionaria si sus propiedades estadísticas no cambian con el tiempo, es decir, presenta una media y varianza constante

práctica separar a su vez el conjunto de entrenamiento en un subconjunto de entrenamiento y en un conjunto de validación donde se evaluarán las distintas configuraciones de hiperparámetros.

Sin embargo, aun así se corre el riesgo de que con un solo conjunto de validación se sobreajuste a los datos pues siempre se estaría evaluando los modelos en el mismo conjunto. Por tanto, en lugar de simplemente particionar el conjunto de entrenamiento en entrenamiento y validación, se puede usar una técnica llamada validación cruzada donde, en su forma mas básica, se especifican k "pliegues" y se usa uno como partición de validación y el resto de entrenamiento. Esto se repite k veces y como métrica final se usa la media de las k métricas.

2.3.2. Evaluación de los modelos

Para evaluar los modelos se les da como entrada una serie de datos de los que se conoce el valor objetivo y se comparan las salidas del modelo con los valores esperados.

Hay diversas funciones para comparar estas diferencias, en el presente proyecto se ha hecho uso de las siguientes:

- **Error absoluto medio:** es la función mas sencilla en la que se calcula la diferencia entre la salida del modelo y el valor esperado. Se hace uso del valor absoluto para evitar que diferencias positivas contrarresten parte de las negativas o viceversa.

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

- **Error cuadrático medio:** similar al error absoluto medio pero se usa el cuadrado de la diferencia en lugar del valor absoluto. Con esto se consigue que se de un peso mayor a diferencias mayores

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

- **Coeficiente de determinación:** es una relación entre la dispersión de los datos alrededor del modelo y alrededor de la media. Si el modelo predice todos los valores correctamente entonces el resultado sería 1. Si el modelo es tan desacertado que la media de los datos sería una mejor predicción entonces el resultado sería 0 o negativo.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

Capítulo 3

Tecnologías

En este capitulo se detallan las tecnologías principales que se han usado en las distintas partes del proyecto.

3.1. Base de datos

Como base de datos relacional existen diversas opciones como Microsoft SQL Server, MariaDB, MySQL, PostgreSQL, etc. Para el proyecto podría haberse usado cualquier solución pues no ha hecho falta nada específico de ninguna base de datos. Se terminó usando PostgreSQL por ser una herramienta de software libre, gratis, respaldada por una comunidad activa, con más de treinta años de desarrollo y por ser bastante usada en empresas porque ofrece una gran cantidad de opciones avanzadas.

3.2. Extracción de datos

El software para la extracción de datos se desarrolló usando JavaScript como lenguaje de programación fuera del navegador gracias a Node.js. Esto se debe a que en este ecosistema está la librería *puppeteer* que permite controlar un navegador mediante código, lo cual fue necesario para poder automatizar el relleno del formulario para descargar los datos como se detalla en el siguiente capítulo.

3.3. Carga de datos

El software para la transformación y carga de los datos en el almacén de datos se desarrolló usando Python como lenguaje de programación. Además se hizo uso de la librería *SQLAlchemy* para conectarse con la base de datos y de la librería *pandas* para leer los datos de los archivos CSV descargados y manipularlos para dejarlos en la forma que se espera en el almacén de datos.

3.4. Tratamiento de datos

Para el tratamiento de datos y entrenamiento de los modelos también se ha usado Python como lenguaje de programación por el ecosistema y comunidad que tiene para estas tareas. En concreto se ha hecho uso de las siguientes librerías:

- **Pandas** para la carga y manipulación de datos.
- **Matplotlib y Seaborn** para la visualización de datos.
- **Scikit-learn** para los modelos de aprendizaje automático.
- **StatsModels** para el modelo ARIMA.

3.5. Aplicación web

Como parte del proyecto se ha de desarrollar una aplicación web que muestre los últimos datos presentes en el almacén de datos y permita realizar predicciones usando el mejor modelo encontrado. Esto se ha decidido implementar en una sencilla API a la cual se conecta la página web. Para el desarrollo de esta API se ha hecho uso de Python y del framework Flask para no tener que implementar el modelo y tratamiento de los datos en un lenguaje distinto.

Como la parte pesada de la computación la realiza la API, la página web puede ser bastante sencilla. A parte de HTML, CSS y JavaScript se ha usado Parcel.js como dependencia de desarrollo para empaquetar la aplicación y Bulma como framework de CSS para darle un aspecto uniforme a la aplicación.

3.6. Otros

Docker es una herramienta que permite agrupar software y sus dependencias en un contenedor aislado del resto del sistema. De esta manera se puedan evitar problemas inesperados por ejecutar los programas en máquinas con configuraciones distintas.

Esto hace que Docker sea una herramienta bastante útil para el proyecto pues permite las características que se detallan a continuación:

- Un proyecto de investigación puede ser reproducido por otros investigadores con relativa facilidad.
- Una aplicación desarrollada localmente se puede desplegar en la nube con la tranquilidad de que el software se ejecutará en el mismo entorno que el local.

- Da la posibilidad de probar en local distintos sistemas como bases de datos con bastante facilidad. En caso de necesitar una base de datos distinta a la que se tiene en el sistema o una versión diferente sólo habría que ejecutar el contenedor correspondiente.

Debido a la complejidad y duración del proyecto se ha hecho uso de un software de control de versiones para gestionar los cambios en el mismo. En concreto se ha usado Git y Github como forja.

Como servicio de computación en la nube se ha elegido Heroku por su plan gratis para la base de datos: ofrecen una base de datos PostgreSQL gratis hasta alcanzar 10000 filas.

Capítulo 4

Desarrollo

Durante el primer capítulo se introdujo el problema que aborda este proyecto. En el segundo capítulo se presentó la teoría detrás de los distintos modelos usados sin entrar en detalles específicos del proyecto. En el tercer capítulo se empezó a entrar en detalles específicos del trabajo hablando de las distintas tecnologías usadas.

En el presente capítulo se abordará más detalladamente el avance del proyecto cronológicamente.

4.1. Extracción de datos

Tras investigar el problema así como las posibles tecnologías a usar para abordarlo se comenzó con la extracción de datos. En principio se contaba con dos fuentes de datos distintas:

- La Red de Control y Vigilancia de la Calidad del Aire de Canarias. Donde se cuenta con datos horarios de concentración de contaminantes y en algunas estaciones también de datos meteorológicos. En el apéndice A se puede ver más información sobre esta fuente de datos.
- La Agencia Estatal de Meteorologia (AEMET). Donde se pueden extraer datos meteorológicos diarios. En el apéndice B se puede ver más información sobre esta fuente de datos.

La idea inicial era usar ambas fuentes pero varias estaciones de la red de calidad del aire también recogen datos meteorológicos y los datos de AEMET solo están disponibles en períodos diarios y hay muchas menos estaciones que en la otra fuente de datos.

Una vez decidida la fuente de datos se procede a la extracción de datos históricos. Para ello hay una página web del gobierno de Canarias donde se pueden descargar los datos tras rellenar un formulario especificando qué se quiere obtener. Sin embargo, este formulario está pensado para extraer los datos manualmente y de mes en mes. Al menos

se pueden pedir los datos de varias estaciones al mismo tiempo pero esto hace que la consulta tarde mucho.

Además de lo tedioso que resulta esta tarea también es importante automatizarla para que se puedan introducir los datos al sistema de manera automática para que en todo momento esté actualizado.

Así se comenzó a desarrollar el software para la extracción de datos. Este es un programa que controla un navegador mediante código, rellena automáticamente un formulario para cada fecha y estación pedidas y descarga los datos en formato CSV.

Cabe mencionar que en la página web del gobierno hay un apartado para ver los datos más recientes pero se informa que estos pueden ser diferentes cuando formen parte de los datos históricos a descargar. Esto es porque internamente se validan los datos con los de otras estaciones para comprobar que no son producto de un error de la estación. Por ello no siempre se encuentran disponibles los datos más recientes para descargar y si extrajésemos los datos actuales además de los históricos podríamos tener en todo momento el sistema actualizado pero podrían haber inconsistencias.

4.2. Carga de datos

Una vez teniendo los datos descargados el siguiente paso fue introducirlos en el almacén de datos.

En primer lugar se llevaron a cabo las configuraciones necesarias para tener en local una base de datos usando Docker. A continuación se especificó el esquema de datos en Python usando el *Object Relational Mapping (ORM)* SQLAlchemy [1]. El modelo de datos se puede ver en representado un diagrama en la figura 4.1. En concreto consta de una tabla de hechos llamada `fact_measure` donde se recoge simplemente el valor medido y se relaciona con las tablas de dimensiones que aportan el resto de información:

- **Fecha**: en la tabla `dim_date` se almacena información acerca de la fecha en que se empezó a tomar la medida (día de la semana, día del mes, día del año, si era festivo o no, etc).
- **Hora**: en la tabla `dim_time` se guarda la hora en que se empezó a tomar la medida.
- **Duración**: en la tabla `dim_duration` está la duración de la medida. Todas las mediciones de la red de calidad del aire son medias horarias pero esta dimensión se incluyó por si se añadían datos de otras fuentes que difieran en esto.
- **Fuente**: en la tabla `dim_station` se encuentra información acerca de la fuente donde se extrajo la medición (de qué fuente de datos, nombre de la estación, propietario de la estación, localización de la estación, etc).

