



Business School

MÁSTER EN DATA SCIENCE Y BUSINESS ANALYTICS
MODALIDAD ONLINE

Machine Learning para Series
Temporales. Estudio de diferentes
modelos para la predicción de los
niveles del río Arno.

TFM elaborado por: Manuel Diz Castro
Tutor/a de TFM: Abel Ángel Soriano Vázquez

- Vigo, 7 de Febrero de 2023 -

Contenido

Resumen	4
1. Introducción.....	5
2. Contextualización	7
2.1 Contexto funcional	7
2.1.1 Actividad de Acea Group. Descripción del desafío planteado	7
2.1.2 Predicción de la Hidrometría de un río. Variables de interés	8
2.1.3 Datos disponibles para realizar la predicción	9
2.2 Contexto tecnológico	10
2.2.1 ¿Qué es una serie temporal?	10
2.2.2 Cómo manipular series temporales en Python	11
2.2.3 Predicción de series temporales. Modelos ARIMA	13
2.2.4 Predicción de series temporales. Machine Learning	15
2.2.5 Redes Neuronales Recurrentes LSTM	22
2.3 Entorno para la resolución del problema.....	25
3. Preprocesado de datos y Análisis exploratorio (primera parte).....	26
3.1 Tratamiento de valores nulos.....	27
3.2 Análisis exploratorio: Variable target (Hidrometría)	29
3.3 Análisis exploratorio: Precipitaciones	34
3.4 Análisis exploratorio: Temperatura	38
4. Análisis exploratorio (segunda parte).....	40
4.1 Influencia de las Precipitaciones - Modelo de regresión lineal	42
4.2 Búsqueda de variables predictoras adicionales - Modelo de Random Forest	46
5. Modelado	51
5.1 Consideraciones previas	51
5.1.1 Horizonte temporal de la predicción	51
5.1.2 Modelo utilizado.....	52

5.2 Modelado con Redes Neuronales Recurrentes LSTM	52
5.2.1 Preprocesado de datos	52
5.2.2 Clases y funciones	54
5.2.3 Entrenamiento del modelo.....	54
6. Obtención de predicciones a futuro - diferentes horizontes temporales.....	59
6.1 Predicción al día siguiente.....	59
6.2 Predicción para dentro de dos semanas	60
7. Conclusiones.....	66
7.1 Recapitulación	66
7.2 Sugerencias para ampliar este estudio	66
Referencias	70
Anexo 1: Recabar datos de Temperatura desde una fuente de datos externa.....	72
Anexo 2: Análisis de valores anómalos en la correlación Hidrometría-Precipitaciones	74
Anexo 3: Obtención de la variable “Momento del año”	76

Resumen

Este trabajo trata sobre la resolución de un desafío lanzado en la Web por una compañía italiana gestora de aguas: elaborar un modelo que permita predecir cuáles serán los niveles del río Arno (región de la Toscana, Italia) en el futuro. La utilidad práctica de estas predicciones es la correcta gestión de los recursos hídricos de la compañía, para poder garantizar en todo momento el suministro a sus clientes.

Este es un tipo de problema que se puede catalogar como una serie temporal. Las series temporales se pueden abordar con técnicas de Machine Learning, y por ello nos serán de gran utilidad los conocimientos que hemos adquirido a lo largo del máster. Eso sí, son un tipo de problema que tienen una serie de singularidades, así que tendremos que adquirir ciertos conocimientos extra antes de intentar resolver el problema.

Una vez conseguido esto, se recogen los datos necesarios para el estudio, se ponen en contexto, y se analizan. Veremos que, para predecir los niveles del río, las variables más importantes son: las precipitaciones ocurridas, la temperatura, y los propios niveles del río en el pasado. Las sutilezas acerca de la forma en la que influyen valores pasados en las medidas del presente (y del futuro), es uno de los grandes desafíos de las series temporales en general. Además, veremos que esta serie temporal se escapa muchas veces de los supuestos que se suelen tomar para modelizarlas, lo cual hará aún más interesante este estudio.

Finalmente, se encuentra un modelo de Redes Neuronales Recurrentes LSTM que devuelve unas predicciones con una precisión sorprendentemente buena. Se utiliza dicho modelo para realizar predicciones a diferentes horizontes temporales, y se proponen una serie de líneas maestras en el caso de que se quisiese profundizar aún más en el estudio de este problema.

1. Introducción

Este trabajo trata sobre la resolución de un desafío lanzado en la Web por una compañía italiana gestora de aguas: elaborar un modelo que permita predecir cuáles serán los niveles del río Arno (región de la Toscana, Italia) en el futuro.

Siendo éste un trabajo dentro del ámbito académico, considero que no sólo es importante exponer los pasos que he tomado y los resultados que he obtenido. Además, se va a hacer hincapié en que el lector comprenda el proceso de aprendizaje de conceptos y herramientas que fue necesario para poder acometer este desafío. También, es interesante que el lector pueda seguir con claridad el proceso de razonamiento. Es decir, cómo se ha roto el problema a resolver en diferentes subproblemas, y cómo se han ido poniendo en práctica los conocimientos y herramientas aprendidos para ir resolviendo cada uno de ellos, hasta cumplir el objetivo final que es obtener predicciones fiables de los niveles del río Arno.

A continuación, detallamos la estructura de capítulos que se ha elegido, y qué se podrá ver en cada uno de ellos.

En un primer capítulo de Contextualización, se va a acercar al lector al problema que tratamos de resolver. Se va a dividir este capítulo en dos bloques.

- Contexto funcional: En primer lugar, se explicará la naturaleza del desafío lanzado por la compañía gestora de aguas. Veremos para qué puede ser útil, a nivel de negocio, la obtención de predicciones de los niveles de un río. Profundizaremos también acerca de las diferentes variables que nos pueden ayudar en este tipo de predicciones, y cómo se relacionan entre ellas.
- Contexto tecnológico: Una vez tenemos claro qué queremos predecir y de qué tipo de datos disponemos, toca responderse cómo se abordará la transformación de datos en predicciones. Este problema es una Serie Temporal, y como tal tiene una serie de particularidades. Veremos cuáles son los conocimientos sobre series temporales que he adquirido específicamente para la realización de este trabajo, y cómo se relacionan con otros conocimientos adquiridos en el máster como procesamiento y visualización de datos (Pandas, Matplotlib) y Machine Learning, para poder resolver este desafío de forma exitosa.

En los siguientes capítulos, se explica de principio a fin cómo se resolvió el problema: desde la carga de datos, hasta la obtención de predicciones a futuro. Estos capítulos coinciden con las secciones homónimas del notebook que se utilizó para resolver el problema. Así, si se quiere consultar con más detalle el código utilizado, será fácil para el lector encontrar la correspondencia entre este documento y el notebook. Los capítulos en cuestión son:

- Preprocesado de datos y Análisis exploratorio (Primera parte)
- Análisis exploratorio (Segunda parte)
- Modelado
- Obtención de predicciones

Y para finalizar, se incluye un capítulo de Conclusiones, donde se repasa cómo se abordó el problema, qué resultados se obtuvieron, y qué ideas se sugieren si se quisiese continuar profundizando en el estudio de este tipo de problema.

2. Contextualización

2.1 Contexto funcional

Este trabajo trata sobre la resolución de un problema planteado por una compañía italiana gestora de aguas, [Acea Group](#), en el marco de una competición subida a la página web de Kaggle:

<https://www.kaggle.com/competitions/acea-water-prediction/>

2.1.1 Actividad de Acea Group. Descripción del desafío planteado

Acea es uno de los operadores líder en el sector Utilities en Italia, gestionando el suministro de agua y electricidad. Se trata del primer operador italiano en el suministro de agua, realizándolo para 9 millones de personas en las regiones italianas de Lazio, Toscana, Umbria, Molise y Campania.

El activo principal de esta compañía para poder realizar su actividad son diferentes cuerpos de agua presentes en las regiones cercanas a donde opera. Éstos (acuíferos, manantiales, ríos, lagos) son los reservóreos de donde puede obtener el agua necesaria para realizar el suministro a sus clientes.

Para la compañía, es importante realizar una predicción de los niveles de agua en cada uno de estos cuerpos de agua, a fin de poder entender cuál es la disponibilidad de agua que tiene en cada uno de sus recursos, y así poder asegurar un suministro continuo a sus clientes.

El desafío que plantea Acea Group es construir, para cada uno de estos cuatro tipos diferentes de cuerpos de agua (acuíferos, manantiales, ríos, lagos), un modelo matemático que se pueda utilizar para predecir la cantidad de agua que se dispondrá en ellos.

Cada uno de estos tipos de cuerpos de agua son de naturaleza diferente, y es por eso que tanto los datos como el modelado deben de ser tratados de forma diferente. En este trabajo, se aborda la modelización de uno de estos cuatro tipos de cuerpos de agua: el río. Para cuantificar la cantidad de agua que se dispone en un río, utilizaremos una métrica denominada Hidrometría, que no es más que el nivel que presenta el río en un momento determinado.

2.1.2 Predicción de la Hidrometría de un río. Variables de interés

En el marco de este desafío, Acea Group está interesada en realizar predicciones de la Hidrometría para el río Arno.

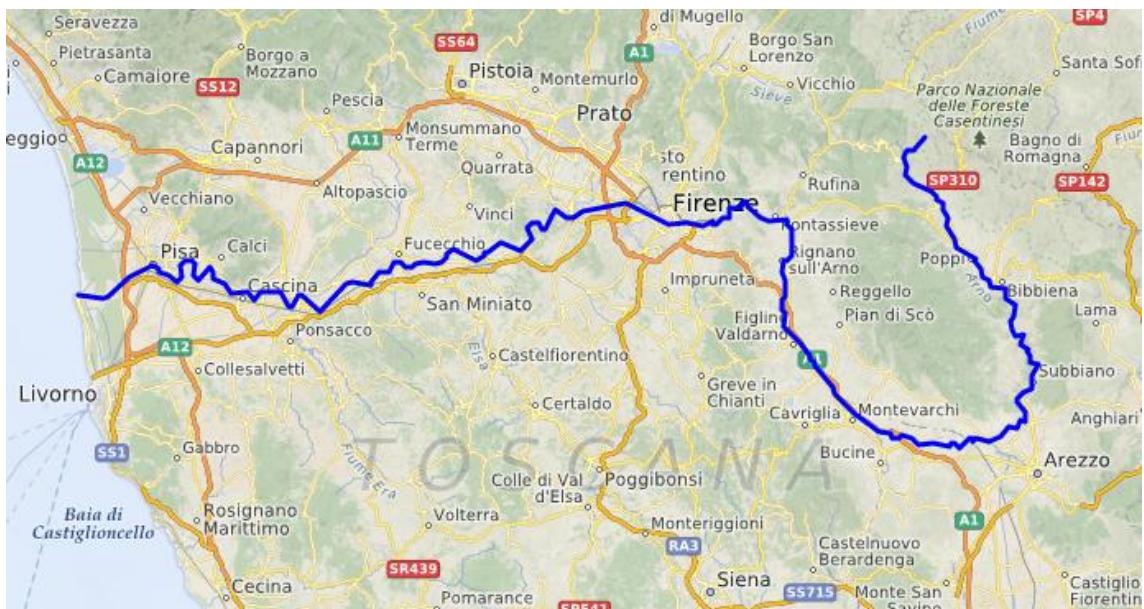


Imagen extraída de: [https://en.wikipedia.org/wiki/File:Arno_\(fleuve\).png](https://en.wikipedia.org/wiki/File:Arno_(fleuve).png)

El río Arno es el segundo río más largo en la Península Itálica, siendo el principal río de la región de la Toscana, y la principal fuente de suministro de agua para el área metropolitana de Florencia-Prato-Pistoia.

Para la predicción de los niveles del río en el futuro, las variables de interés serán:

- Las Precipitaciones caídas en regiones cercanas al río
- La Temperatura en regiones cercanas al río
- La propia serie histórica de la Hidrometría, en tanto que lo que haya ocurrido en el pasado nos puede ser de interés para predecir lo que ocurrirá en el futuro

En cuanto a por qué son Precipitaciones y Temperatura variables influyentes en la Hidrometría de un río, parece que se trata de una pregunta que se responde de forma bastante intuitiva.

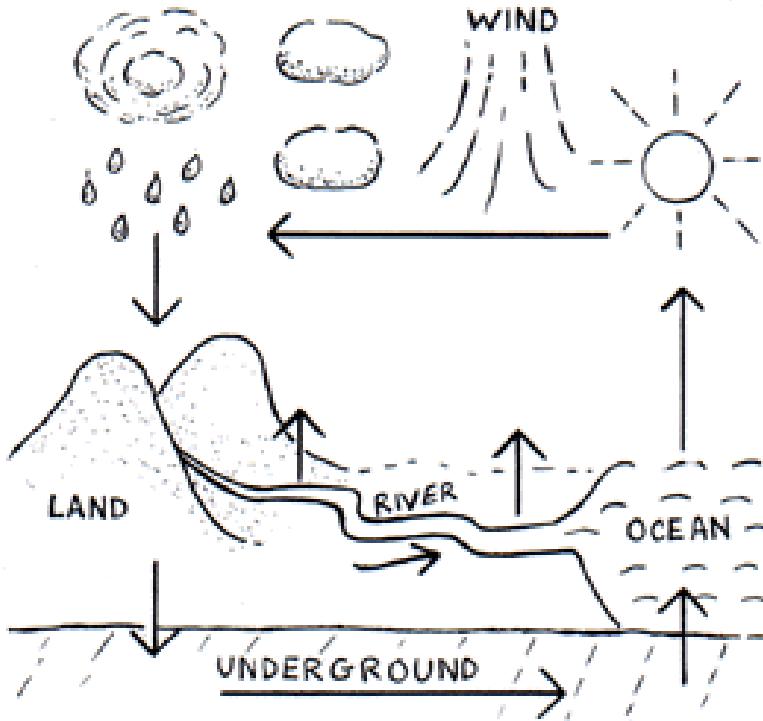


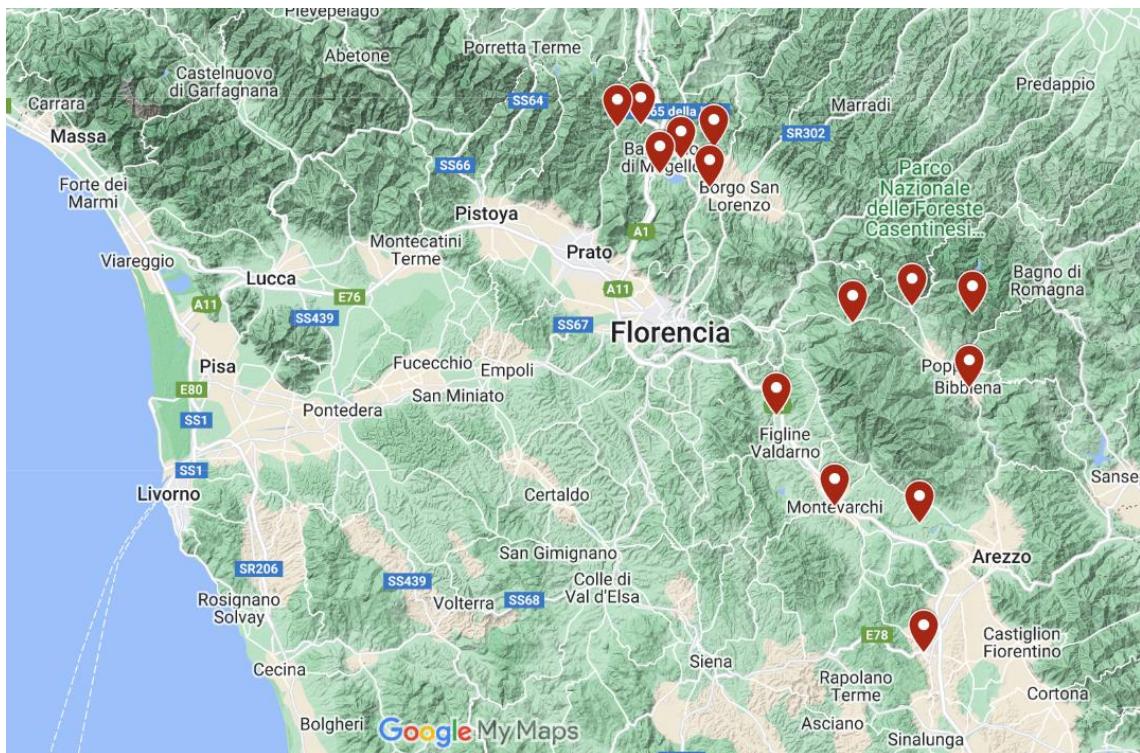
Imagen extraída de: <https://ypete.org.uk/factsheets/rivers/rivers-and-the-water-cycle>

El agua procedente de las precipitaciones en regiones cercanas a la cuenca del río se incorporará a la misma y harán que los niveles crezcan. Cuanta mayor sea la temperatura, cabe esperar que mayor sea la evaporación de agua en la cuenca del río, y por tanto los niveles del río decrezcan.

2.1.3 Datos disponibles para realizar la predicción

En el desafío subido a Kaggle, Acea Group proporciona un fichero .csv con los datos disponibles para realizar la predicción. Se recogen datos desde el 1 de Enero de 1998, hasta el 30 de Junio de 2020. Las variables proporcionadas son las siguientes:

- Hidrometría (variable objetivo): expresada en metros (m.), medidos en la estación hidrométrica de Nave di Rosano (Rosano es una localidad situada a unos pocos kilómetros de Florencia, río arriba).
- Temperatura: expresada en grados centígrados ($^{\circ}\text{C}$), medida en la ciudad de Florencia.
- Precipitaciones: expresadas en mm/m^2 , cantidades recogidas en diferentes puntos de la región que estamos estudiando, a saber: Le Croci, Cavallina, S. Agata, Mangona, S. Piero, Vernio, Stia, Consuma, Incisa, Montevarchi, S. Savino, Laterina, Bibbiena, Camaldoli.



Fuente: elaboración propia.

NOTA: en los datos proporcionados por Acea Group, no se indican las localizaciones exactas de cada uno de estos lugares, simplemente se indica el nombre de los mismos. Se ha elaborado este mapa a partir de búsquedas en Google Maps de los nombres de estas localizaciones.

2.2 Contexto tecnológico

El problema descrito en el capítulo anterior es una serie temporal. En este capítulo, veremos en primer lugar en qué consisten este tipo de problemas. A continuación, repasaremos algunos de los conceptos y herramientas que necesitamos conocer para intentar resolver un problema de series temporales.

2.2.1 ¿Qué es una serie temporal?

Una serie temporal es una secuencia de puntos de datos recogida en intervalos de tiempo, lo cual nos permite rastrear los cambios a través del tiempo (Kulkarni, A., Booz, R., Toth, A., 2022). Algunos ejemplos clásicos de series temporales son la evolución de la cantidad de ventas (en unidades monetarias) de una compañía, o la evolución de la cantidad de pasajeros que toman el avión en un determinado país.

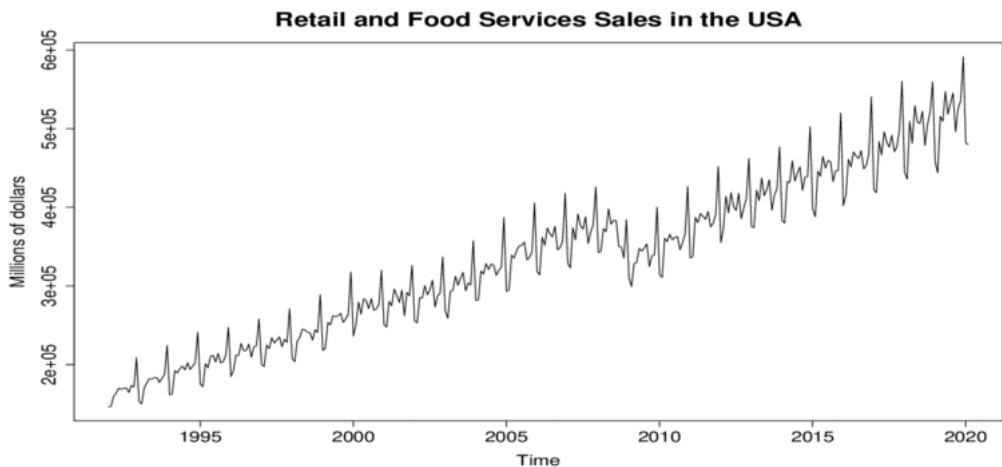


Imagen extraída de: <https://www.researchgate.net/>

U.S. passenger air travel

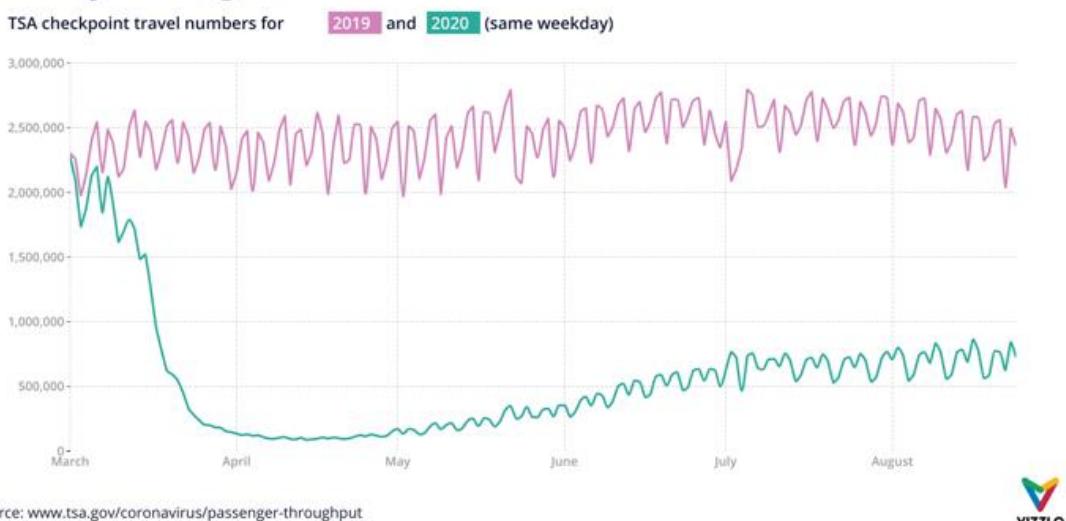


Imagen extraída de: <https://vizzlo.com/>

Al igual que en los ejemplos que acabamos de mostrar, el problema que nos ocupa también es una secuencia de puntos de datos recogida en intervalos de tiempo. En nuestro caso, se han recogido, de forma diaria, los datos de Hidrometría del río Arno, además de otros datos que pueden ayudar a la predicción como son las Precipitaciones recogidas y la Temperatura. Nuestro objetivo será captar patrones en la evolución de estos datos para así conseguir nuestro objetivo, que es predecir cuál será la Hidrometría en el futuro.

2.2.2 Cómo manipular series temporales en Python

Antes de poder aventurarse a realizar predicciones sobre una serie temporal, es necesario investigar los datos que se disponen, tratar de captar las relaciones que hay entre las

diferentes variables de estudio, tratar de captar patrones que se puedan estar repitiendo en el tiempo. A este efecto, hay dos librerías para Python que nos serán de gran utilidad:

- Pandas para el procesamiento de datos
- Matplotlib para la visualización

2.2.2.1 Procesamiento de series temporales con Pandas

A los conocimientos que ya poseemos sobre esta librería tras la realización de los módulos del máster, resulta interesante añadir algunos conocimientos específicos sobre procesado de series temporales. La librería incluye una serie de clases y métodos que facilitan enormemente el trabajo con series temporales (Jansen, S., 2023).

- Clase datetime: sirve para poder trabajar con datos en formato fecha y tiempo. Necesario para poder aplicar alguno de los métodos que veremos a continuación. Además, se puede pasar un objeto de tipo string en una función que espera un objeto de tipo datetime, y la conversión se realiza de forma automática. Por ejemplo, si se quiere filtrar una serie temporal para observar sólo los datos de Abril de 2016, basta con pasar como argumento la sencilla cadena de texto ‘2016-04’.
- Upsampling/downsampling: Incrementar/disminuir la frecuencia de los datos.
 - En determinadas situaciones, puede ser interesante disminuir la frecuencia de los datos. Por ejemplo, si tenemos datos diarios y queremos ver datos agregados para un periodo más largo (menos frecuencia), como puede ser el volumen de ventas anual. En este caso, podemos usar el método .agg(). Si lo utilizamos sobre un objeto de tipo datetime, podemos hacer cómodamente agrupados por intervalos temporales como meses, años, etc.
 - A veces, puede surgir la necesidad contraria, aumentar la frecuencia de los datos, por ejemplo, tener datos de tipo mensual y bajarlos a un grano más fino como son los datos diarios. Para ello, podemos utilizar el método .resample() en conjunción con un método de interpolación, .ffill() por ejemplo.
- “Mover” una serie en el tiempo. Con el método .shift(), podemos desplazar una serie temporal hacia delante o hacia atrás en el tiempo. Muy útil para poder comparar valores presentes de una serie temporal, frente a valores de esa misma serie en el pasado.

- Rolling windows: Método `.rolling()`. Imaginemos que tenemos una serie temporal con frecuencia diaria, sobre el volumen de ventas de una determinada compañía. Con las rolling windows podemos conocer agregados interesantes como, para cada día de la serie temporal, saber cuánto se ha vendido en los tres meses anteriores.

2.2.2.2 Visualización con Matplotlib

Nuevamente, estas librerías de Python están perfectamente preparadas para facilitarnos el trabajo con series temporales. Podemos aprovecharnos de la gran integración que existe entre Matplotlib y Pandas para crear gráficos con muy pocas líneas de código. Por ejemplo, con una sentencia tan sencilla como

```
df.loc['2013-01':'2013-06', ['Sales']].plot()
```

, podemos visualizar un diagrama de línea con la evolución diaria de las ventas de una compañía en el primer semestre del año 2013.

2.2.3 Predicción de series temporales. Modelos ARIMA

Este tipo de modelos son muy populares para resolver problemas de series temporales. Para este trabajo en concreto, se ha considerado que no es apropiado intentar modelizarlo utilizando ARIMA. De todas formas, es interesante conocer (a muy alto nivel) los fundamentos de este tipo modelos, lo cual nos ayudará a entender por qué para nuestro caso concreto no son la mejor opción.

ARIMA es el acrónimo de AutoRegressive, Integrated, Moving Average. Se trata de la conjunción entre modelos de tipo AR y modelos de tipo MA (Fulton, J., 2023).

Para los modelos de tipo AR, nos basamos en la suposición de que el valor de una variable en un momento determinado, depende de los valores de esa misma variable en el pasado:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \epsilon_t$$

La ecuación anterior sería la de un modelo AR de orden 2. Imaginemos de nuevo que tenemos una serie temporal de frecuencia mensual con el volumen de ventas de una compañía. Con este modelo de orden 2, asumimos que podemos predecir las ventas en un mes cualquiera (y_t), a partir de las ventas en el mes anterior (y_{t-1}) multiplicadas por un determinado coeficiente (a_1 , marca el grado de influencia entre las ventas de este mes y

el anterior), más las ventas de hace dos meses (y_{t-2}) multiplicadas por otro coeficiente a_2 , más un término de error ε_t .

Para los modelos de tipo MA, en el ejemplo tenemos un modelo MA de orden 1

$$y_t = m_1 \varepsilon_{t-1} + \varepsilon_t$$

nos basamos en la suposición de que el volumen de ventas en un mes cualquiera (y_t) se puede predecir a partir del error cometido prediciendo el volumen de ventas del mes anterior (ε_{t-1}) multiplicado por un determinado coeficiente (m_1), más un término de error ε_t .

Además, estos modelos pueden ser completados con un término que recoja las variaciones estacionales (*seasonality*), y la influencia de otras variables que no son la propia variable que queremos predecir (variables exógenas). Así, se tendría un modelo de tipo SARIMAX (Seasonal, AutoRegressive, Integrated, Moving Average, eXogeneus).

Antes de intentar modelar nuestros datos con los modelos que veremos en posteriores capítulos, que son de naturaleza más compleja que los SARIMAX, se intentó modelizarlo de esta manera, sin llegar a resultados interesantes. Uno de los motivos podría ser que los supuestos sobre los que se basan estos modelos, no se cumplen en el tipo de problema que estamos estudiando.

- Por ejemplo, para los modelos de tipo AR hemos visto que, para predecir las ventas en un mes determinado, hacemos una regresión de las ventas frente a las ventas del mes anterior. Por lo tanto, tiene que haber una relación sólida entre ambos valores (para valores altos de ventas en un mes, los valores de ventas del mes anterior también son altos, y viceversa). Veremos que la variable objetivo de este problema (Hidrometría) no tiene este tipo de relación sólida con sus valores pasados, depende de factores tan imprevisibles a corto plazo como las Precipitaciones. En pocas palabras: *la imprevisibilidad de las variables predictoras nos supone un problema*.
- Además, aunque en los modelos SARIMAX podemos considerar variables exógenas para realizar las predicciones (es decir, para predecir la Hidrometría podemos usar no sólo los propios valores pasados de la Hidrometría, sino también valores presentes de Temperatura y Precipitaciones), esto no es suficiente. Para

predecir la Hidrometría en un momento determinado puede ser necesario conocer no sólo las Precipitaciones o Temperaturas de ese día, sino también las de días anteriores. Además, las relaciones entre las variables exógenas y la objetivo deben de ser lineales para que funcione un SARIMAX. En pocas palabras, *las relaciones no lineales entre las variables* nos supone un problema, y la *importancia de valores pasados de las variables exógenas*, también.

- Por último, indicar que para poder utilizar este tipo de modelos la serie temporal debe ser *estacionaria*, y la nuestra no lo es.

Todo esto no quiere decir que no podamos captar patrones para predecir la Hidrometría, simplemente significa que quizá estos modelos no capten el tipo de patrones que se dan en nuestro problema. Ello nos obliga a considerar otro tipo de modelos.

2.2.4 Predicción de series temporales. Machine Learning

En esta sección, veremos cómo podemos integrar los conocimientos que hemos adquirido en el máster sobre Machine Learning, para poder adaptarlos a la resolución de problemas de series temporales.

Hay dos tipos de features que son específicas de series temporales, a diferencia de otros problemas modelables con ML: time-step features y lag features (Holbrook, R., 2023). Quedémonos por ahora con esta idea básica, un poco más adelante la desarrollaremos y la pondremos en contexto.

Por otra parte, para describir la evolución de una serie temporal, podemos detectar tres tipos de componentes: tendencia, estacionalidad, ciclos (Holbrook, R., 2023). Esta descomposición nos ayudará, a la hora de tener que analizar una serie temporal, para ir detectando patrones de diferente naturaleza en los datos. Una vez bien identificados los patrones, será más fácil entender cómo modelarlos y predecirlos.

2.2.4.1 *Tendencia*

Las tendencias son evoluciones a largo plazo de nuestra serie temporal.

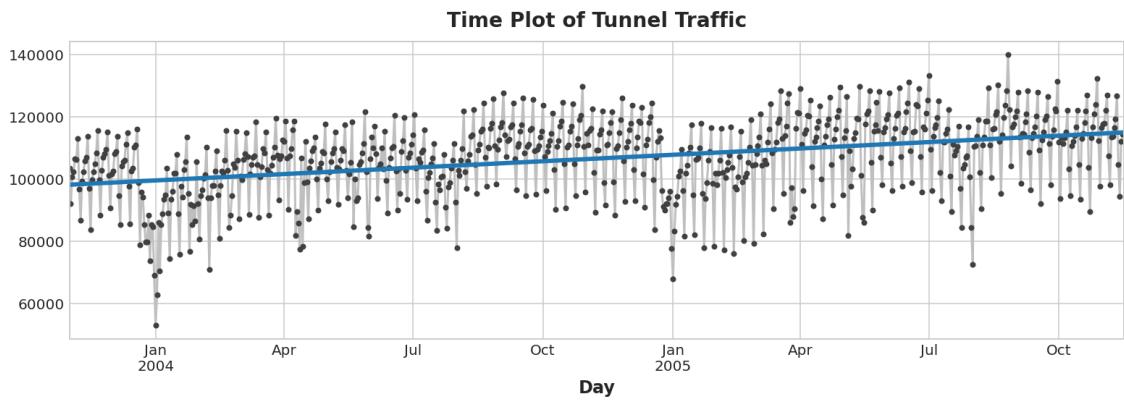


Imagen extraída de: <https://www.kaggle.com/code/ryanholbrook/linear-regression-with-time-series>

Por ejemplo, para esta serie temporal sobre tráfico en un túnel, vemos que el tráfico experimenta subidas y bajadas a corto plazo (línea gris), pero a largo plazo se observa que la tendencia es ligeramente ascendente (línea azul). Es decir, hay una relación entre una variable independiente (feature, en este caso el tiempo), y una variable dependiente que queremos predecir (target, en este caso el tráfico).