[1] Un *ORM* es una técnica para interactuar con una base de datos relacional usando un paradigma orientado a objetos [17]

- **Variable medida**: en la tabla `dim_measurement_type` hay información acerca de la variable medida (nombre, diminutivo, unidades).

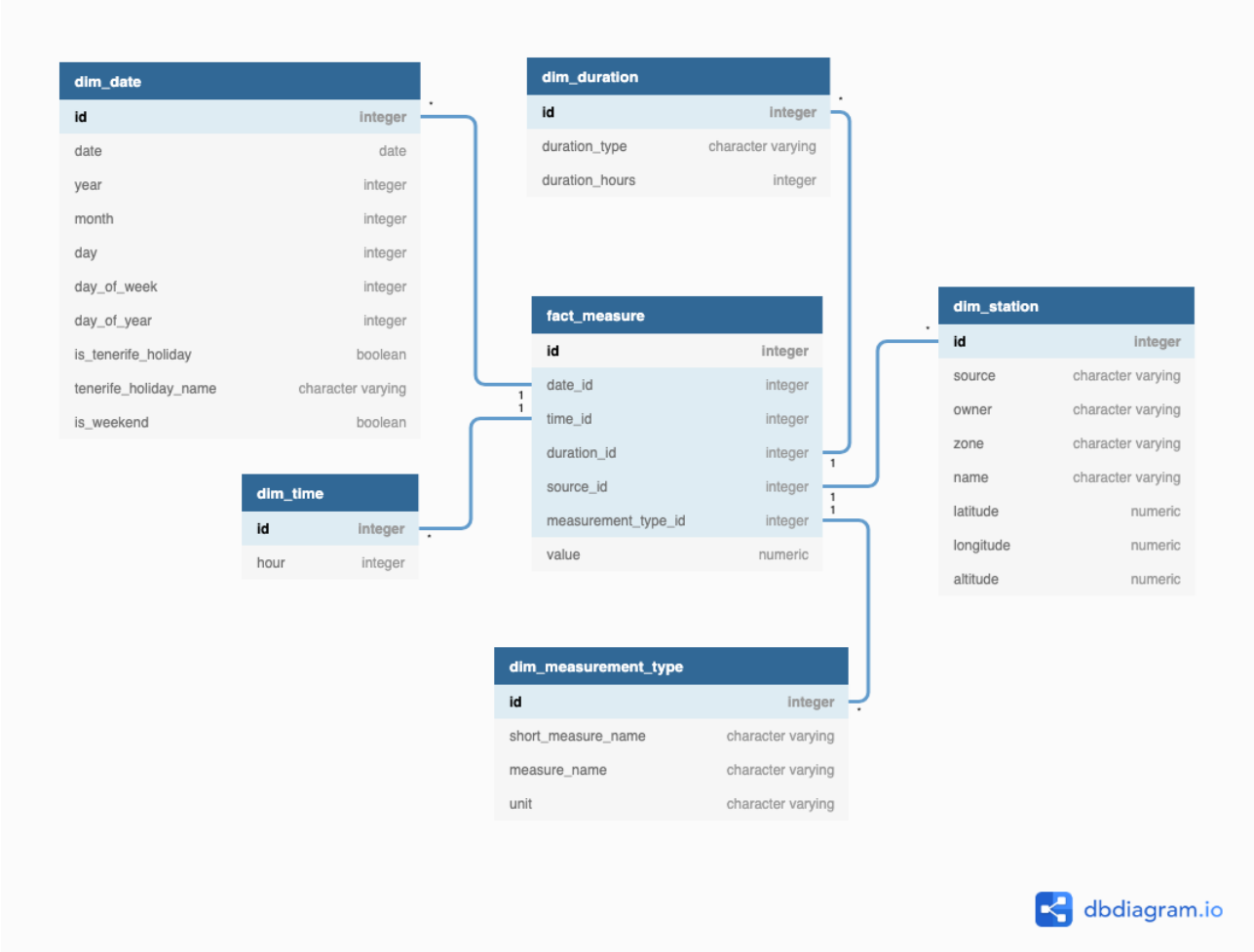


Figura 4.1: Modelo de datos del almacén de datos

Hecho esto se procedió a desarrollar el código para insertar los valores de las dimensiones y a leer los datos descargados, transformarlos para que se ajusten al almacén de datos e insertarlos en la tabla de hechos de manera automática.

No todas las estaciones pueden realizar las mismas medidas como se puede observar en la tabla A.2 del apéndice A. Por ello en esta parte se filtró el conjunto de estaciones a insertar de manera que estuvieran las que recogen medidas de varias de las variables más interesantes para el proyecto. En el caso de variables de contaminación estas son las principales consideradas por la OMS como se indicó en la introducción: el polvo en suspensión, el ozono, el dióxido de nitrógeno y el dióxido de azufre. En el caso de las variables meteorológicas: las relativas al viento, precipitaciones y temperatura pues son las que más afectan la presencia de contaminantes en la atmósfera. En concreto todas las variables en el sistema son las que se ven en la tabla 4.1. Y las estaciones son: Caletillas, Casa Cuna, Deposito De Tristan, García Escámez, La Buzanada, La Hidalga - Arafo, La Zamora, Parque La Granja, Piscina Municipal, San Isidro y Tome Cano.

Los nombres acortados que se ven en la tabla 4.1 son otro ejemplo de transformación de los datos aplicado en esta parte para hacerlos homogéneos en el sistema pues no

Abreviatura	Nombre completo	Unidades
PM2.5	Polvo en suspensión de 2.5 micrómetros o menos	$\mu g/m^3$
PM10	Polvo en suspensión de 10 micrómetros o menos	$\mu g/m^3$
O3	Ozono troposférico	$\mu g/m^3$
NO2	Dióxido de nitrógeno	$\mu g/m^3$
SO2	Dióxido de azufre	$\mu g/m^3$
WS	Velocidad del viento	m/s
WD	Dirección del viento	Grados
P	Presión atmosférica	mb
PP	Precipitaciones	l/m^2
RH	Humedad relativa	$\%$
SR	Radiación solar	W/m^2
T	Temperatura	Grados Celsius

Tabla 4.1: Variables medibles presentes en el almacén de datos

todas las estaciones nombran igual a las variables, algunos nombres difieren de los que se ven en la tabla A.1 del apéndice A en mayúsculas o minúsculas o en algún signo de puntuación.

4.3. Preprocesado de datos

Con el almacén de datos listo se continuó añadiendo la configuración necesaria de Docker para tener un servidor local de cuadernos Jupyter [2].

Tras esto se empezó con el preprocesado de los datos, es decir, preparar los datos para su posterior procesamiento.

Lo primero que se hizo fue agrupar los atributos por estación, fecha y hora de manera que se tenga una fila por estación y fecha y en columnas las distintas variables medidas. Esto es porque tal y como vienen los datos del almacén de datos haciendo una consulta sencilla se encuentran todas las medidas como filas distintas aún si son de la misma fecha y estación.

4.3.1. Datos no disponibles

A continuación se vio el porcentaje de datos no disponibles en 2019 y resultó que solamente la estación de Tome Cano tenía un porcentaje aceptable para el proyecto, al resto o les faltaba algún atributo o tenían un porcentaje de datos no disponibles bastante

[2] Los cuadernos Jupyter son aplicaciones web que sirven como documentos que contienen celdas de texto y código y son bastantes usados en el entorno de tratamiento de datos por la sencillez con la que se puede mostrar los resultados del proceso

alto como se puede ver en la tabla 4.2. Además este porcentaje es bastante mayor en caso de incluir datos anteriores a 2019, es por esta razón que se decidió usar solo los datos de Tome Cano de enero a julio de 2019.

Estación	PM2.5	PM10	O3	NO2	SO2	WS	WD	P	PP	RH	SR	T
CALETILLAS	0.69	1.96	0.44	19.01	0.52	100.00	100.00	100.00	100.00	100.00	100.00	100.00
CASA CUNA	22.79	21.83	21.03	27.68	21.58	20.75	20.75	20.75	100.00	20.75	100.00	20.75
DEPOSITO DE TRISTAN	23.70	23.70	23.73	24.12	23.32	22.99	22.99	23.01	100.00	22.99	100.00	22.99
GARCIA ESCAMEZ	20.36	20.36	20.92	20.53	20.61	20.36	20.36	20.36	100.00	20.36	100.00	20.36
LA BUZANADA	1.02	0.83	0.58	0.50	0.58	100.00	100.00	100.00	100.00	100.00	100.00	100.00
LA HIDALGA - ARAFO	22.16	22.30	21.72	33.50	25.08	29.97	21.03	20.92	20.92	21.25	21.03	21.03
LA ZAMORA	100.00	5.68	4.19	1.88	2.76	15.81	3.81	100.00	3.86	3.92	4.33	4.06
PARQUE LA GRANJA	26.30	26.30	21.91	21.80	39.71	20.89	20.89	20.89	100.00	20.89	100.00	20.89
PISCINA MUNICIPAL	100.00	20.23	20.58	22.02	21.41	0.33	0.47	0.33	0.33	0.33	0.33	0.33
SAN ISIDRO	3.70	3.12	100.00	2.15	2.10	100.00	100.00	100.00	100.00	100.00	100.00	100.00
TOME CANO	4.53	5.74	1.49	1.49	2.76	0.47	1.71	0.36	0.30	0.36	0.36	0.36

Tabla 4.2: Porcentaje de datos no disponibles contando de enero a mayo de 2019

Los datos no disponibles se rellenaron mediante interpolación lineal ya que debido a la fuerte relación temporal de los datos este método es más apropiado que usar otros como la media o mediana. En la figura 4.2 se puede ver un ejemplo de esta estrategia aplicada a la concentración PM2.5.