Nuestros datos de partida sobre este problema, tendrían la siguiente forma:

	NumVehicles
Day	
2003-11-01	103536
2003-11-02	92051
2003-11-03	100795
2003-11-04	102352
2003-11-05	106569

Imagen extraída de: <https://www.kaggle.com/code/ryanholbrook/linear-regression-with-time-series>

Para modelizar la relación respecto al tiempo, tenemos que introducir el time-step como feature (como ya habíamos indicado, esta es una de las grandes particularidades de las series temporales). Una forma muy sencilla sería la siguiente: asignar a cada día para el que hemos recogido observaciones de tráfico, un número secuencial que aumente de forma cronológica:

	NumVehicles	Time
Day		
2003-11-01	103536	0
2003-11-02	92051	1
2003-11-03	100795	2
2003-11-04	102352	3
2003-11-05	106569	4

Imagen extraída de: <https://www.kaggle.com/code/ryanholbrook/linear-regression-with-time-series>

Utilizamos un modelo sencillo (por ejemplo, regresión lineal), con “Time” como feature y “NumVehicles” como target, y tendríamos un sencillísimo primer modelo de series temporales.

2.2.4.2 Estacionalidad

Una serie temporal tiene estacionalidad (*seasonality*) cuando hay un cambio regular y periódico en la media de la serie temporal. Los patrones estacionales normalmente tienen que ver con repeticiones en la fecha/tiempo (patrones horarios, diarios, anuales, etc.) (Holbrook, R., 2023).

Es decir, respecto a la tendencia global que vimos antes para una serie temporal, queremos ahora “hilar más fino”, detectar posibles repeticiones en el tiempo que expliquen las subidas y bajadas que se inscriben dentro de la tendencia general.

Imaginemos que en la serie temporal sobre tráfico en un túnel, hemos detectado que existe un patrón repetitivo de tipo semanal. Por ejemplo, hacia principios de semana suele haber menos tráfico, y a medida que se va acercando el fin de semana ese tráfico va creciendo. Para poder modelizarlo, tendremos que “indicar” al modelo a qué momento de la semana pertenece cada una de las observaciones sobre el tráfico (medición del número de vehículos).

Nuevamente, se trata de introducir una time-step como feature, solo que esta vez no va a ser un número incremental como en el caso de la tendencia, sino un indicador de en qué momento de la semana estamos. Una forma muy sencilla sería añadir “Día de la semana” (número del 1 al 7) como feature. Ahora bien, hay métodos más precisos para este tipo de modelización que veremos más adelante (4.2 Búsqueda de variables predictoras adicionales - Modelo de Random Forest).

Una vez introducimos la feature sobre el momento de la semana en nuestro modelo sobre el tráfico en un túnel, conseguimos captar esos patrones que escapaban en el sencillo modelo que teníamos para tendencia.

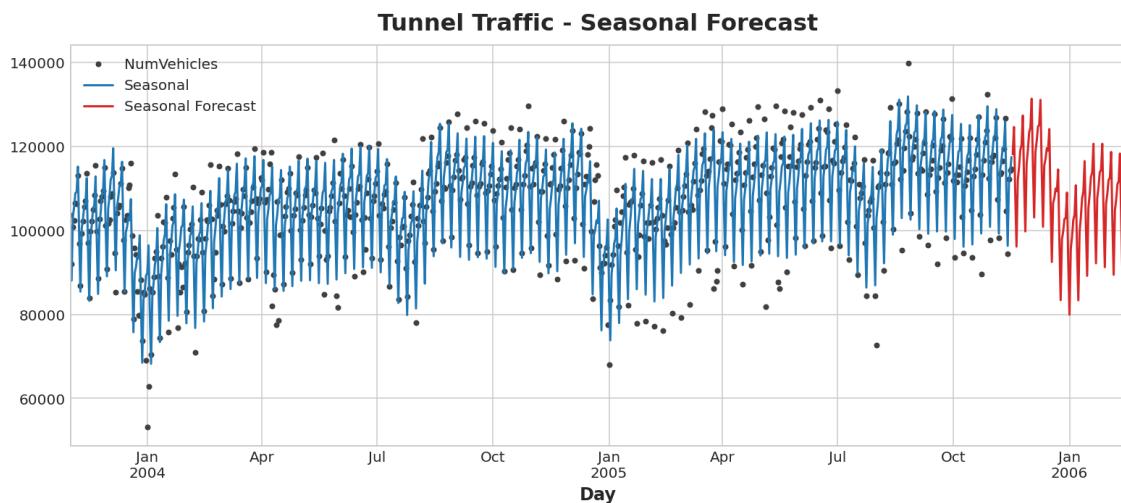


Imagen extraída de: <https://www.kaggle.com/code/ryanholbrook/seasonality>

2.2.4.3 Ciclos

Para los dos tipos de patrones anteriores, tendencia y estacionalidad, hay un denominador común: se trata de patrones que dependen del tiempo. Pero en las series temporales podemos encontrarnos con patrones que no dependen del tiempo, como son los ciclos.

En los ciclos, el valor de una serie temporal no depende de una variable de tipo temporal, sino de los propios valores de la serie en momentos previos. Los comportamientos cíclicos son característicos de sistemas que se pueden afectar a sí mismos, o cuyas reacciones persisten con el tiempo (Holbrook, R., 2023).

Veamos algunos ejemplos de series temporales que siguen patrones cíclicos:

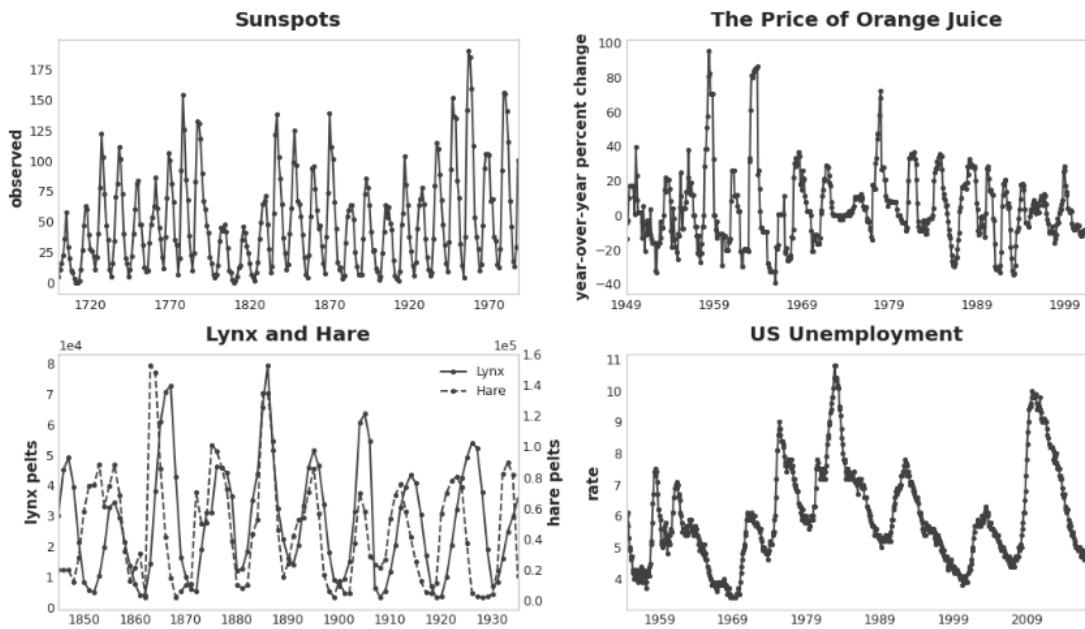
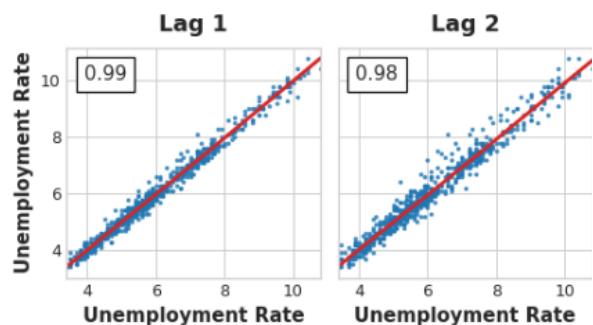


Imagen extraída de: <https://www.kaggle.com/code/ryanholbrook/time-series-as-features>

Fijémonos por ejemplo, en la serie temporal de la esquina inferior derecha (Tasa de desempleo en Estados Unidos). Como podemos ver, esta serie experimenta subidas y bajadas, pero no lo hace de forma regular respecto a un periodo temporal. A veces dos picos de desempleo consecutivos ocurren con unos pocos años de diferencia, a veces la diferencia entre ellos es de casi una década. De nada nos valdría intentar modelizar esta serie añadiendo una feature estacional, como hicimos anteriormente con el momento de la semana. Es aquí donde entra en juego el otro tipo de features característico de las series temporales: lag features.

Para modelizar este tipo de series, lo que hay que estudiar es cómo los valores pasados de la serie influyen en el valor actual. Una forma muy sencilla e intuitiva de estudiar estas relaciones son los lag plots.

	y	y_lag_1	y_lag_2
Date			
1954-07	5.8	NaN	NaN
1954-08	6.0	5.8	NaN
1954-09	6.1	6.0	5.8
1954-10	5.7	6.1	6.0
1954-11	5.3	5.7	6.1



Imágenes extraídas de: <https://www.kaggle.com/code/ryanholbrook/time-series-as-features>

Vemos en la imagen de la izquierda que, en primer lugar, creamos las lag features. Éstas no son más que los propios valores de la serie temporal en instante pasados, siendo “lag 1” la tasa de desempleo del mes anterior, y “lag 2” la tasa de desempleo de hace dos meses. En los diagramas de la derecha, enfrentamos valores pasados contra el actual y analizamos la relación. Vemos que la tasa de desempleo del mes actual está muy relacionada tanto con la tasa de empleo del mes anterior, como con la tasa de empleo de dos meses anteriores. Es una relación lineal y directamente proporcional: cuanto más alta ha sido la tasa de desempleo en el mes anterior, más alta se espera que sea en el mes actual. Por lo tanto, ambas parecen features prometedoras para ser incluidas en un modelo que realice predicciones.

Respecto a qué lags de una serie temporal son los más interesantes para incluir como features, simplemente señalar que hay una serie de conceptos muy útiles para ayudarnos a escoger. Por razones de simplicidad, no los vamos a incluir en esta exposición.

2.2.4.4 Variables exógenas

Hasta ahora, hemos visto que para predecir series temporales se puede incluir en los modelos 2 tipos de variables: aquellas relacionadas con el tiempo, y aquellas relacionadas con valores de la propia serie en instantes pasados. Pero una serie temporal puede depender de otro tipo de variables.

Por ejemplo, imaginemos que, en el túnel que estudiamos anteriormente, ha habido una ampliación de carriles en el intervalo de tiempo recogido en las observaciones. Eso probablemente aumentaría el número de vehículos que transitan, y no podemos captar ese patrón sólo con time-step features. Habría que introducir una variable exógena “Número de carriles de la calzada”.

De la misma forma, en nuestra serie que tiene un patrón cíclico (Desempleo en USA), podría ser que los valores pasados no consigan explicar con suficiente precisión el valor actual de desempleo. Una feature que sirva como indicadora del estado de la economía nacional (PIB, por ejemplo) podría ayudar a afinar las predicciones.

Estas variables exógenas se pueden introducir como features en el modelo, de forma que se complementen con las time-step features y lag features.

2.2.4.5 Recapitulación

En pocas palabras, hemos visto que los patrones presentes en las series temporales son de tres tipos (tendencia, estacionalidad, ciclos), y que para modelizarlos tenemos que incluir variables de tipo temporal (time-step features), variables relacionadas con valores pasados de la serie temporal (lag features), así como variables externas como en cualquier otro problema de machine learning (variables exógenas).

Sin embargo, los modelos SARIMAX (recordemos, Seasonal AutoRegressive Integrated MovingAverage eXogeneus) son capaces de abordar todos estos tipos de patrones, considero que esta no es la gran ventaja de utilizar ML respecto a SARIMAX.

Como ya se ha indicado al final de la sección 2.2.3 Predicción de series temporales. Modelos ARIMA: *la imprevisibilidad de las variables predictoras, las relaciones no lineales entre las variables y la importancia de valores pasados de las variables exógenas*; todas ellas nos desanimaban a utilizar SARIMAX para nuestro problema de predicción de Hidrometría. Utilizando un modelo de machine learning, podemos solventar este tipo de obstáculos.

Los modelos de ML nos dan mucha más variedad, podemos elegir entre diferentes tipos de modelos, dependiendo de cómo sean las relaciones entre las variables de estudio en nuestro problema. Y, sobre todo, considero que son modelos mucho más flexibles y fáciles de entender que los SARIMAX. Ello nos ayudará para abordar las dos grandes singularidades de nuestro problema: lo imprevisibles que son las lluvias, y lo importante que es saber cuánto ha llovido en días pasados para poder predecir los niveles actuales de un río.

Aquí es donde ha residido el gran desafío para resolver este problema. Los enfoques que se utilizan habitualmente para modelizar series temporales, se basan en supuestos que no se cumplen en nuestro problema de estudio. Ya no se trata sólo de no poder utilizar SARIMAX, incluso algunos de los enfoques más típicos para modelizar series temporales con ML no nos resultaban válidos. Por ejemplo, algunas librerías de Python que se valoraron para utilizarlas en este problema, sí que estaban muy bien preparadas si quisiésemos considerar valores pasados de Hidrometría como la principal influencia en los valores actuales de Hidrometría. Pero no era posible parametrizar valores pasados de una variable que no fuese la variable objetivo.

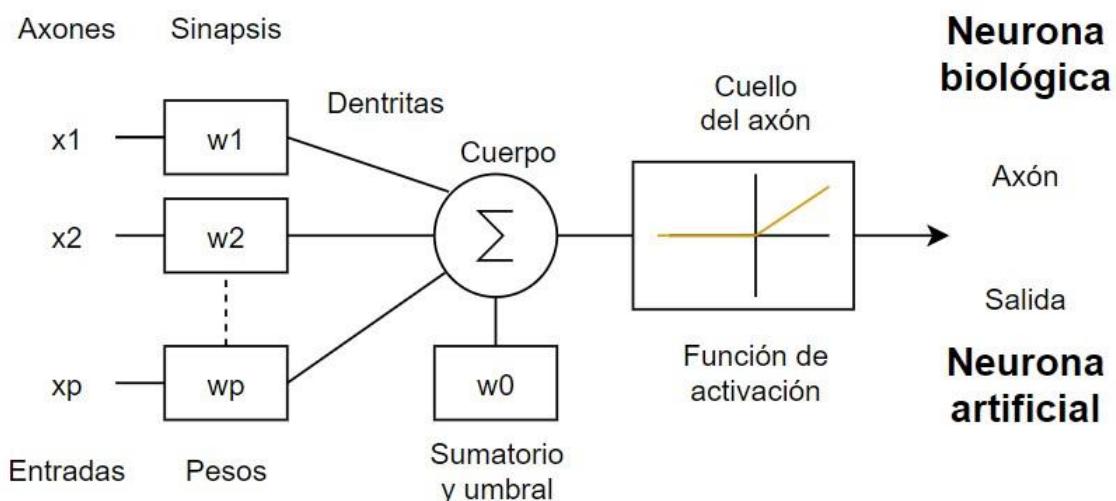
Finalmente, se encontró un modelo que resolvía muy bien todos estos desafíos planteados, y fue el que se utilizó definitivamente para resolver este problema: se trata de un modelo de Redes Neuronales Recurrentes LSTM.

2.2.5 Redes Neuronales Recurrentes LSTM

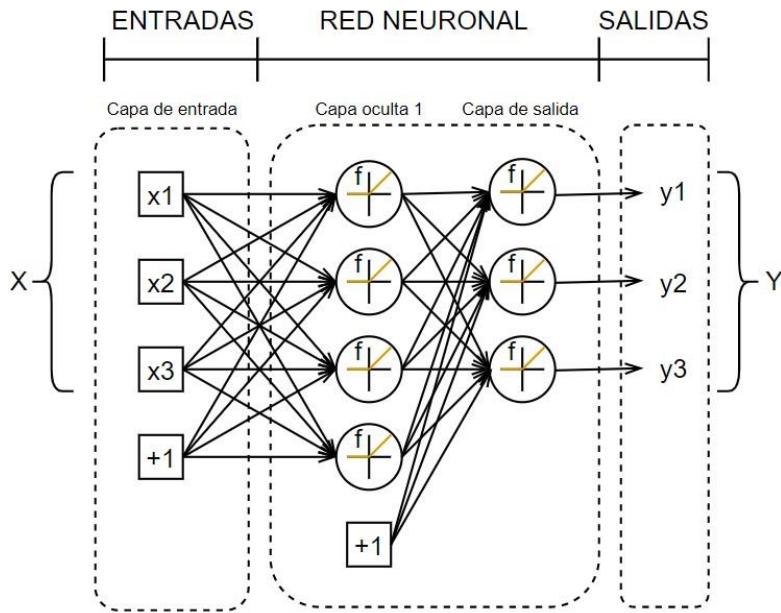
Las redes neuronales recurrentes LSTM son una variante de las redes neuronales recurrentes, que a su vez son una variante de las redes neuronales artificiales.