4.3.2. Creación de atributos

A los modelos de aprendizaje automático se les da como entrada los valores de todas las mediciones de las 24 horas anteriores y como valor a predecir la concentración media de PM2.5 durante 24 horas. Para ello se tuvo que procesar los datos de nuevo de manera que ahora las columnas fueran todas las $24 \times 12 = 288$ mediciones hechas cada 24 horas y se extrajo la variable objetivo calculando la concentración media de PM2.5 durante las 24 horas siguiente para cada observación (en el caso de predecir para 48 horas la variable objetivo es la media de las 24 horas siguientes a 24 horas de las observaciones).

Con esto se tiene que la entrada de los modelos es de $24 \times 12 = 288$ dimensiones. Además la correlación entre las mismas variables en el tiempo es muy alta como se puede observar en la figura 4.4, por eso se ha decidido usar PCA para reducir la dimensionalidad de los datos antes de pasárselos a los modelos para entrenar.

Pero antes de aplicar PCA se ha decidido transformar los datos ya que como se puede ver en la figura 4.3 la distribución de la variable objetivo es bastante asimétrica y aplicando una transformación adecuada como un logaritmo o una raíz cuadrada se puede lograr que los *outliers* se encuentren más cerca de la mayor parte de observaciones y tengan un efecto menor en los modelos.

Por supuesto, antes de realizar la transformación de los datos y PCA se separaron los datos en un conjunto de entrenamiento y otro de testeo. Esto se hizo de manera aleatoria dejando el 33 % de los datos para el conjunto de testeo usando el método `train_test_split` de la librería *scikit-learn*.

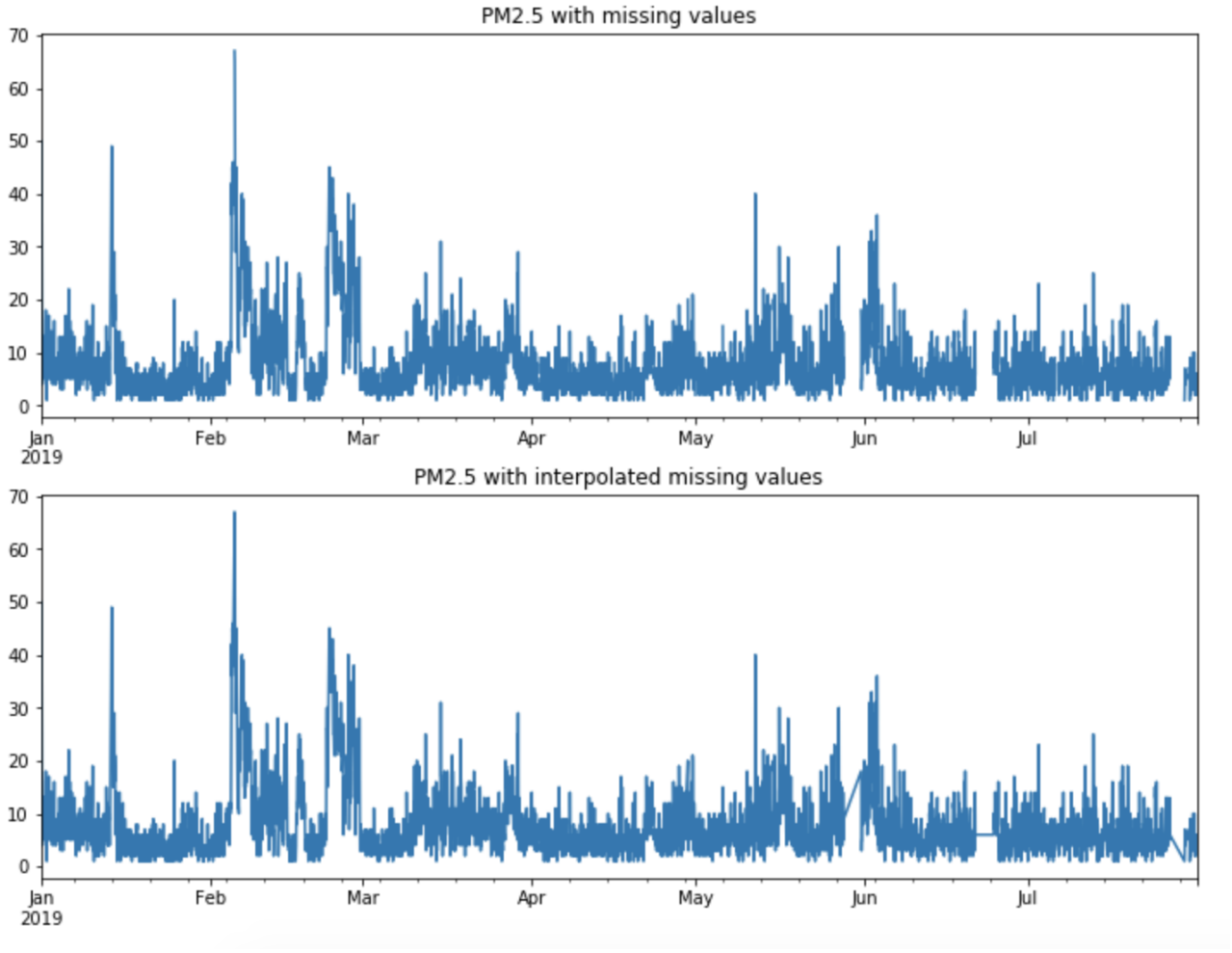


Figura 4.2: PM2.5 antes y después de interpolar

En la primera figura se puede ver una serie temporal de la concentración de PM2.5 sin manipular. En la figura de abajo se puede observar la misma serie temporal pero con los valores no disponibles rellenados mediante interpolación

```
variance: 26.359289305997173
count    5041.000000
mean        7.890159
std         5.134130
min         1.898438
25%         5.083333
50%         6.250000
75%         9.145833
max        44.833333
Name: PM2.5, dtype: float64
```

```
variance: 0.1835593993129883
count    5041.000000
mean        2.077356
std         0.428438
min         1.064172
25%         1.805553
50%         1.981001
75%         2.317063
max         3.825012
Name: PM2.5, dtype: float64
```

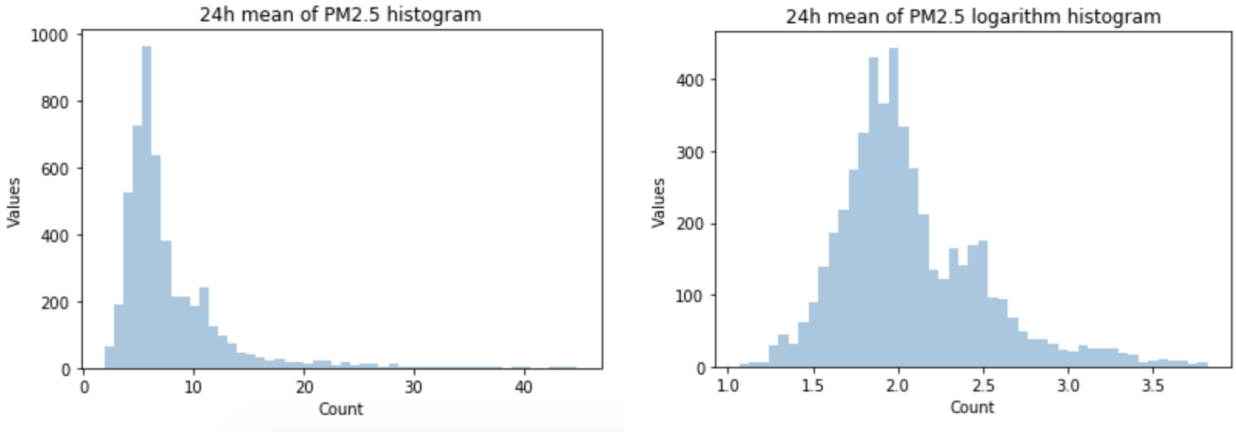


Figura 4.3: Histogramas de la media de 24 horas de PM2.5

En la figura de la izquierda se puede observar información y el histograma de la media de 24 horas de la concentración de PM2.5 (la variable objetivo de los modelos de predicción). Se ve que presenta una importante asimetría con una cola a la derecha. También cabe destacar que la varianza es de $26{,}36$ unidades, por lo que cualquier modelo que tenga un error cuadrático medio igual o mayor no sería de mucha utilidad. Por otra parte a la derecha se puede ver información y el histograma del logaritmo de la misma variable, se puede observar que la asimetría aunque sigue habiendo es bastante más débil.

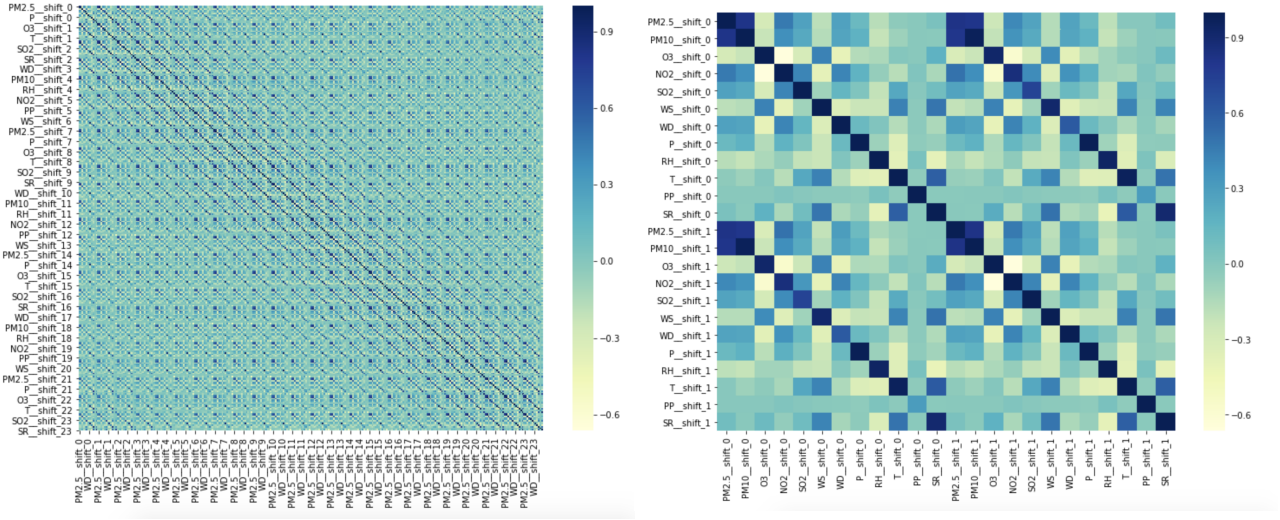


Figura 4.4: Mapa de calor de correlaciones entre los atributos

En ambas figuras se observa un mapa de calor donde el color indica el coeficiente de correlación de Pearson: medida de relación lineal de dos variables. Un valor de 0 indica que no hay relación lineal entre ambas mientras que un valor cercano a 1 o -1 indica una fuerte relación lineal. En este caso las variables que se están comparando son los atributos del conjunto de entrenamiento. En la figura de la izquierda se encuentran todos los 288 atributos mientras que en la derecha se encuentran los valores de las dos primeras horas para ver mejor el diagrama. Se puede ver una fuerte correlación positiva entre las mismas variables en distintos tiempos. Y entre el PM2.5 y PM10.

4.4. Aprendizaje automático

A partir del conjunto de entrenamiento se entrenaron los modelos de predicción vistos en el apartado 2.2.

En primer lugar el mejor número de dimensiones a las que reducir el conjunto de datos con PCA resultó ser 8 (con los cuales se retiene el 90 % de la varianza de los datos de 288 dimensiones). Este valor se obtuvo considerando PCA como parte de los modelos de predicción y tratando el número de dimensiones como un hiperparámetro. Se probaron diversos valores de la manera que se describirá a continuación y resultó que todos los modelos daban mejores resultados cuando se retiene el 90 % de la varianza frente al 99 % u 80 %.

Para buscar los mejores hiperparámetros para los modelos, se eligió una serie de valores posibles para cada hiperparámetro y se buscó la mejor combinación mediante una búsqueda exhaustiva usando la clase `GridSearchCV` de *scikit-learn* donde, para cada combinación de hiperparámetros, se entrena el modelo con validación cruzada y la combinación que da mejor resultado es la elegida como mejor. Los mejores hiperparámetros encontrados para cada modelo se pueden ver en la tabla 4.3. Los valores que no aparecen se debe a que se ha usado el valor por defecto de *scikit-learn* o a que el hiperparámetro no es aplicable al modelo.

Las métricas de validación cruzada para predecir la concentración media de PM2.5 de las 24 horas siguientes se pueden ver en la tabla 4.4. Y en la tabla 4.5 se tienen los

Hiperparámetro	Regresión lineal	Árbol de decisión	Random Forest	Gradient Boosting
max_depth	-	11	-	10
min_samples_split	-	20	-	5
n_estimators	-	-	500	500
learning_rate	-	-	-	0.05
n_iter_no_change	-	-	-	5

Tabla 4.3: Hiperparámetros de los modelos de aprendizaje automático

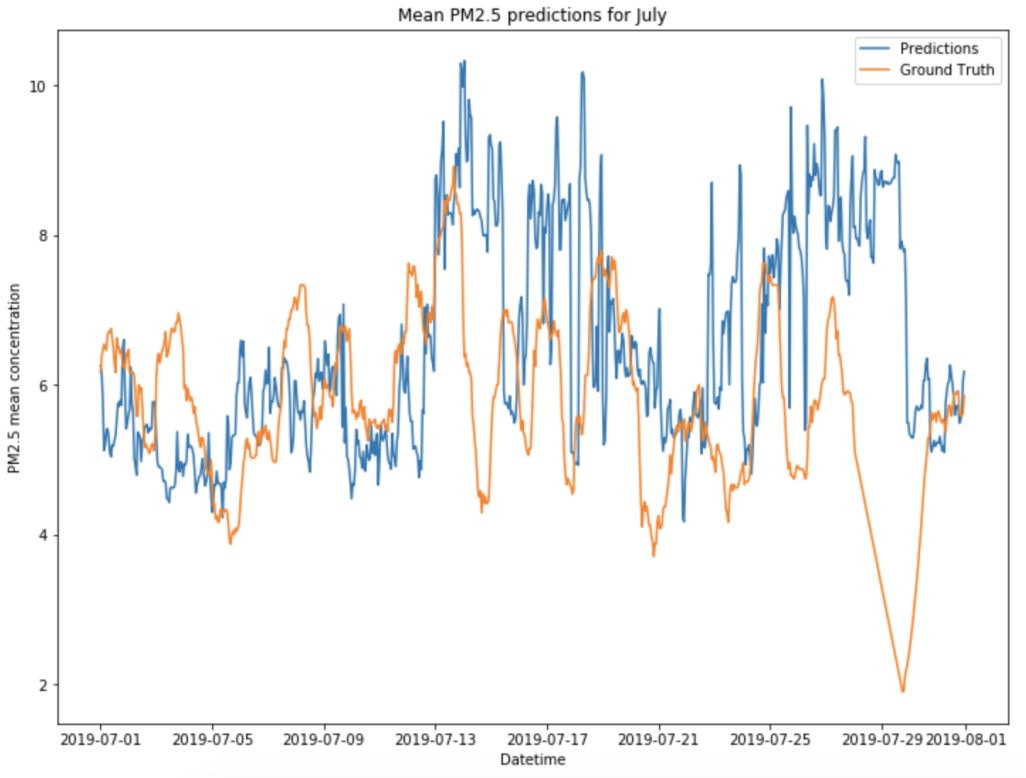
resultados del conjunto de testeo.

Por otro lado, en la tabla 4.6 se pueden ver las métricas obtenidas de validación cruzada para predecir la concentración media de PM2.5 de las 24 horas siguientes a las 24 horas. Y en la tabla 4.7 se tienen los resultados del conjunto de testeo. Los hiperparámetros de estos modelos son prácticamente los mismos, solamente el árbol de decisión tiene `min_samples_split` a 3 observaciones en lugar de las 20 del modelo anterior. Como se puede observar estos resultados son inferiores y hay una diferencia importante entre las métricas de validación cruzada y las el conjunto de testeo.