2.2.5.1 Redes neuronales artificiales

Tal y como aprendimos en el módulo 6 – unidad 5 (Aprendizaje automático - Modalidades y técnicas de Deep Learning), las redes neuronales son estructuras lógicas inspiradas en la neurociencia. Intentan imitar el comportamiento del cerebro a partir de neuronas artificiales interconectadas. Las entradas a cada neurona artificial (x) son multiplicadas por unos pesos (w) y sumadas, y su resultado es transformado por medio de una función que se denomina función de activación.



Las redes neuronales denominadas perceptrón multicapa son las de uso más común en problemas de clasificación y de regresión con datos estructurados. El perceptrón multicapa está formado por neuronas interconectadas con una organización jerárquica. En el siguiente ejemplo, se ilustra una red neuronal de perceptrón multicapa con dos capas ocultas:



El entrenamiento de este tipo de redes consiste en actualizar los pesos hasta que se consigue minimizar la diferencia entre los valores predichos por la red, y los valores reales. Así, a partir de las variables predictoras (x), podemos obtener el valor predicho (y).

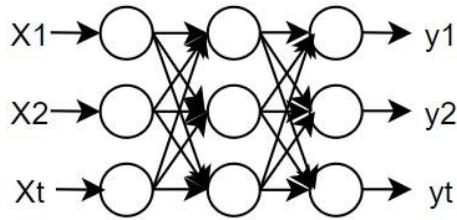
Una de las ventajas de este tipo de redes, y que nos interesa mucho de cara al tipo de problema que queremos resolver, es que tienen la capacidad de aprender relaciones no lineales y complejas entre las variables de estudio (Mahanta, J., 2017).

Ahora bien, como ya hemos visto, las series temporales son un tipo de problema de datos estructurados que tiene una serie de peculiaridades, como puede ser la importancia de valores pasados de la serie en la predicción de los siguientes valores de la serie.

2.2.5.2 Redes neuronales recurrentes

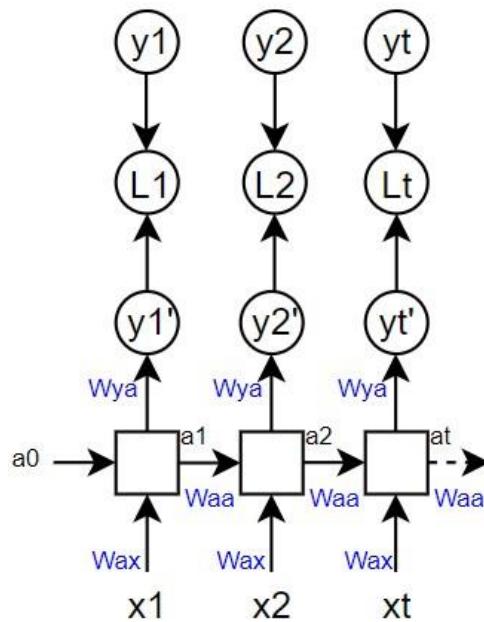
Las redes neuronales recurrentes (recurrent neural networks o RNN) son una familia de redes neuronales para procesar datos en secuencias.

Imaginemos que queremos resolver un problema de series temporales utilizando una red neuronal de perceptrón multicapa estándar. Podríamos plantear una arquitectura de red de esta forma:



Donde X_n serían los valores ordenados cronológicamente de una serie temporal, e y_n sería lo que queremos predecir, es decir el siguiente valor de la serie: para X_1 sería X_2 , para X_2 sería X_3 , etc. Para poder captar correctamente los patrones que se van formando a lo largo de la secuencia, de X_1 a X_t , necesitamos modificar la arquitectura.

Las redes neuronales recurrentes lo que hacen es que, además de recibir cada entrada, reciben una salida con cierta información de la entrada anterior.



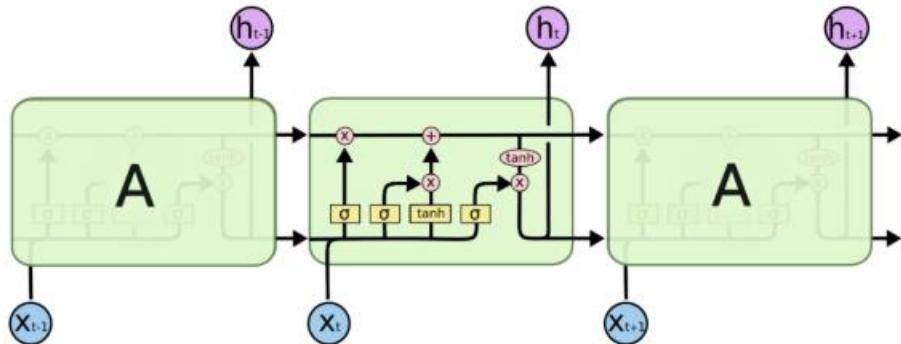
Así, la red puede realizar predicciones para X_t , teniendo en cuenta lo que ha ocurrido en las predicciones previas de la secuencia.

Sin embargo, las RNN sólo recuerdan una cantidad pequeña de valores previos en la secuencia, no son apropiados para recordar largas secuencias de datos (Siami Namin, S; Siami Namin, A., 2018).

2.2.5.3 Redes neuronales LSTM

LSTM (acrónimo de Long-Short Term Memory), son un tipo especial de RNN con características adicionales para memorizar secuencias de datos (Siami Namin, S; Siami

Namin, A., 2018). La memorización de la tendencia previa de los datos es posible a través de ciertas “puertas” y una línea de memoria en la arquitectura de la red.



Esta capacidad de las redes LSTM de retener información de momentos que no son inmediatamente recientes respecto al momento que queremos predecir, nos será de gran utilidad para la resolución de nuestro problema de predicción de Hidrometría. Como veremos a lo largo de la resolución del problema, para poder predecir la Hidrometría correctamente no sólo es necesario conocer los valores de las variables predictoras (Precipitaciones, Temperatura, la propia Hidrometría) en los días anteriores. Hay cierta información del histórico que también es necesario tener en cuenta.

2.3 Entorno para la resolución del problema

En los capítulos siguientes, manipulamos los datos para resolver el problema que hemos planteado. Como ya se indicó en el capítulo 2.1.3 Datos disponibles para realizar la predicción, el punto de partida son los datos de nuestra serie temporal, capturados en un fichero .csv.

El procesado, análisis, modelado, y predicción de los datos se realiza en un Jupyter Notebook. El lenguaje de programación utilizado es Python, ayudándonos de algunas de las librerías más populares para ciencia de datos: Pandas para procesamiento de datos, Matplotlib para visualización, Scikit-learn para modelado de Machine Learning. Además, se utilizará Tensorflow para realizar el modelado con Redes Neuronales LSTM.