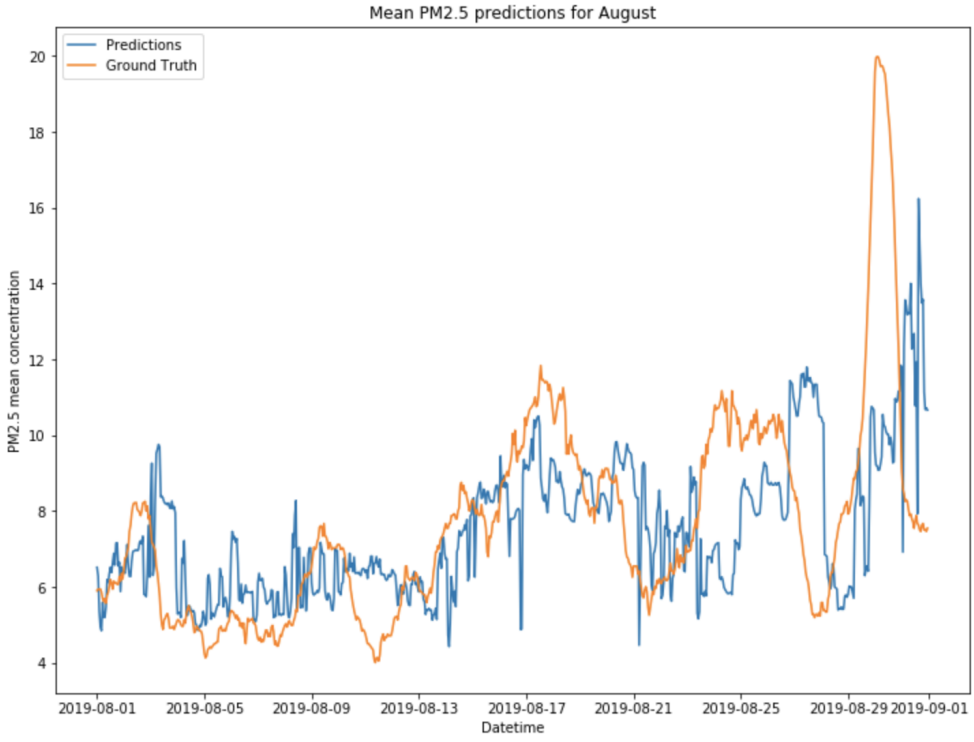
Se puede apreciar que en todos los casos el orden de peor a mejor modelo es el mismo: regresión lineal, árbol de decisión, *random forest* y *gradient boosting* es el mejor algoritmo en todos los casos. También se puede apreciar que la diferencia entre los resultados de validación cruzada y los de testeo es bastante pequeña en el caso de la predicción de las 24 horas siguientes lo cual nos puede indicar que son resultados fiables. Pero no podemos decir lo mismo de las predicciones de las 24 horas siguientes a las 24 horas pues la diferencia entre validación cruzada y testeo es importante.

Ahora, para poder decir si el resultado de *gradient boosting* para las 24 horas es bueno o no, hay que tener en cuenta que la variable objetivo tiene una distribución es muy asimétrica (figura 4.3) donde la mayoría de los valores se encuentran en un pequeño rango: el primer cuartil es aproximadamente 5 y el tercero 9. Por lo que normalmente se esperan valores que no varíen mucho, así que las predicciones de un modelo con un error cuadrático medio de aproximadamente 3.5 unidades podrían no ser buenas en el día a día.

Para visualizar los resultados del mejor modelo y evaluar más fielmente cómo se comportaría en un entorno real se ha entrenado con la misma configuración de hiperparámetros usando datos de enero a junio y evaluado usando datos de julio, los resultados se pueden comprobar en la gráfica 4.5a. De la misma manera, una vez se ha tenido acceso a los datos de agosto se procedió a realizar el mismo experimento y los resultados se pueden comprobar en la figura 4.5b. Cabe destacar que como se esperaba las concentraciones medias suelen variar en un rango pequeño excepto en unos casos puntuales a finales de agosto. Y en este pequeño rango las diferencias entre el valor real y el predicho por el modelo son notables.



(a) Predicciones con Gradient Boosting para julio



(b) Predicciones con Gradient Boosting para agosto

Figura 4.5: Predicciones con Gradient Boosting entrenado con datos de enero a junio para los meses de julio y agosto

Los valores de las gráficas son medias de la concentración de PM2.5 en intervalos de 24 horas. El descenso repentino que se puede apreciar a final de julio (figura a) se debe a que en esas fechas no habían datos disponibles y ese pico fue creado por la interpolación.

Métrica	Regresión lineal	Árbol de decisión	Random Forest	Gradient Boosting
R2	0.57503	0.79410	0.84025	0.87369
MAE	1.80133	1.09836	0.86516	0.58691
MSE	11.64299	5.57631	4.38584	3.49478

Tabla 4.4: Resultados de los modelos de aprendizaje automático en validación cruzada de 10 pliegues para predecir la media de las 24 horas siguientes

Métrica	Regresión lineal	Árbol de decisión	Random Forest	Gradient Boosting
R2	0.51560	0.76930	0.83369	0.84742
MAE	1.83898	1.06552	0.78654	0.59749
MSE	11.82795	5.63302	4.06082	3.72572

Tabla 4.5: Resultados de los modelos de aprendizaje automático en el conjunto de testeo para predecir la media de las 24 horas siguientes

Métrica	Regresión lineal	Árbol de decisión	Random Forest	Gradient Boosting
R2	0.19818	0.68190	0.73962	0.80155
MAE	2.50368	1.20163	1.05658	0.74493
MSE	20.15199	7.98895	6.54916	5.02128

Tabla 4.6: Resultados de los modelos de aprendizaje automático en validación cruzada de 10 pliegues para predecir la media de las 24 horas siguientes a las 24 horas

Métrica	Regresión lineal	Árbol de decisión	Random Forest	Gradient Boosting
R2	0.09392	0.48731	0.56826	0.70275
MAE	2.63135	1.50655	1.38260	1.06085
MSE	26.45434	14.96861	12.60515	8.67874

Tabla 4.7: Resultados de los modelos de aprendizaje automático en el conjunto de testeo para predecir la media de las 24 horas siguientes a las 24 horas

4.4.1. ARIMA

En los otros modelos se pudo dar como variables de entradas las medidas de las horas previas para predecir una media. Sin embargo, esto con ARIMA no es posible pues es un modelo que necesita de una serie temporal completa y sus predicciones son para los siguientes elementos de la serie.

Para este modelo se han usado las medias diarias de la concentración de PM2.5 sin transformar ya que la serie es estacionaria y se busca predecir la media diaria del día siguiente. Por ejemplo, con la serie del 1 de enero de 2019 al 30 de julio de 2019 se busca predecir la media del 31 de julio.

Para realizar validación cruzada no se puede simplemente particionar el conjunto de datos en k pliegues y usar uno para testear y el resto para entrenar ya que aquí existe una dependencia temporal entre los datos que se tiene que respetar. En su lugar se particiona

el conjunto en k pliegues y se usa el primero para entrenar y el segundo para validar. A continuación se usan los dos primeros para entrenar y el tercero para validar y así sucesivamente. Por supuesto, esto se ha hecho en el conjunto de entrenamiento que contiene las medias diarias de concentración de PM2.5 de enero a junio de 2019 y como conjunto testeo se ha dejado la media diaria de julio de 2019.

Usando esta estrategia los mejores hiperparámetros de ARIMA encontrados son $(0, 0, 0)$ con un error cuadrático medio en validación cruzada de 35.6 lo cuál significa que el modelo no ha sido capaz de distinguir ninguna característica de la serie temporal, simplemente lo trata como ruido. En la figura 4.6 se puede ver una gráfica de los resultados del modelo en el conjunto de testeo.

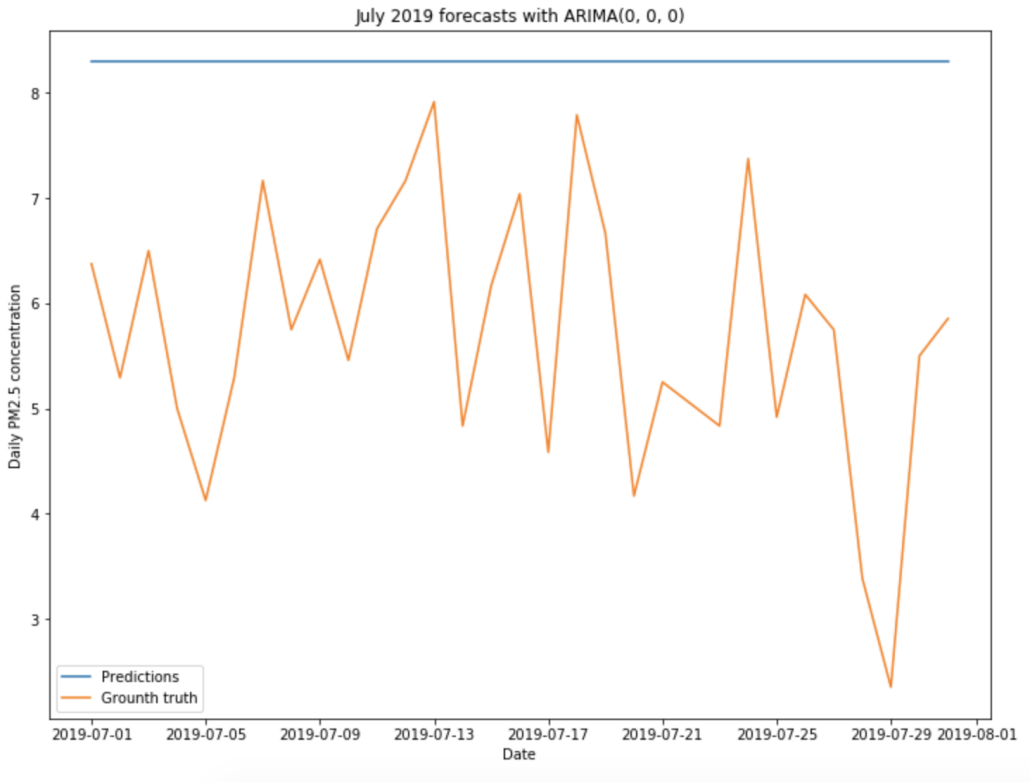


Figura 4.6: Predicciones con ARIMA(0, 0, 0) para julio

Una posible explicación de estos resultados tan pésimos con ARIMA puede ser que el uso de validación cruzada usando progresivamente más datos de entrenamiento nos perjudicara a la hora de encontrar los mejores hiperparámetros.

4.5. Aplicación web

4.5.1. API

Para la aplicación web se desarrolló una sencilla API usando el framework Flask. Esta aplicación se conecta al almacén de datos para poder desempeñar los tres servicios que ofrece:

- Información acerca de los datos, en concreto los distintos nombres de las variables que se pueden medir:

  URL: `/apf/api/v1.0/meta`

- Lista de los n últimos valores en el sistema para la variable pedida empezando a partir de un *offset* dado:

  URL: `/apf/api/v1.0/measures/[variable]/[n]/[offset]`

- Predicción para la concentración media de PM2.5 de las 24 horas empezando en el tiempo dado:

  URL: `/apf/api/v1.0/forecast/24/[timestamp]`

A la hora de desplegar esta API en Heroku también fue necesario hacer lo mismo con el almacén de datos. Sin embargo, en la base de datos gratis de Heroku solo se pueden almacenar hasta 10000 registros, por lo que no se pueden tener todos los datos ahí. Por esto en este almacén de datos solo se han puesto los datos más recientes en el momento del despliegue: los últimos días de julio. Y, para que esta aplicación se acerque a la realidad el modelo de Gradient Boosting usado para las predicciones en la nube ha sido entrenado usando solo los datos de enero a junio.

4.5.2. Web

La página web se divide en tres partes que se pueden ver en la figura 4.7. En la primera sección se encuentra una breve descripción de la página. En la sección de últimas medidas se interactúa con la API para mostrar las últimas medidas del sistema y en la parte de predicciones se interactúa también con la API para mostrar predicciones para la fecha y hora introducida por el usuario.

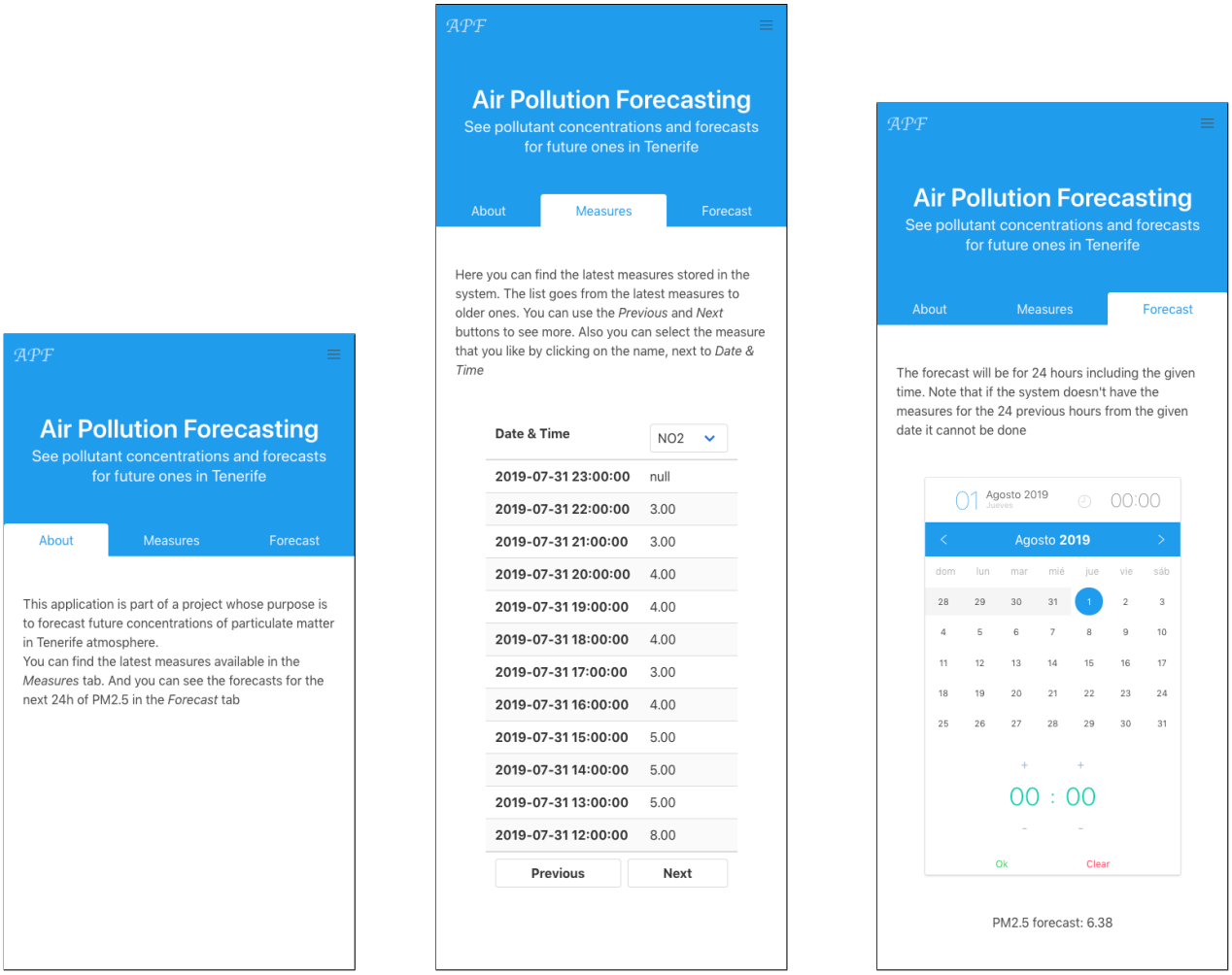


Figura 4.7: Capturas de pantalla de la página web

En la primera imagen empezando por la izquierda se puede ver la sección de la página web con un breve descripción de la página. En la segunda imagen se puede ver la sección de las últimas medidas donde se muestra una tabla en la que se puede seleccionar la variable que visualizar además de unos botones para ver valores anteriores o más recientes. Finalmente en la tercera imagen se puede ver la sección dedicada a predicciones donde se selecciona en un calendario el día y la hora a partir de la que realizar las predicciones y debajo aparece el resultado.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

Predecir la concentración de los principales contaminantes en la atmósfera no resulta una tarea sencilla por la gran cantidad de variables que afectan la presencia de los contaminantes en el aire: desde sucesos puntuales como incendios a hechos periódicos como el volumen de tráfico.

Sin embargo, teniendo en cuenta lo vulnerable que somos los seres vivos a agentes dañinos en el aire que respiramos, poder realizar este tipo de predicciones permitiría alertar a la población en caso de que se pueda llegar a límites peligrosos. De manera que las autoridades competentes puedan activar con tiempo políticas de reducción de la contaminación como restricciones de tráfico y advertir a los grupos de población más sensible, como las personas más jóvenes o mayores, para que utilicen algún tipo de protección como máscaras en el exterior.

El propósito principal de este proyecto ha sido abordar este problema en Tenerife usando modelos de aprendizaje automático entrenados con datos públicos de concentración de contaminantes y meteorológicos. El trabajo ha estado bastante limitado por la calidad de los datos y de su fuente, se empezó teniendo acceso a dos fuentes de datos con varias estaciones pero se descartó una por ser redundantes sus datos y de periodicidad diaria. Se tuvo que desarrollar una herramienta para descargar los datos de manera automática y otra para conseguir unos datos homogéneos en el almacén de datos. Además se tuvieron que descartar varias estaciones por no proveer las medidas deseadas. Y finalmente nos quedamos con los datos de 2019 de una sola estación debido al elevado porcentaje de datos no disponibles.

Sin embargo, a pesar de todo esto los resultados finales del mejor modelo usado, *Gradient Boosting* aunque no sean ideales resultan prometedores para continuar con el trabajo.

5.2. Líneas futuras

Hay muchas vías por las que se puede seguir el proyecto, a continuación se enumeran unas posibilidades:

- Se puede probar ARIMA con menos datos de entrenamiento para ver si en un período menor de tiempo se puede capturar alguna propiedad. O con muchos más datos si fuese posible.
- Si se tuviera acceso a más datos también se podrían probar modelos de *Deep Learning* que son el actual estado del arte para resolver distintos problemas con datos secuenciales como texto, audio o video.
- Se podría aplicar el conocimiento obtenido en este trabajo a datos de China pues es una zona en la que la predicción de la calidad del aire tiene un interés especial debido a las altas concentraciones de contaminantes. Y así se podrían comparar nuestros modelos con los de otros investigadores.
- Se podría cambiar la formulación del problema con el objetivo de intentar hacerlo más fácil para predecir. Por ejemplo tratar de predecir el cambio del nivel de concentración en lugar de la concentración media. Es decir, solamente predecir si el nivel de concentración va a aumentar o disminuir durante las próximas horas.