3. Preprocesado de datos y Análisis exploratorio (primera parte)

En primer lugar, cargamos los datos contenidos en un fichero .csv a un dataframe Pandas, y observamos las diferentes variables de estudio que disponemos.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 8217 entries, 1998-01-01 to 2020-06-30
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Rainfall_Le_Croci    6026 non-null   float64
 1   Rainfall_Cavallina   6026 non-null   float64
 2   Rainfall_S_Agata     6026 non-null   float64
 3   Rainfall_Mangona     6026 non-null   float64
 4   Rainfall_S_Piero     6026 non-null   float64
 5   Rainfall_Vernio      4283 non-null   float64
 6   Rainfall_Stia        1283 non-null   float64
 7   Rainfall_Consuma     1283 non-null   float64
 8   Rainfall_Incisa      4568 non-null   float64
 9   Rainfall_Montevarchi 1647 non-null   float64
 10  Rainfall_S_Savino    1283 non-null   float64
 11  Rainfall_Laterina    1283 non-null   float64
 12  Rainfall_Bibbiena    2378 non-null   float64
 13  Rainfall_Camaldoli   1283 non-null   float64
 14  Temperature_Firenze 6192 non-null   float64
 15  Hydrometry_Nave_di_Rosano 8169 non-null   float64
dtypes: float64(16)
memory usage: 1.1 MB
```

Como podemos ver, los datos que disponemos abarcan un intervalo de tiempo desde el 1998-01-01 hasta el 2020-06-30 (del 1 de Enero de 1998 hasta el 30 de Junio de 2020, en adelante utilizaremos esta nomenclatura Año-Mes-Día para las fechas). Las variables #0 a #13 representan las Precipitaciones (mm/m^2) recogidas en diferentes pluviómetros, la variable #14 representa la Temperatura ($^{\circ}\text{C}$), y la variable #15, que es la variable objetivo y que se tratará de predecir, es la Hidrometría (medida en metros, m., nivel del río expresado en altura) en la estación de medición de Nave di Rosano.

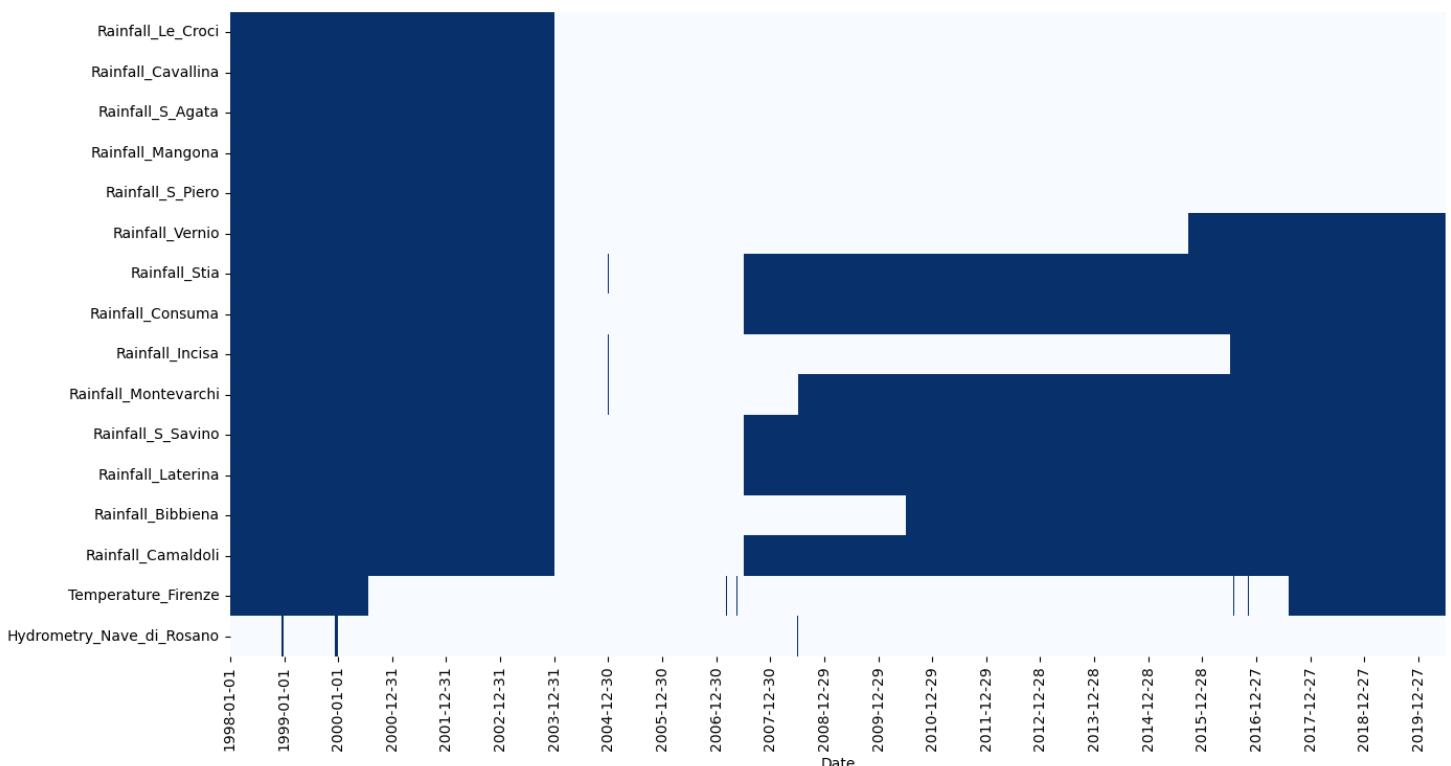
Los datos tienen granularidad día, es decir, en esta serie temporal hay una observación por cada uno de los días comprendidos en el intervalo de tiempo.

Una de las primeras cosas que llama la atención, es que tenemos una cantidad significativa de valores nulos para muchas de las variables de estudio. Comenzaremos por abordar este aspecto.

3.1 Tratamiento de valores nulos

A partir de los datos de nuestro dataframe, se construye una visualización en la que se podrá detectar los puntos en el tiempo en los que tenemos valores faltantes en cada una de las variables. Para cada variable de estudio (eje vertical), durante todo el intervalo de tiempo estudiado (eje horizontal), se marca en azul los registros que no aparecen en nuestros datos (valores nulos).

Valores nulos (azul)



A continuación, se analizan los resultados obtenidos, y las decisiones que se tomarán a nivel de tratamiento de datos.

- Vemos que, desde el 1998-01-01 (inicio de las mediciones) hasta 2003-12-31, apenas tenemos datos de las variables de estudio. Nos faltan la totalidad de datos acerca de las Precipitaciones, y parte de los datos de Temperatura.
 - Decisión: Se eliminarán todos los datos pertenecientes a este intervalo.
- Para algunos de los pluviómetros (Vernio, Stia, Consuma, Incisa, Montevarchi, S_Saviro, Laterina, Bibbiena, Camaldoli), falta información en un intervalo de tiempo muy significativo (el que va desde aproximadamente el 2006-12-30 hasta

el final de las mediciones). De hecho, se observa en la gráfica anterior que para estos pluviómetros tenemos más valores nulos, que no nulos.

- Decisión: Eliminaremos todos los datos de estos pluviómetros de nuestro estudio.
- A partir de finales de 2017, no hay mediciones de temperatura.
 - Decisión: Se recabarán esos datos, y se añadirán al dataframe.
- Fuera de las tres casuísticas relatadas anteriormente, se observa la presencia de unos pocos casos aislados de valores nulos para la Temperatura y la Hidrometría.
 - Decisión: se imputarán estos valores. Una vez hagamos el análisis exploratorio de estas dos variables, se determinará el método de imputación más apropiado.

Las dos primeras decisiones tomadas implican simplemente el eliminado de datos, no hace falta ahondar en explicaciones.

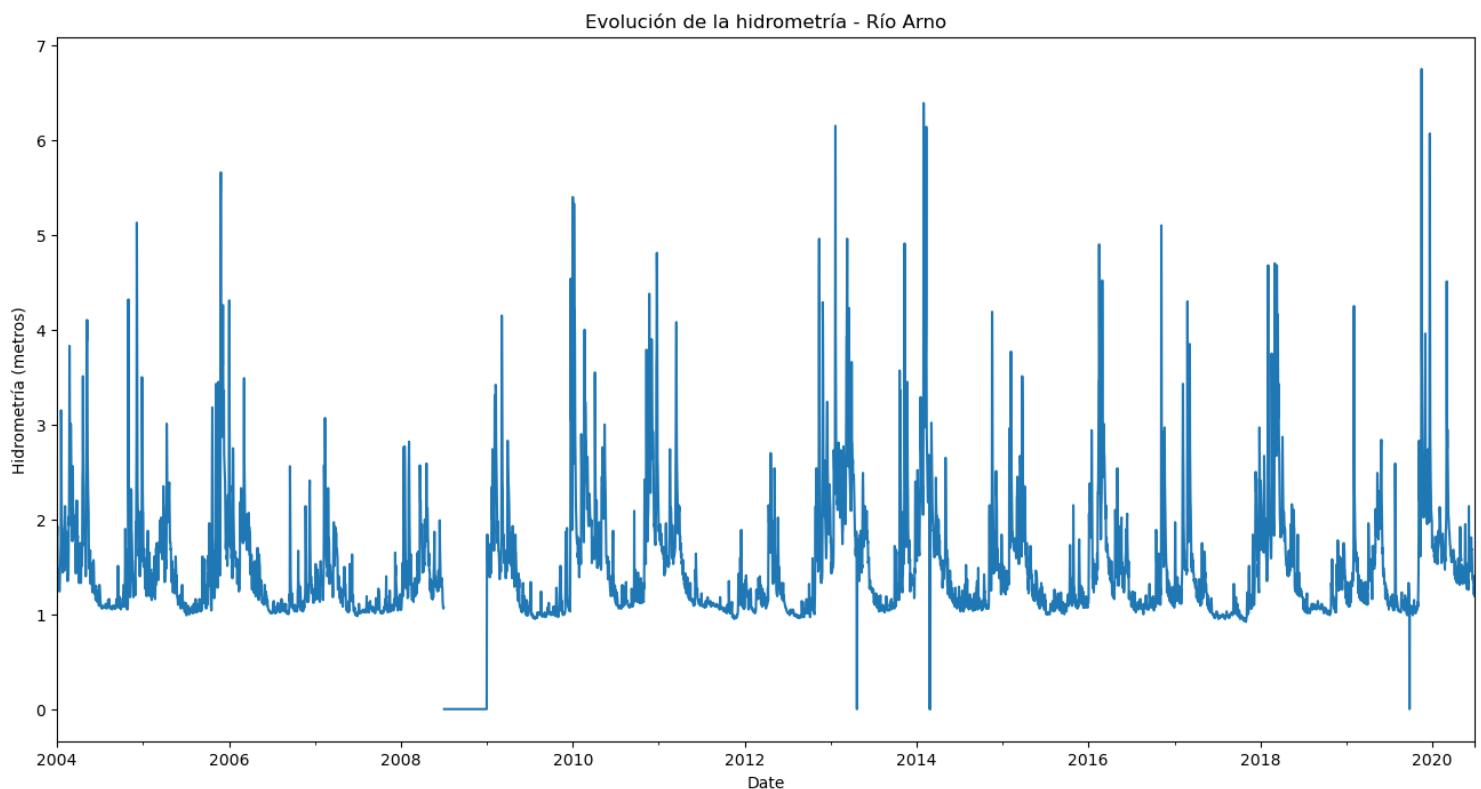
Respecto a cómo se han recabado datos de la Temperatura a partir de 2017, se ofrecen explicaciones en el Anexo 1: Recabar datos de Temperatura desde una fuente de datos externa.

Respecto a la imputación de valores nulos aislados, se llevará a cabo una vez se haga el análisis exploratorio de las variables en cuestión, eligiendo el método de imputación que mejor se adapte a cada variable.

Completado el tratamiento de nulos, ya estamos preparados para comenzar el análisis exploratorio de los datos.

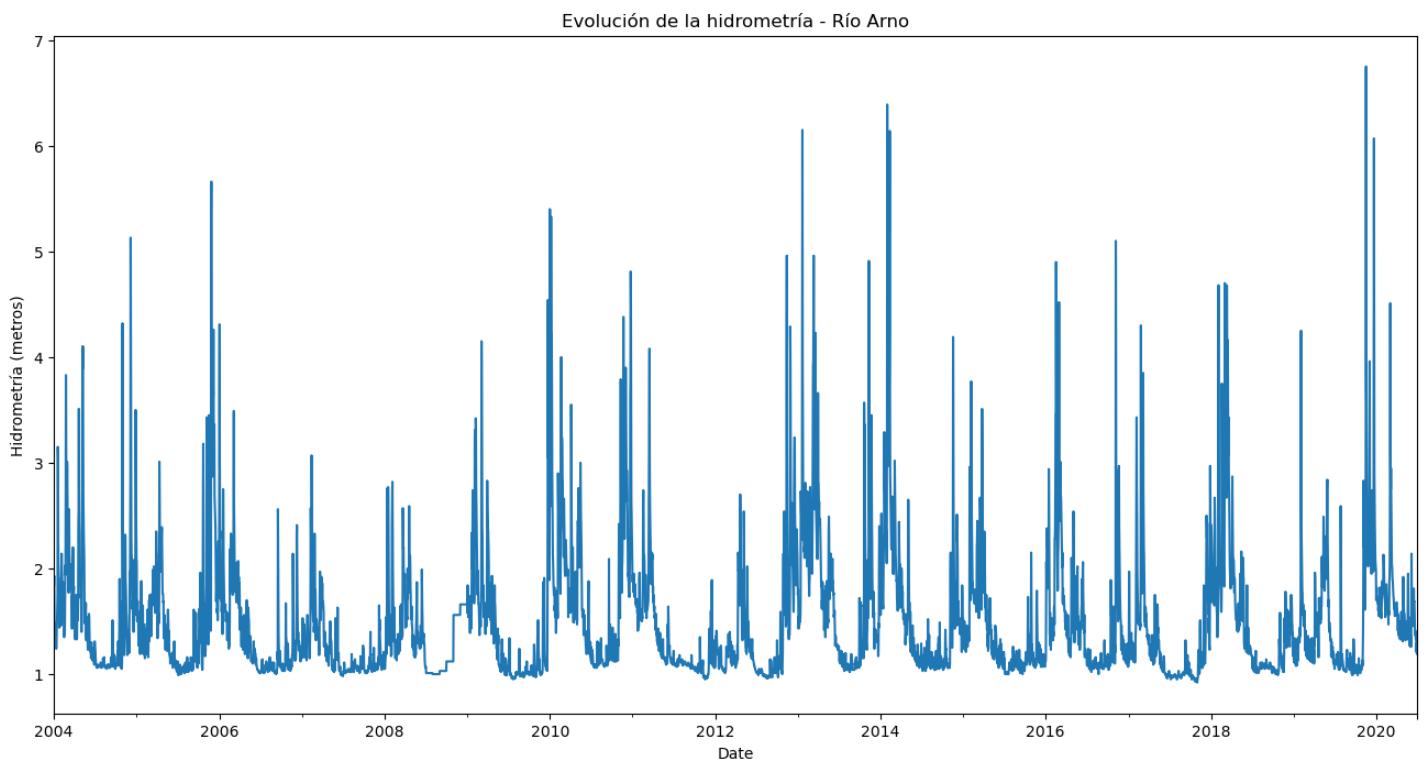
3.2 Análisis exploratorio: Variable target (Hidrometría)