Capítulo 6

Summary and Conclusions

6.1. Conclusions

Forecasting the concentration of the main pollutants in the atmosphere is not an easy task due to the high number of variables that affect their presence in the air: from occasional events like fires to frequent episodes like traffic volume.

However, taking into account how vulnerable living beings are to harmful agents in the air that we breathe, being able to do this kind of predictions would allow us to warn the population in case of any potential danger. This way competent authorities could activate policies to reduce the pollution ahead of time like restricting the traffic and warn sensitive groups like children or elderly people to take some kind of protection like a mask if they go outside.

The main purpose of this project was to approach this problem in Tenerife using machine learning models trained with public data of pollutant concentrations and weather variables. This project was very limited to the quality of data and its source. At first we had two sources of data but one was discarded as the data was redundant and the granularity daily. We had to develop a software in order to download the data automatically and another software for having the data stored uniformly in the warehouse. Furthermore several stations had to be discarded because they didn't provided the expected measures. Finally we were left with data from 2019 and from one station due to the high percentage of missing data.

Nonetheless, the results of the best model trained, *Gradient Boosting*, were not ideal but they are promising to keep working on this project.

6.2. Future work

There are many ways that could be followed to continue the project. Some of them are listed below:

- ARIMA could be tested with fewer training data points to see if in a shorter span of time it can capture any useful feature of the time series. Or with a lot more data if possible
- With more data we could also try *Deep Learning* models because they are the current state-of-the art for solving different problems with sequential data like text, audio or video.
- We could try to apply what was learnt in this project to pollution data of China. This is because China is an area where air quality forecasting has a special interest due to the high pollutant concentrations. This way we could compare our models with the ones of other researchers.
- Our approach to try to solve the problem could be switched to a simpler one in order to try making it easier to predict. For example, we could try to predict the change direction of the concentration instead of the concentration mean. In other words, we could try to predict whether the current concentration will increase or decrease in the next few hours.

Capítulo 7

Presupuesto

En este capítulo se muestran los costes estimados del proyecto. El cual solo se compone de recursos humanos y hardware ya que el software usado es gratuito.

7.1. Costes de Hardware

Tipo	Descripción	Precio
Macbook Pro	Ordenador desde el que se realizó todo el proyecto	1349€

Tabla 7.1: Costes de hardware

7.2. Costes de Recursos Humanos

Horas de trabajo estimadas	Coste por hora	Total
540	10€	5400€

Tabla 7.2: Costes de recursos humanos

7.3. Costes de Totales

Hardware	Recursos humanos	Total
1349€	5400€	6749€

Tabla 7.3: Costes totales

Apéndice A

Información de la fuente de datos de la red de calidad del aire de Canarias

En Canarias, se dispone de varias estaciones automáticas que toman mediciones sobre concentraciones de contaminantes y factores atmosféricos como el material particularizado, ozono troposférico, dirección y velocidad del viento, etc. Hay estaciones que pertenecen a empresas privadas, en concreto a ENDESA y CEPSA, y estaciones que pertenecen al gobierno. Los datos de estas estaciones son tratados por el Centro de Evaluación y Gestión de la Calidad del Aire (CEGCA) [18] para que, en caso de que el nivel de calidad del aire incumpla la normativa, se activen los protocolos adecuados para disminuir emisiones contaminantes.

Los datos se pueden descargar en el sitio web del gobierno de canarias en intervalos horarios, diarios, mensuales o cada ocho horas. Las variables medibles en esta red se pueden ver en la tabla A.1. Y en la tabla A.2 se puede ver las estaciones de la red además de las variables que se pueden obtener de cada estación. Finalmente, en la tabla A.3 se puede ver la longitud y latitud en la que se encuentra cada estación.

Abreviatura	Nombre completo	Unidades
BC	Carbón Negro	µg/m³
BEN	Benceno	µg/m³
CO	Concentración de monóxido de carbono	mg/m³
DD	Direccion del viento	Grados
H2S	Concentración de ácido sulfhídrico	µg/m³
HR	Humedad relativa	%
LL	Precipitacion	l/m²
NO	Concentración de monóxido de nitrógeno	µg/m³
NO2	Concentracion de dióxido de nitrógeno	µg/m³
NOX	Concentracion de óxidos de nitrógeno	µg/m³
O3	Concentración de ozono	µg/m³
PM1	Particulas en suspensión <1um	µg/m³
PM10	Particulas en suspensión <10um	µg/m³
PM2.5	Particulas en suspensión <2.5um	µg/m³
PRB	Presion barometrica	mb
RS	Radiacion solar	W/m²
SO2	Concentración de dióxido de azufre	µg/m³
TMP	Temperatura media	ºC
TOL	Tolueno	µg/m³
TRS	*Total Reduced Sulphur*	µg/m³
VV	Velocidad del viento	m/s
XIL	Concentración de Xileno	µg/m³
m-p XIL	Concentración de m-p Xileno	µg/m³

Tabla A.1: Variables medibles en la red de control y vigilancia de la calidad del aire de Canarias

Nombre de estación	Dueño	Parámetros que recogen
Casa Cuna	CEPSA	BEN, CO, DD, HR, NO2, NOX, O3, PM10, PM2.5, PRB, SO2, TMP, TOL, TRS, VV, XIL
Depósito Tristán - Sta Cruz de TF	CEPSA	BEN, CO, DD, HR, NO2, NOX, O3, PM10, PM2.5, PRB, SO2, TMP, TOL, TRS, VV, XIL
García Escámez - Sta Cruz de TF	CEPSA	BEN, CO, DD, HR, NO2, NOX, O3, PM10, PM2.5, PRB, SO2, TMP, TOL, TRS, VV, XIL
Parque de la Granja - Sta Cruz de TF	CEPSA	BEN, CO, DD, HR, NO2, NOX, O3, PM10, PM2.5, PRB, SO2, TMP, TOL, TRS, VV, XIL
Tome Cano	MEDIO AMBIENTE	BEN, CO, DD, HR, LL, NO, NO2, NOX, O3, PM10, PM2.5, PRB, RS, SO2, TMP, TOL, VV, XIL
Vuelta Los Pájaros - Sta Cruz de Tenerife	CEPSA	BEN, CO, DD, HR, NO2, O3, PM10, PM2.5, PRB, SO2, TMP, TOL, VV, XIL
Piscina Municipal - Sta Cruz de TF	MEDIO AMBIENTE	BEN, CO, DD, HR, LL, NO, NO2, NOX, O3, PM10, PM2.5, PRB, RS, SO2, TMP, TOL, TRS, VV, XIL, m-p
Tena Artigas - Sta Cruz de Tenerife	MEDIO AMBIENTE	BC, DD, HR, LL, NO, NO2, NOX, O3, PM10, PM2.5, PRB, RS, SO2, TMP, VV
Tío Pino - Sta Cruz de Tenerife	MEDIO AMBIENTE	DD, HR, LL, NO, NO2, NOX, O3, PM10, PM2.5, PRB, RS, SO2, TMP, VV
Palmetum - Sta Cruz de Tenerife	MEDIO AMBIENTE	BEN, CO, DD, HR, NO, NO2, NOX, O3, PRB, RS, SH2, SO2, TMP, TOL, TRS, VV, XIL, m-p
Portátil Comandancia Marítima - Sta Cruz de Tenerife	MEDIO AMBIENTE	DD, HR, NO, NO2, PRB, SO2, TMP, VV
Portátil Hacienda - Sta Cruz de Tenerife	MEDIO AMBIENTE	NO, NO2, NOX, SO2
Portátil Bomberos - Sta Cruz de Tenerife	MEDIO AMBIENTE	DD, NO, NO2, NOX, SO2, VV
Portátil Comisaría - Sta Cruz de Tenerife	MEDIO AMBIENTE	DD, NO, NO2, NOX, SO2, VV
Los Gladiolos	DECONOCIDO	NO, NO2, NOX, O3, PM10, PM2.5, SO2
Mercatenerife	DECONOCIDO	NO2, PM10, SO2
Puerta litoral - Refinería	DECONOCIDO	TRS
Puerta principal - Refinería	DECONOCIDO	TRS
Refinería	DECONOCIDO	TRS
Refinería - Torre Meteorológica	DECONOCIDO	DD, HR, PRB, RS, TMP, VV
Vieja y clavijo	DECONOCIDO	SO2
Balsa de Zamora - Los Realejos	MEDIO AMBIENTE	CO, DD, HR, LL, NO, NO2, NOX, O3, PM10, PM2.5, PRB, RS, SO2, TMP, VV
La Hidalga - Arafo	MEDIO AMBIENTE	CO, DD, HR, LL, NO, NO2, NOX, O3, PM10, PM2.5, PRB, RS, SO2, TMP, VV
Barranco Hondo	ENDESA	DD, VV
Buzanada	ENDESA	CO, DD, HR, NO, NO2, NOX, O3, PM10, PM2.5, PRB, SO2, TMP, VV
Caletillas	ENDESA	DD, NO, NO2, NOX, O3, PM1, PM10, PM2.5, SO2, VV
Depósito La Guancha - Candelaria	ENDESA	CO, NO, NO2, NOX, O3, PM10, PM2.5, SO2
El Río	ENDESA	CO, NO, NO2, NOX, O3, PM10, PM2.5, SO2
Galletas	ENDESA	NO, NO2, NOX, PM1, PM10, PM2.5, SO2
Granadilla	ENDESA	CO, NO, NO2, NOX, PM1, PM10, PM2.5, SO2
Igueste	ENDESA	NO, NO2, NOX, O3, PM1, PM10, PM2.5, SO2
Médano	ENDESA	NO, NO2, NOX, O3, PM1, PM10, PM2.5, SO2
San Isidro	ENDESA	DD, HR, NO, NO2, NOX, PM1, PM10, PM2.5, SO2, TMP, VV
Tajao	ENDESA	CO, NO, NO2, O3, PM10, PM2.5, SO2
Torre Meteorológica de Candelaria	DESCONOCIDO	NO, NO2, PM1, PM10, PM2.5, SO2