Empezamos visualizando toda la serie temporal para nuestra variable target. Recordemos que, tras el tratamiento de nulos, el intervalo de tiempo que estudiaremos ha sido limitado al 2004-01-01 hasta 2020-06-30.



En primer lugar, como ya se había comentado, tratamos los valores atípicos ($= 0$) y nulos que aún estaban presentes. Se aprovecha la circunstancia de que de esta variable sigue un patrón en cierta medida cíclico, para imputar estos valores basándonos en el valor medio de la Hidrometría para el mes al que pertenezcan. Por ejemplo, si tenemos un valor atípico/nulo para el mes de Abril, lo imputaremos con el valor medio de la Hidrometría para todos los meses de Abril de la serie temporal.

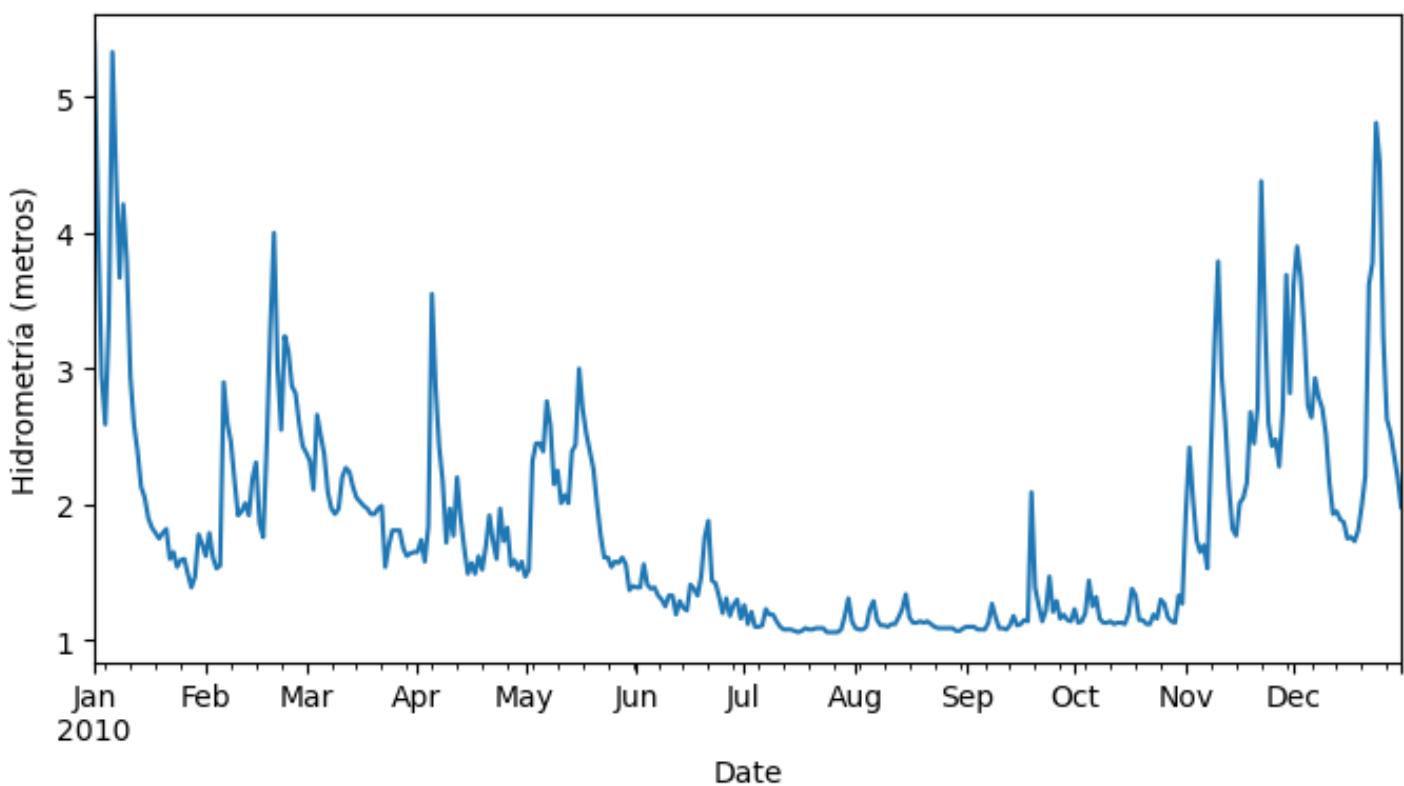
Una vez hechas las imputaciones, esta es la visualización de la serie temporal.



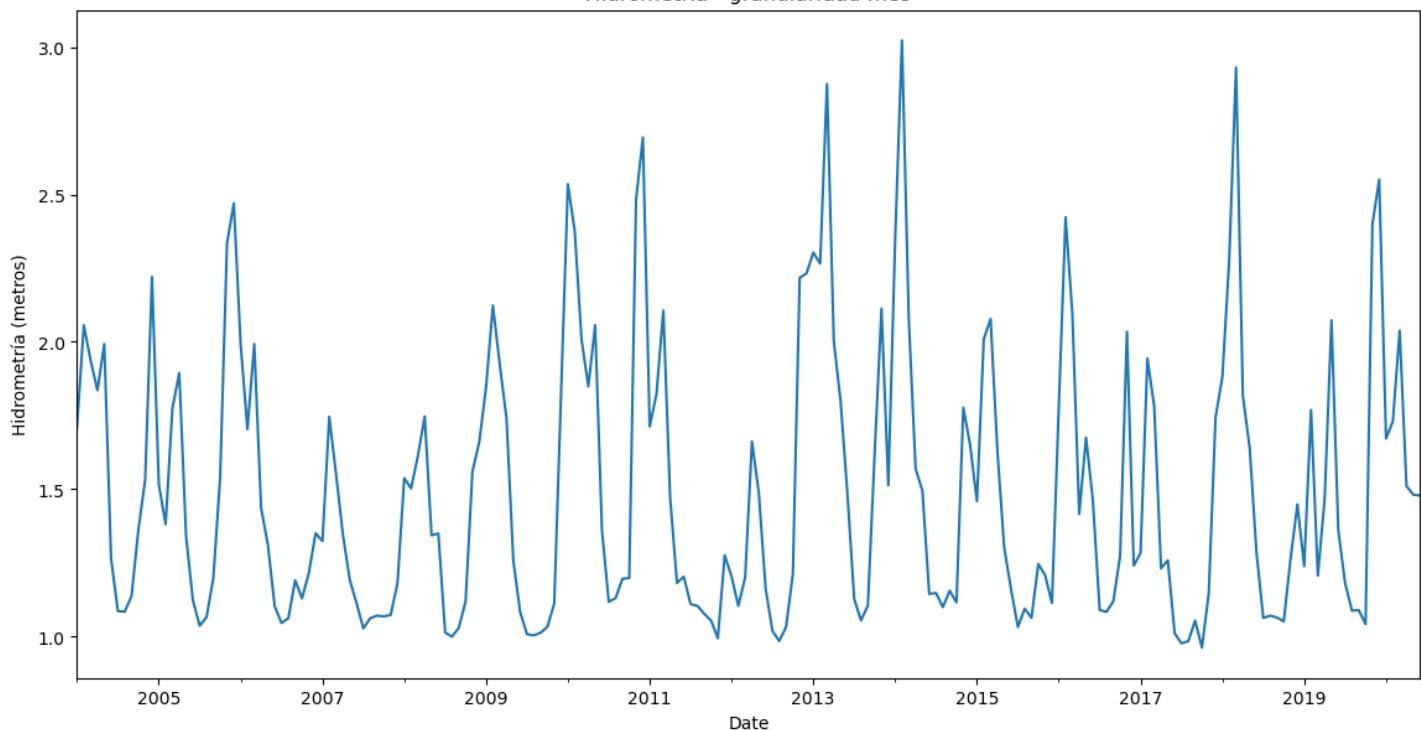
Además de la visualización de la serie temporal completa, vamos a incluir otras dos visualizaciones que facilitarán el análisis:

- Visualización de la serie temporal para un año en concreto. En la visualización anterior, al tener demasiados datos, las líneas se solapan entre ellas, observando un sólo año se observará mejor cómo evoluciona la serie.
- Visualización de la serie temporal con datos de granularidad mes, en vez de día. Nos interesa suavizar esas subidas y bajadas bruscas que se intuyen en la visualización anterior, para poder captar mejor las tendencias generales.

Hidrometría año 2010



Hidrometría - granularidad mes



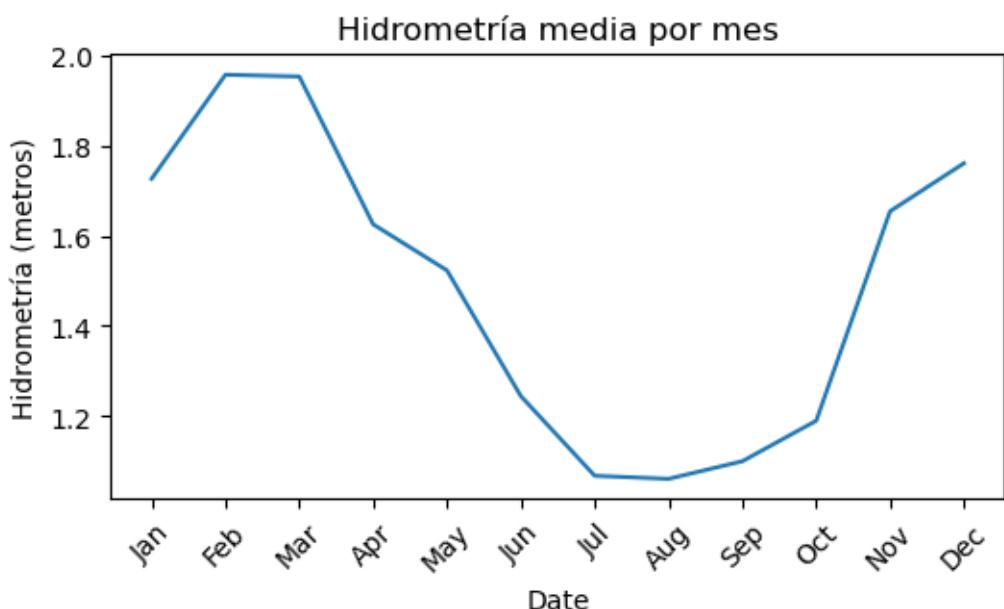
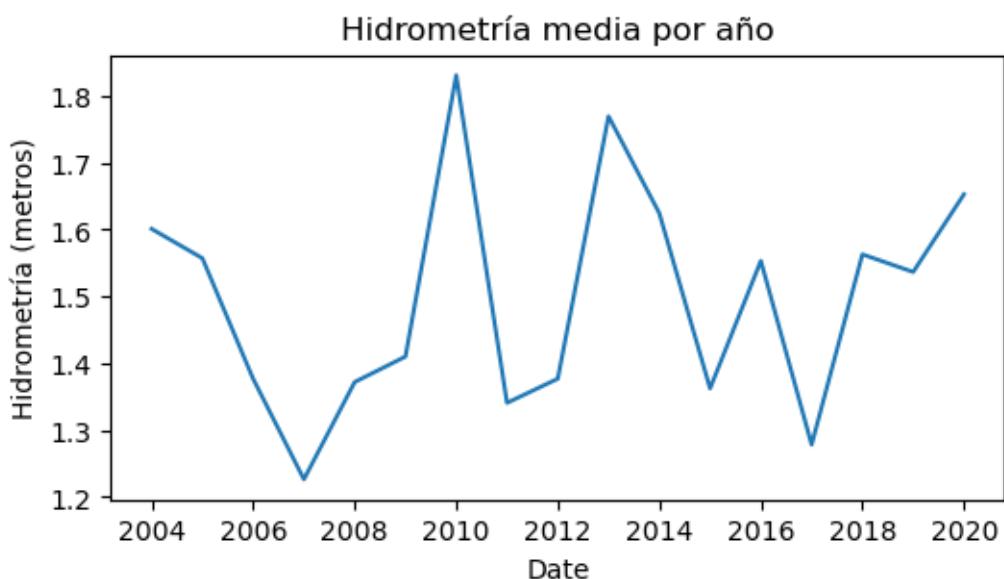
Como ya se había adelantado, se observa que los niveles del río siguen un patrón anual, con cierta variabilidad y ciertos matices cada año.

- Tendencia general: En los meses de verano se dan los niveles más bajos, en invierno los niveles más altos, y en primavera y otoño se ve una transición.
- Matiz: En la visualización para un año concreto, se aprecia que los niveles del río tienen muchos "picos": se dan subidas y bajadas bruscas del nivel.
- Matiz: En la visualización para granularidad mes, se ve que hay variaciones de año a año.
 - En invierno, hay años que el río lleva más caudal, y años que lleva bastante menos (la altura de las "montañas" que describe la línea, varían cada año)
 - Cada año, son meses diferentes los que tienen los caudales más altos (la forma de las "montañas" que describe la línea, son totalmente diferentes de año en año)

A la hora de analizar una serie temporal, estos matices han de ser tenidos muy en cuenta. Parece que se trata de una serie que sí que tiene un comportamiento *cíclico* (se van repitiendo ciertos patrones en el tiempo), pero no tiene *estacionalidad* (los patrones no ocurren en intervalos de tiempo regulares, y no siempre se dan de la misma forma).

En estas fases iniciales del análisis, vamos a centrarnos en explorar las tendencias generales. Nos olvidamos por ahora de esos matices a la tendencia general, en secciones posteriores ahondaremos sobre ellos.

Visualizamos los valores medios de la hidrometría para cada año y para cada mes, teniendo en cuenta todos los datos de la serie temporal.



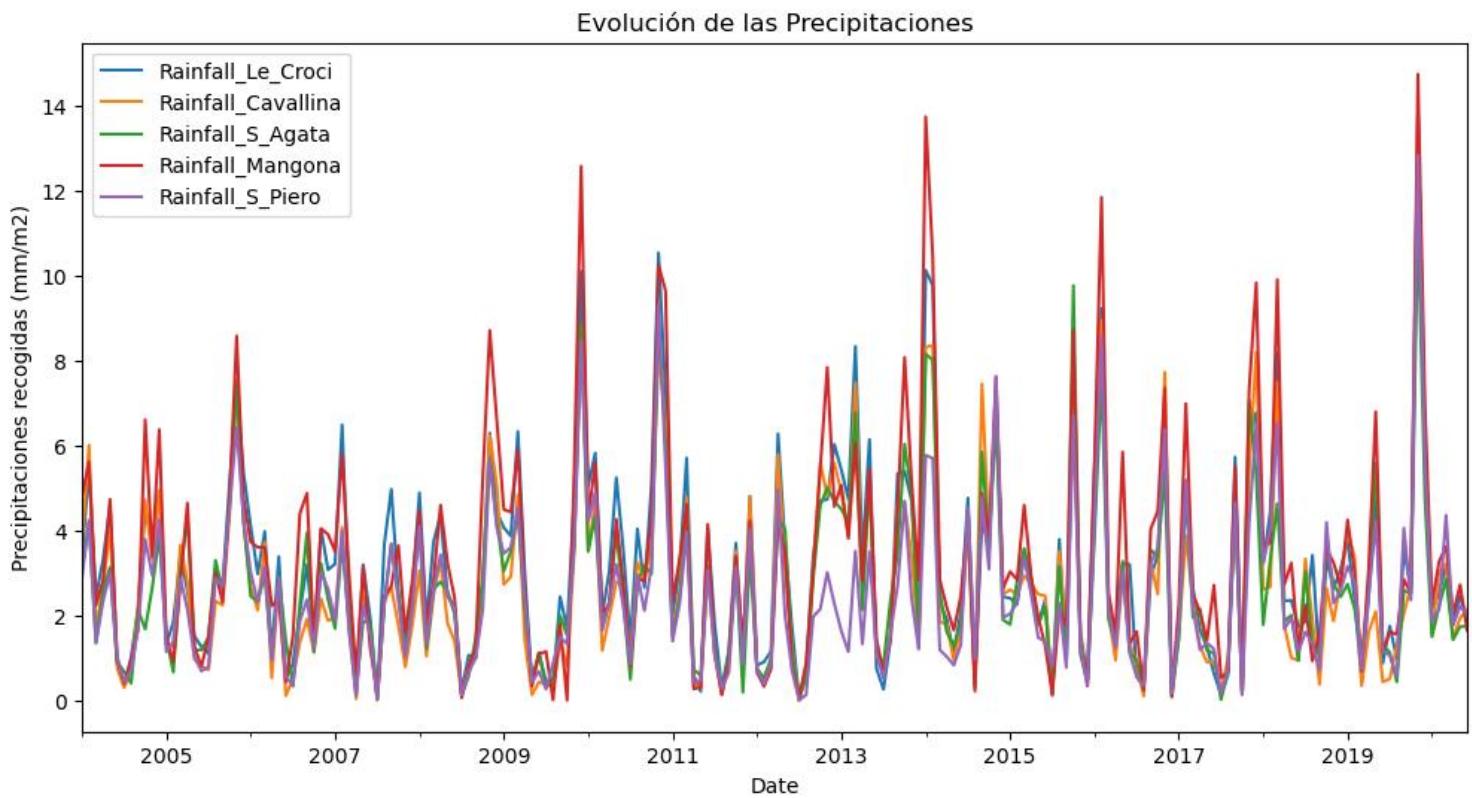
CONCLUSIONES

Los niveles medios del río, medidos de año en año, experimentan cierta variabilidad, oscilando entre 1.2 metros (año 2005) y 1.8 metros (año 2010), aproximadamente.

A su vez, dentro de cada año hay variabilidad dependiendo del mes del año, con una diferencia de casi 1 metro entre los valores medios de los meses con más caudal de río (Febrero y Marzo) y los meses con menos caudal (Julio y Agosto).

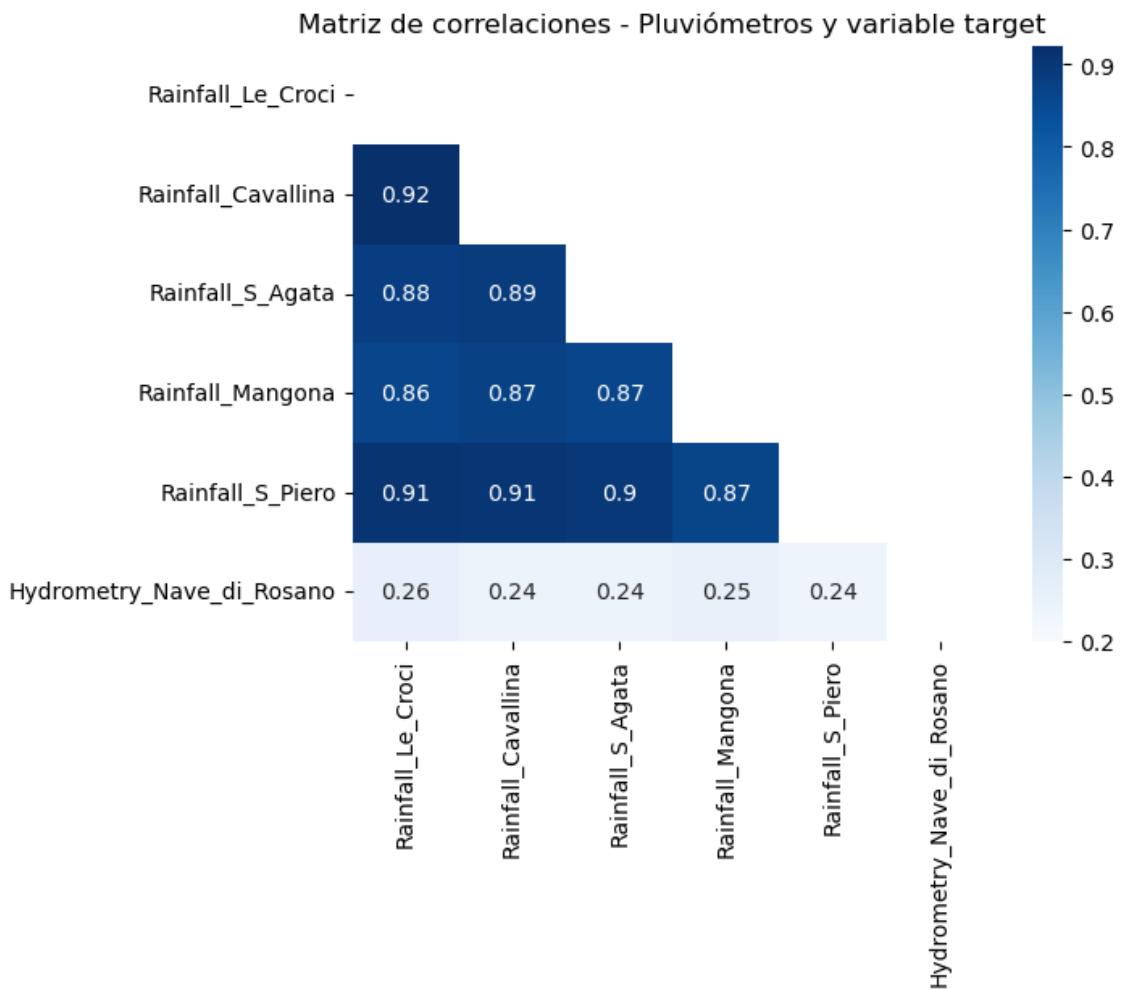
3.3 Análisis exploratorio: Precipitaciones

Visualizamos las series temporales de los diferentes puntos de medición de precipitación (variables "Rainfall_ ", pluviómetros). En este caso, para facilitar la interpretación de las mismas, se muestran las precipitaciones medias durante cada mes de la serie temporal.



Las líneas que representan los valores para cada uno de los pluviómetros, parece que se solapan bastante. Es posible que la información que nos aporten los diferentes pluviómetros sea similar, lo cual nos abre una oportunidad para simplificar el número de variables de estudio en nuestro problema.

Se va a explorar esta posible similitud de forma más cuantitativa, a través de los coeficientes de correlación. Se utiliza el método `.corr()` de la librería Pandas, utilizando el método de correlación por defecto (Pearson).



Vemos que la correlación entre todos y cada uno de los pluviómetros es muy alta. También vemos que la correlación de cada pluviómetro con la target es muy similar.

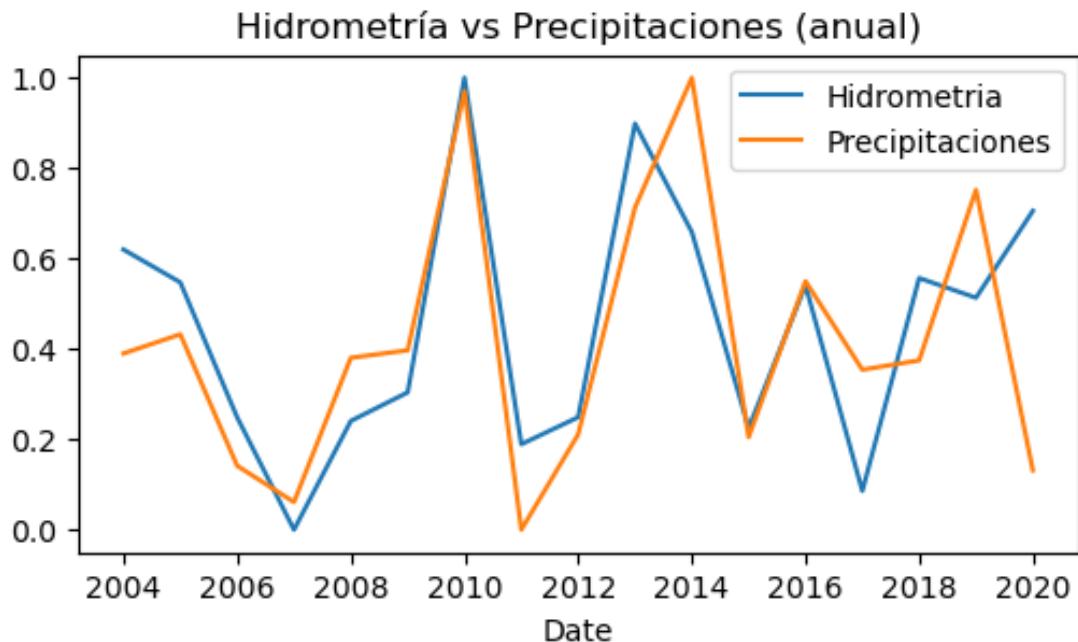
Para simplificar el análisis, se decide sustituir las variables que representan a cada pluviómetro, por una variable "Mean Rainfall" que represente la media de los valores recogidos diariamente en cada pluviómetro.

```
df['Mean_Rainfall'] = (df['Rainfall_Le_Croci'] + df['Rainfall_Cavallina'] + df['Rainfall_S_Agata']  
+ df['Rainfall_Mangona'] + df['Rainfall_S_Piero']) / 5  
df.drop(['Rainfall_Le_Croci', 'Rainfall_Cavallina', 'Rainfall_S_Agata', 'Rainfall_Mangona', 'Rainfall_S_Piero'],  
axis=1, inplace=True)
```

Intuitivamente, cabe esperar que las lluvias tengan una gran influencia en los niveles del río. Vamos a realizar algunas visualizaciones para testear esta hipótesis.

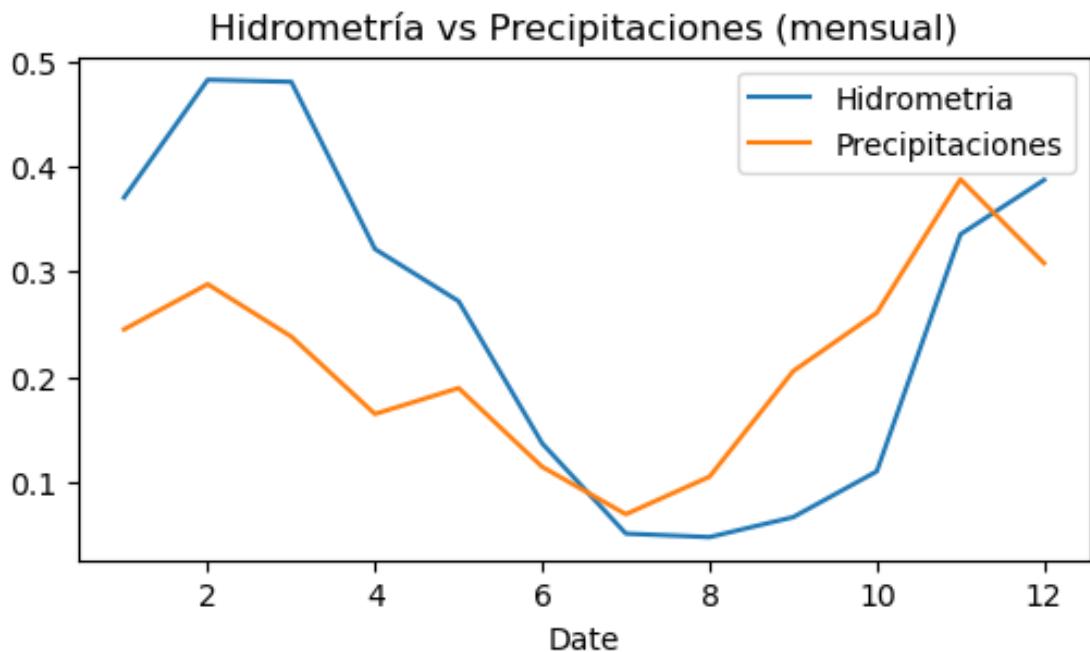
Se repiten las visualizaciones que habíamos mostrado al analizar la Hidrometría: Precipitación media por año, y Precipitación media por mes. Normalizamos los datos y los solapamos con las visualizaciones análogas de Hidrometría.

NOTA: Como método de normalización de los datos, se escoge una normalización min-max (Wikipedia, Feature Scaling, 2022). Parece una buena opción para este caso particular, puesto que al hacerla sobre datos agregados (por Mes/Año), se evitan valores extremos que hubiesen podido distorsionar los resultados.



En esta visualización, se observa cómo hay una correlación evidente entre los niveles medios anuales del río, y las precipitaciones recogidas. Cuanto más llueve un año determinado, mayores son los niveles del río. Eso sí, llama la atención cómo, para los últimos años de nuestra serie temporal (a partir de 2017), parece como si se perdiese esa correlación tan estrecha que hay en los años previos. Se analiza esta anomalía y sus posibles causas (ver Anexo 2: Análisis de valores anómalos en la correlación Hidrometría-Precipitaciones), se determina que quizás se pueda tratar de una anomalía relacionada con la forma de procesar los datos (agrupaciones y normalizaciones), y no necesariamente un patrón genuinamente presente en los mismos.

Prosiguiendo con el análisis, comparamos la Precipitación media por mes, frente a la Hidrometría media por mes.



Vemos que la tendencia es similar para ambas: llueve más y el río tiene más caudal en invierno, y ocurre lo contrario en verano. Eso sí, la amplitud de la oscilación de la hidrometría es más grande. Se observa una tendencia llamativa:

Parece que las lluvias tienen gran capacidad de provocar subidas del nivel del río en los meses de invierno (línea azul por encima de línea naranja), mientras que en verano ocurre lo contrario: aunque haya una cantidad apreciable de lluvia, no provoca un cambio muy acusado en los niveles del río.

Esto es un factor que puede que haya que tener en cuenta a la hora de modelizar: las lluvias influyen claramente en los niveles del río (cuanto más llueve, más nivel lleva el río), pero no influyen de la misma forma en todas las épocas del año. Puede ocurrir por diferentes motivos: ¿mayor capacidad de drenado del suelo?, ¿mayor evaporación a causa de la temperatura?

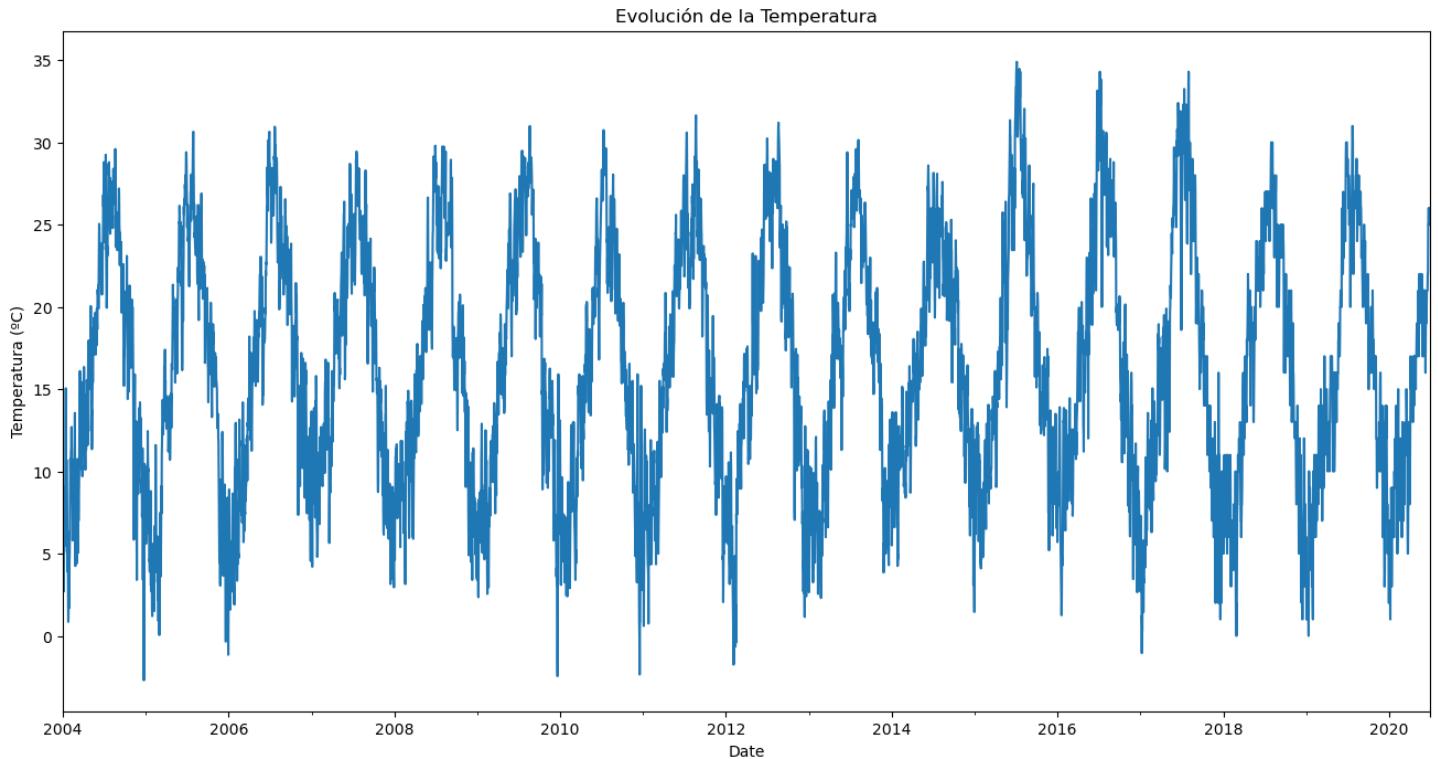
Precisamente es la temperatura nuestra siguiente variable de estudio.

CONCLUSIONES

Como era de esperar, hay una relación directamente proporcional entre la cantidad de precipitaciones y el nivel del río. Eso sí, hay ciertos matices en cómo cambia esta relación a lo largo del tiempo, que necesitaremos entender mejor antes de comenzar a modelar.

3.4 Análisis exploratorio: Temperatura

Visualizamos toda la serie temporal.



Recordemos que esta variable aún presentaba una pequeña cantidad de nulos. Al igual que ocurría para el caso de la Hidrometría, vemos que esta serie temporal tiene un patrón cíclico, así que nuevamente imputamos los valores nulos a partir de los valores medios de Temperatura del respectivo mes.

Llegados a este punto, ya no se tienen valores nulos en todo el dataframe:

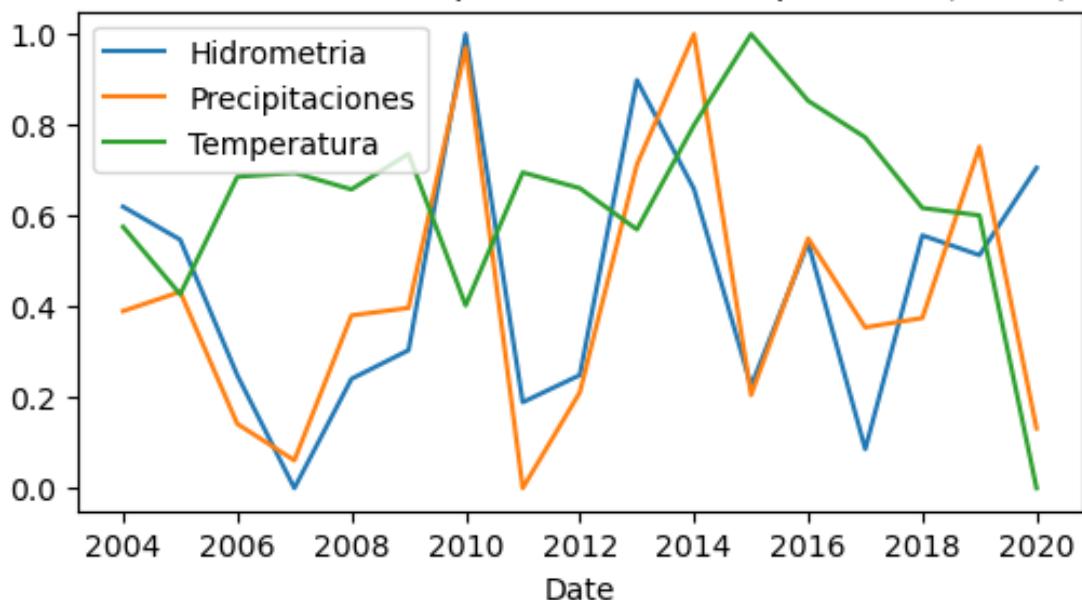
```
print('Número de nulos por variable:\n')
df.isna().sum()
```

Número de nulos por variable:

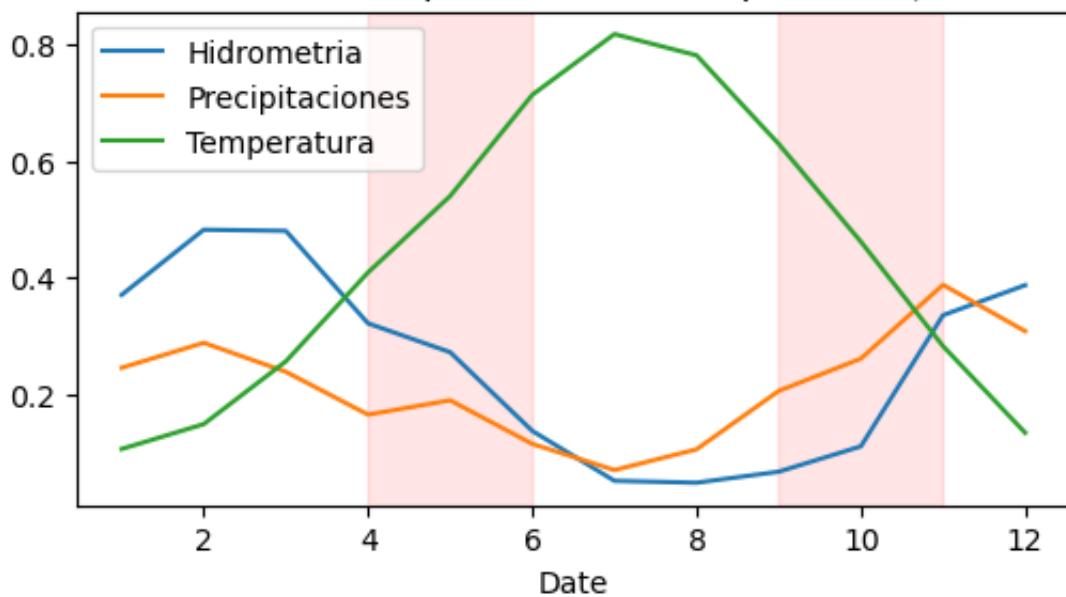
Temperature_Firenze	0
Hydrometry_Nave_di_Rosano	0
Mean Rainfall	0

Para profundizar en el análisis de esta variable, vamos a realizar nuevamente las visualizaciones de los datos agrupados por mes, y agrupados por año. Y los solaparemos a las mismas agrupaciones para Hidrometría y Precipitaciones.

Hidrometría vs Precipitaciones vs Temperatura (anual)



Hidrometría vs Precipitaciones vs Temperatura (mensual)



CONCLUSIONES

Como era de esperar, se observa que cuanta más Temperatura, menores son las Precipitaciones y menores es la Hidrometría.

Respecto a la pregunta lanzada anteriormente, de si la Temperatura podría ser la causante de que las lluvias tengan más o menos efecto en las subidas del río. Parece que no hay una relación clara, comparemos lo que ocurre en los intervalos entre Abril-Junio y Septiembre-Noviembre (regiones sombreadas en rojo en el gráfico anterior): las

Temperaturas son similares, pero entre Abril y Junio las Precipitaciones tienen mucho efecto sobre la Hidrometría, mientras que en Septiembre-Noviembre, no tanto.

4. Análisis exploratorio (segunda parte)

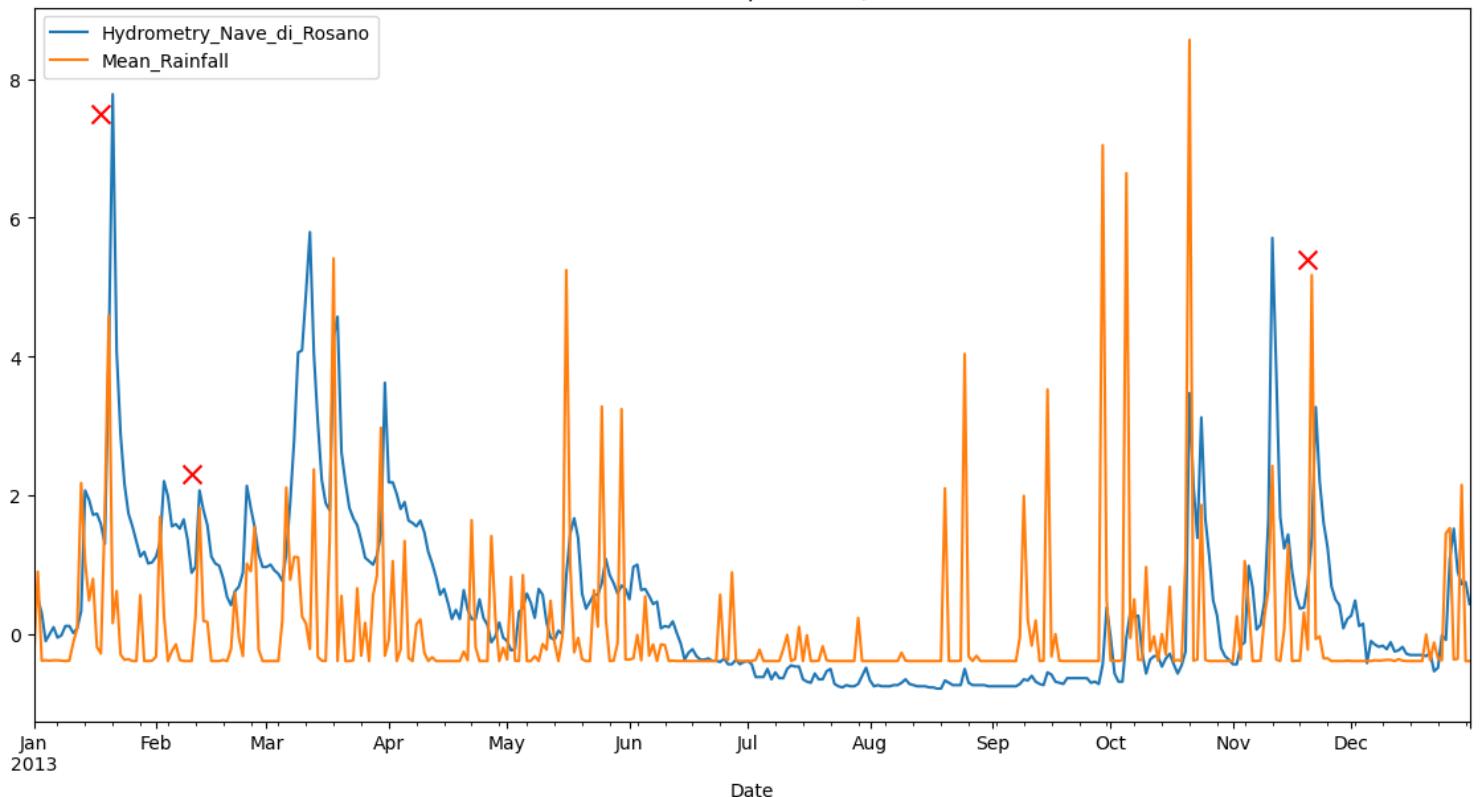
En el capítulo anterior, se ha realizado un análisis teniendo en cuenta todo el rango de datos de nuestra serie temporal, y nos hemos centrado en captar las tendencias más generales. Para poder captar relaciones más sutiles entre las variables, es necesario bajar más en el nivel de detalle, y eso es precisamente lo que haremos en este capítulo.

Como ya hemos aprendido, la evolución de las variables de nuestro dataset sigue un patrón anual. Vamos a aprovechar esta circunstancia para estudiar un año en concreto (2013), probablemente los patrones que detectemos en un año en particular, sean extrapolables a la totalidad del dataset.

Sabemos que las precipitaciones tienen gran influencia en las subidas de nivel del río. Vamos a representar las dos series temporales solapadas.

NOTA: en este caso, al no haber agrupación previa, se considera que una normalización min-max no es una buena opción (los valores extremos pueden desvirtuar los resultados). Esta vez, se opta por una normalización estándar (Wikipedia, Feature Scaling, 2022).

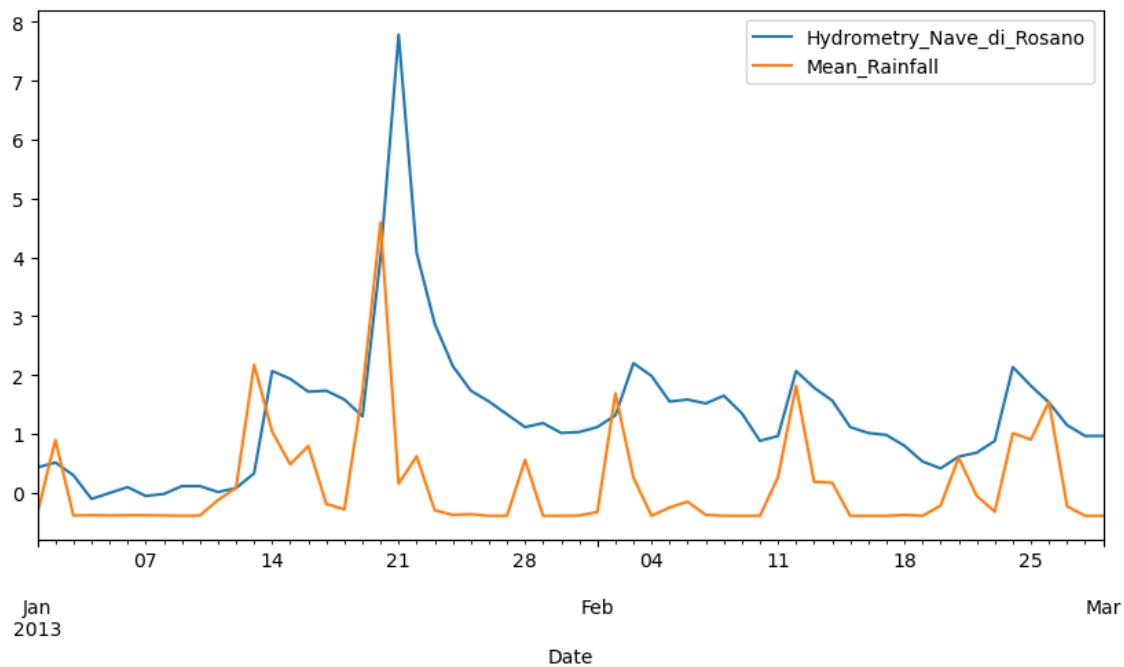
Hidrometría vs Precipitaciones, año 2013



Aunque ese patrón se repite a lo largo de todo el año, fijémonos en los tres momentos marcados con las cruces rojas. En todas ellas comienza una subida en los niveles del río (pico en la línea azul), que coincide casi simultáneamente con un pico en las precipitaciones (línea naranja). Sin embargo, vemos que la línea naranja cae rápidamente, mientras que la azul cae de una forma mucho más amortiguada.

En otras palabras, parece que las lluvias en un determinado momento afectan a los niveles del río durante varios días después de que hayan ocurrido.

Bajamos aún más al detalle (Enero-Febrero de 2013).



A simple vista, en este intervalo temporal escogido, parece que las lluvias tardan aproximadamente un día en hacer efecto en los niveles del río, y dicho efecto se hace notar durante varios días, cada vez de forma más amortiguada.

Dicho de otra forma, para predecir el nivel del río en un día en concreto, no vale con conocer lo que ha llovido ese mismo día. Es aún más importante conocer cuánto ha llovido los días anteriores.

Para poder estudiar esta relación más exhaustivamente, y de forma cuantitativa, vamos a hacer una suposición muy sencilla:

Los niveles del río se pueden predecir, conociendo simplemente lo que ha llovido ese mismo día, y lo que ha llovido los días anteriores.

4.1 Influencia de las Precipitaciones - Modelo de regresión lineal

Para poner a prueba esta suposición, vamos a entrenar un modelo de regresión lineal utilizando datos de 2013. La variable objetivo (target), será la Hidrometría de un día determinado. Las variables predictoras, serán las Precipitaciones recogidas durante el mismo día, y durante los días anteriores. Una vez entrenado el modelo, realizaremos una predicción de los niveles del río en el año siguiente (2014), lo compararemos con los valores reales, y extraeremos conclusiones.

El dataset que tenemos como punto de partida es el siguiente. Para cada día del año 2013, vemos que tenemos la Hidrometría, y las Precipitaciones que se recogieron ese mismo día:

	Hydrometry_Nave_di_Rosano	Mean_Rainfall
Date		
2013-01-01	1.75	0.28
2013-01-02	1.80	9.44
2013-01-03	1.67	0.04
2013-01-04	1.43	0.08
2013-01-05	1.49	0.04
...
2013-12-27	2.40	0.16
2013-12-28	2.03	0.24
2013-12-29	1.92	18.60
2013-12-30	1.94	0.00
2013-12-31	1.75	0.00

365 rows × 2 columns

Pero, como hemos comentado, tenemos que considerar como variables predictoras no sólo las Precipitaciones que han ocurrido ese mismo día, sino también las Precipitaciones que han ocurrido los días anteriores. Por tanto, generamos un dataset en el que, para cada observación de Hidrometría en un día concreto, se añada las lluvias que han ocurrido los 14 días anteriores.