Tabla A.2: Variables medibles por estación en la red de control y vigilancia de la calidad del aire de Canarias

Nombre de estación	Longitud	Latitud
Casa Cuna	-16.27769219311504	28.45103088037192
Tena Artigas - Sta Cruz de TF	-16.27685561780176	28.45537560999706
Vuelta Los Pájaros	-16.27697222222222	28.462
Depósito de Tristán - Santa Cruz de TF	-16.27877561502283	28.45815968532356
Unidad Móvil 3 - Piscina Municipal Sta Cruz de TF	-16.26340423894506	28.45791579546199
García Escámez - Sta Cruz de TF	-16.27184954317193	28.45664139423877
Parque La Granja - Sta Cruz de TF	-16.26487493264499	28.46300210751911
Tío Pino	-16.27012462756456	28.45925503570103
Palmetum	-16.25852644443512	28.452545726989584
Bomberos	-16.26097493125278	28.45828864775618
Comisaría	-16.25866186061195	28.45912395381277
Comandancia	-16.2456057616887	28.47720124167907
Hacienda	-16.24870801361327	28.46323011938556
Tome Cano	-16.26186591001509	28.4621688991889
La Zamora	-16.57072474718117	28.3831334121515
Barranco Hondo	-16.3581166775973	28.39342430257058
La Guancha - Candelaria	-16.36833427699988	28.38016131594777
Caletillas	-16.36193673089008	28.37672185381682
Igueste	-16.37197069941999	28.38054612249282
La Buzanada	-16.65275205038472	28.07264560673482
El Rio	-16.52369883375512	28.14507452830099
Las Galletas	-16.65582224157534	28.00778986209971
Granadilla	-16.57757403405965	28.11249291171923
El Médano	-16.53603007019015	28.04732410738179
San Isidro	-16.55983699771885	28.08003389965855
San Miguel de Tajao	-16.47161086862779	28.11139253073869
Arafo	-16.39991556602504	28.33734903726858

Tabla A.3: Localización de las estaciones

Apéndice B

Información de la fuente de datos de AEMET

La Agencia Estatal de Meteorología (AEMET) a través de AEMET OpenData provee al público de una *API REST* para descargar gratuitamente datos meteorológicos diarios y otros productos como gráficas elaborados por la agencia.

Las estaciones de esta fuente de datos en Tenerife se pueden ver en la tabla B.1. Mientras que las variables medibles se pueden ver en la tabla B.2.

Estación	Identificador	Latitud	Longitud	Altitud (m)
STA. CRUZ DE TENERIFE	C449C	28º 27’ 48” N	16º 15’ 19” W	35
AEROPUERTO NORTE	C447A	28º 28’ 39” N	16º 19’ 46” W	632
AEROPUERTO SUR	C429I	28º 02’ 51” N	16º 33’ 39” W	64
IZAÑA	C430E	28º 18’ 32” N	16º 29’ 58” W	2371
GÜÍMAR	C439J	28º 19’ 06” N	16º 22’ 56” W	115
PUERTO DE LA CRUZ	C459Z	28º 25’ 05” N	16º 32’ 53” W	25

Tabla B.1: Estaciones de Tenerife en la fuente de datos meteorológicos de AEMET

ID	Descripción	Tipo de dato	Unidad	Siempre presente en todas las estaciones
fecha	fecha del dia (AAAA-MM-DD)	string		true
indicativo	indicativo climatológico	string		true
nombre	nombre (ubicación) de la estación	string		true
provincia	provincia de la estación	string		true
altitud	altitud de la estación en m sobre el nivel del mar	float	m	true
tmed	Temperatura media diaria	float	grados celsius	false
prec	Precipitación diaria de 07 a 07	float	mm	false
tmin	Temperatura Mínima del día	float	ºC	false
horatmin	Hora y minuto de la temperatura mínima	string	UTC	false
tmax	Temperatura Máxima del día	float	ºC	false
horatmax	Hora y minuto de la temperatura máxima	string	UTC	false
dir	Dirección de la racha máxima	float	decenas de grado	false
velmedia	Velocidad media del viento	float	m/s	false
racha	Racha máxima del viento	float	m/s	false
horaracha	Hora y minuto de la racha máxima	float	UTC	false
sol	Insolación	float	horas	false
presmax	Presión máxima al nivel de referencia de la estación	float	hPa	false
horapresmax	Hora de la presión máxima (redondeada a la hora entera más próxima)	float	hora entera	false
presmin	Presión mínima al nivel de referencia de la estación	float	hPa	false
horapresmin	Hora de la presión mínima (redondeada a la hora entera más próxima)	float	hora entera	false

Tabla B.2: Información acerca de las variables medibles en la fuente de datos meteorológicos de AEMET

Bibliografía

[1] U. of California Santa Barbara. (2008). What is the chemical reaction that occurs in the body when carbon monoxide is inhaled?, dirección: http://scienceline.ucsb.edu/getkey.php?key=1856 (visitado 03-11-2018).

[2] O. M. de la Salud. (2018). Nueve de cada diez personas de todo el mundo respiran aire contaminado, dirección: http://www.who.int/es/news-room/detail/02-05-2018-9-out-of-10-people-worldwide-breathe-polluted-air-but-more-countries-are-taking-action (visitado 05-12-2018).

[3] ——, (2018). Calidad del aire y salud, dirección: https://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health (visitado 06-12-2018).

[4] U. S. E. P. Agency. (2019). What is Acid Rain?, dirección: https://www.epa.gov/acidrain/what-acid-rain (visitado 29-08-2019).

[5] (2018). KDD Cup of Fresh Air, dirección: https://biendata.com/competition/kdd_2018/ (visitado 14-03-2019).

[6] J. H. et al. (2018). Spatio-Temporal Feature Based Air Quality Prediction, dirección: https://www.dropbox.com/s/2glhcssotrccbqm/1st.ppt?dl=0 (visitado 14-03-2019).

[7] T. Getmax. (2018). KDD Cup 2018 Solution of Fresh Air, dirección: https://www.dropbox.com/s/ukviloiemjjdc7c/2nd.pptx?dl=0 (visitado 14-03-2019).

[8] T. Bui, V. Le y S. Cha, "A Deep Learning Approach for Air Pollution Forecasting in South Korea Using Encoder-Decoder Networks & LSTM", *CoRR*, vol. abs/1804.07891, 2018. arXiv: 1804.07891. dirección: http://arxiv.org/abs/1804.07891.

[9] C.-J. Huang y P.-H. Kuo, "A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart Cities", *Sensors*, vol. 18, pág. 2220, jul. de 2018. doi: 10.3390/s18072220.

[10] MariaDB. (2015). Exploring Early Database Models, dirección: https://mariadb.com/kb/en/library/exploring-early-database-models/ (visitado 17-02-2019).

[11] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks", *Commun. ACM*, vol. 13, págs. 377-387, ene. de 1970. doi: 10.1007/978-3-642-48354-7_4.

[12] B. Campbell. (2010). ACID and database transactions?, dirección: https://stackoverflow.com/a/3740307 (visitado 28-02-2019).

[13] K. Group. (). Star Schemas and OLAP Cubes, dirección: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/star-schema-olap-cube/ (visitado 03-09-2019).

[14] T. Hastie, R. Tibshirani y J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ép. Springer series in statistics. Springer, 2009, isbn: 9780387848846. dirección: https://books.google.es/books?id=eBSgoAEACAAJ.

[15] scikit-learn. (). Tree algorithms: ID3, C4.5, C5.0 and CART, dirección: https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart (visitado 04-09-2019).

[16] R. Hyndman y G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2014, isbn: 9780987507105. dirección: https://books.google.es/books?id=gDuRBAAAQBAJ.

[17] e-satis. (2009). What is an ORM, how does it work, and how should I use one?, dirección: https://stackoverflow.com/a/1279678 (visitado 07-09-2019).

[18] G. de Canarias. (). Centro de evaluación y gestión de la calidad del aire (CEGCA), dirección: http://www.gobiernodecanarias.org/cptss/sostenibilidad/temas/calidad/centro-evaluacion-gestion-calidad-aire/ (visitado 10-09-2019).