```

for lag in range(1,15):
    df_rl['Rainfall_lag'+str(lag)] = df_rl['Mean_Rainfall'].shift(lag)

df_rl.dropna(inplace=True)

```

df_r1

El dataset sobre el que realizaremos el modelado es:

Date	Hydrometry_Nave_dl_Rosano	Mean_Rainfall	Rainfall_lag1	Rainfall_lag2	Rainfall_lag3	Rainfall_lag4	Rainfall_lag5	Rainfall_lag6	Rainfall_lag7	Rainfall_lag8	Rainfall_lag9	Rainfall_lag10	Rainfall_lag11	Rainfall_lag12	Rainfall_lag13	Rainfall_lag14
2013-01-15	2.65	6.40	10.48	18.80	3.48	1.96	0.04	0.00	0.04	0.08	0.08	0.04	0.08	0.04	9.44	0.28
2013-01-16	2.52	8.68	6.40	10.48	18.80	3.48	1.96	0.04	0.00	0.04	0.08	0.08	0.04	0.08	0.04	9.44
2013-01-17	2.53	1.48	8.68	6.40	10.48	18.80	3.48	1.96	0.04	0.00	0.04	0.08	0.08	0.04	0.08	0.04
2013-01-18	2.44	0.80	1.48	8.68	6.40	10.48	18.80	3.48	1.96	0.04	0.00	0.04	0.08	0.08	0.04	0.08
2013-01-19	2.27	15.48	0.80	1.48	8.68	6.40	10.48	18.80	3.48	1.96	0.04	0.00	0.04	0.08	0.08	0.04
...	
2013-12-27	2.40	0.16	14.04	13.48	0.08	0.28	1.96	0.08	2.80	0.00	0.00	0.00	0.00	0.00	0.04	0.20
2013-12-28	2.03	0.24	0.16	14.04	13.48	0.08	0.28	1.96	0.08	2.80	0.00	0.00	0.00	0.00	0.00	0.04
2013-12-29	1.92	18.60	0.24	0.16	14.04	13.48	0.08	0.28	1.96	0.08	2.80	0.00	0.00	0.00	0.00	0.00
2013-12-30	1.94	0.00	18.60	0.24	0.16	14.04	13.48	0.08	0.28	1.96	0.08	2.80	0.00	0.00	0.00	0.00
2013-12-31	1.75	0.00	0.00	18.60	0.24	0.16	14.04	13.48	0.08	0.28	1.96	0.08	2.80	0.00	0.00	0.00

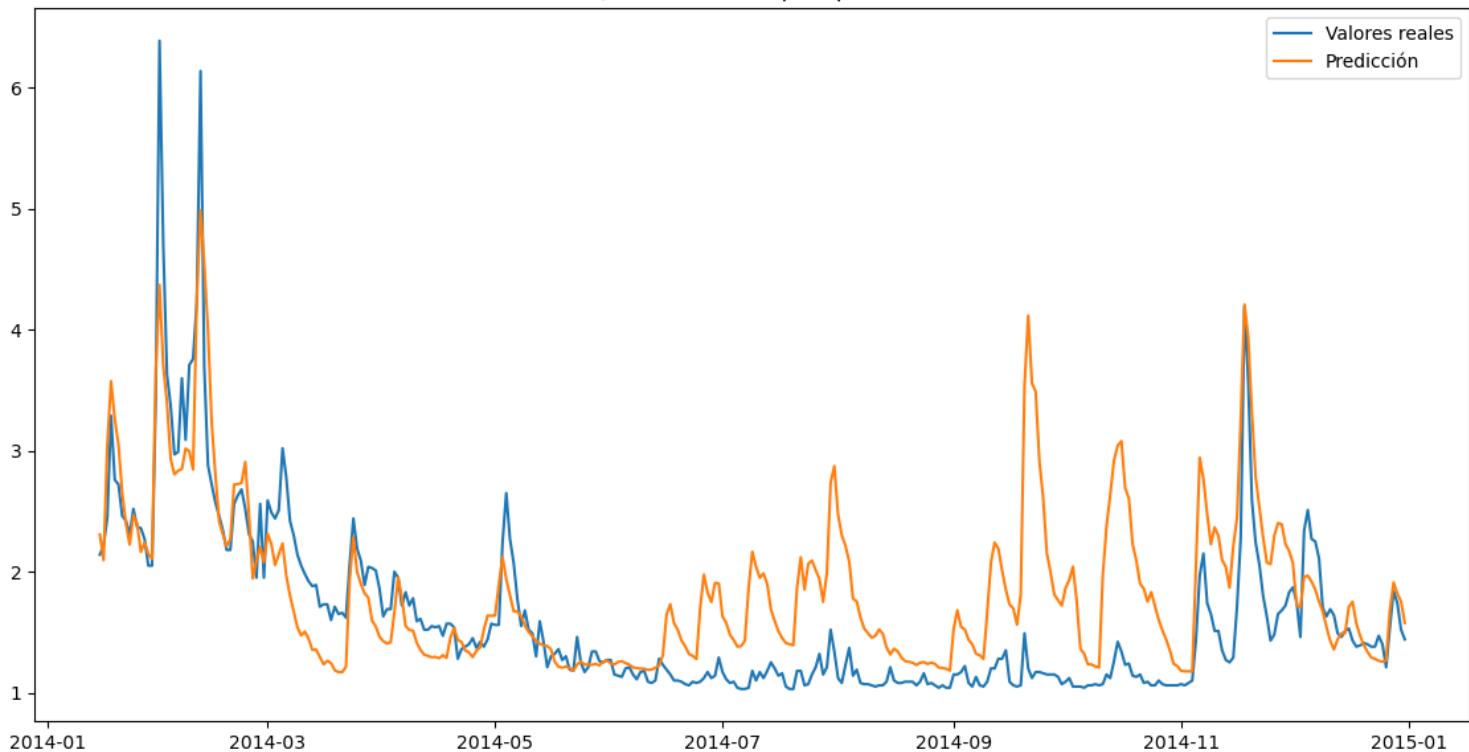
351 rows x 16 columns

Por ejemplo, para el día 2013-01-19, la Hidrometría observada fue de 2,27 m., las Precipitaciones recogidas ese mismo día fueron 15,48 mm/m², y las Precipitaciones recogidas 3 días antes fueron 8,68 mm/m².

Entrenamos un modelo de regresión lineal con estos datos de 2013, e intentamos predecir la Hidrometría de todos los días de 2014 utilizando estas mismas variables predictoras que hemos utilizado para entrenar el modelo: las Precipitaciones recogidas ese mismo día y los días anteriores.

La siguiente visualización solapa los valores reales de Hidrometría para todos los días de 2014, frente a las predicciones que realiza nuestro modelo de regresión lineal:

Predicción Hidrometría, utilizando sólo precipitaciones de días anteriores



Vemos cómo las Precipitaciones de días anteriores tienen gran capacidad de predicción sobre la Hidrometría. El modelo tiene gran capacidad para predecir las subidas y bajadas de los niveles del río. En otras palabras, la *forma* de la línea azul y la línea naranja de los gráficos anteriores es muy similar.

Ahora bien, como ya habíamos adelantado en el análisis de tendencias generales, vemos cómo las precipitaciones no influyen de la misma forma en todos los meses del año. El modelo predice de más en los meses de verano (porque las lluvias en esa época, en realidad, tienen menor capacidad de influencia en los niveles del río de la que el modelo está suponiendo), y predice de menos en invierno.

En cuanto a la capacidad de influencia que tienen las Precipitaciones de un día en concreto en la Hidrometría, según van pasando los días: la podemos averiguar extrayendo los coeficientes de nuestro modelo de regresión lineal. Al estar todas las variables predictoras (Precipitaciones ocurridas x días antes) en la misma escala, cuanto mayor sea el coeficiente asignado a un día x en el modelo, mayor peso de le está dando en la predicción a lo que haya llovido ese día.

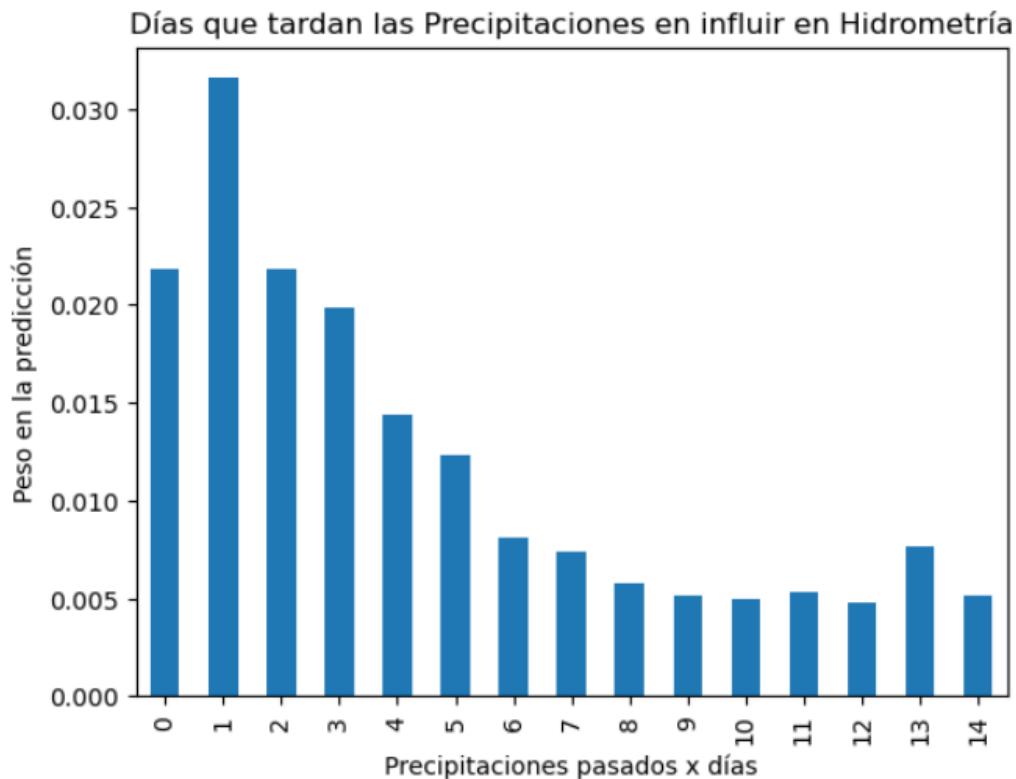
Extraemos los coeficientes del atributo “coef_”, de la instancia “lineal” de la clase “sklearn.linear_model.LinearRegression” utilizada para entrenar el modelo:

```

ax = pd.DataFrame(lineal.coef_).plot(kind='bar')
plt.xlabel('Precipitaciones pasados x días')
plt.ylabel('Peso en la predicción')
ax.get_legend().remove()
plt.title('Días que tardan las Precipitaciones en influir en Hidrometría')

plt.show()

```



Vemos que cuando más efecto hacen las lluvias en los niveles del río es un día después de que ocurran. La influencia va decayendo progresivamente a medida que pasan los días. Para proseguir con nuestro modelado, consideraremos únicamente las Precipitaciones ocurridas hasta 7 días antes, ya que, como podemos ver en el gráfico anterior, es a partir de esa cantidad de días que la influencia comienza a estabilizarse, y ser residual.

CONCLUSIONES

Las Precipitaciones tienen gran peso en la predicción de los niveles del río. Además, hemos podido cuantificar cuántos días previos hace falta tener en cuenta, y la influencia de cada uno de ellos. Pero falta por encontrar un factor que nos ayude a comprender la variación en la influencia de las Precipitaciones sobre la Hidrometría en diferentes épocas del año.

4.2 Búsqueda de variables predictoras adicionales - Modelo de Random Forest

En este capítulo, vamos a proponer varias variables que pueden ser candidatas para explicar la variación referida en las conclusiones del capítulo anterior. Es decir, nuestro objetivo ahora es encontrar una variable que ayude al modelo a “comprender” que las Precipitaciones no influyen en la misma medida en diferentes épocas del año.

Las tres variables que consideramos como candidatas son las siguientes:

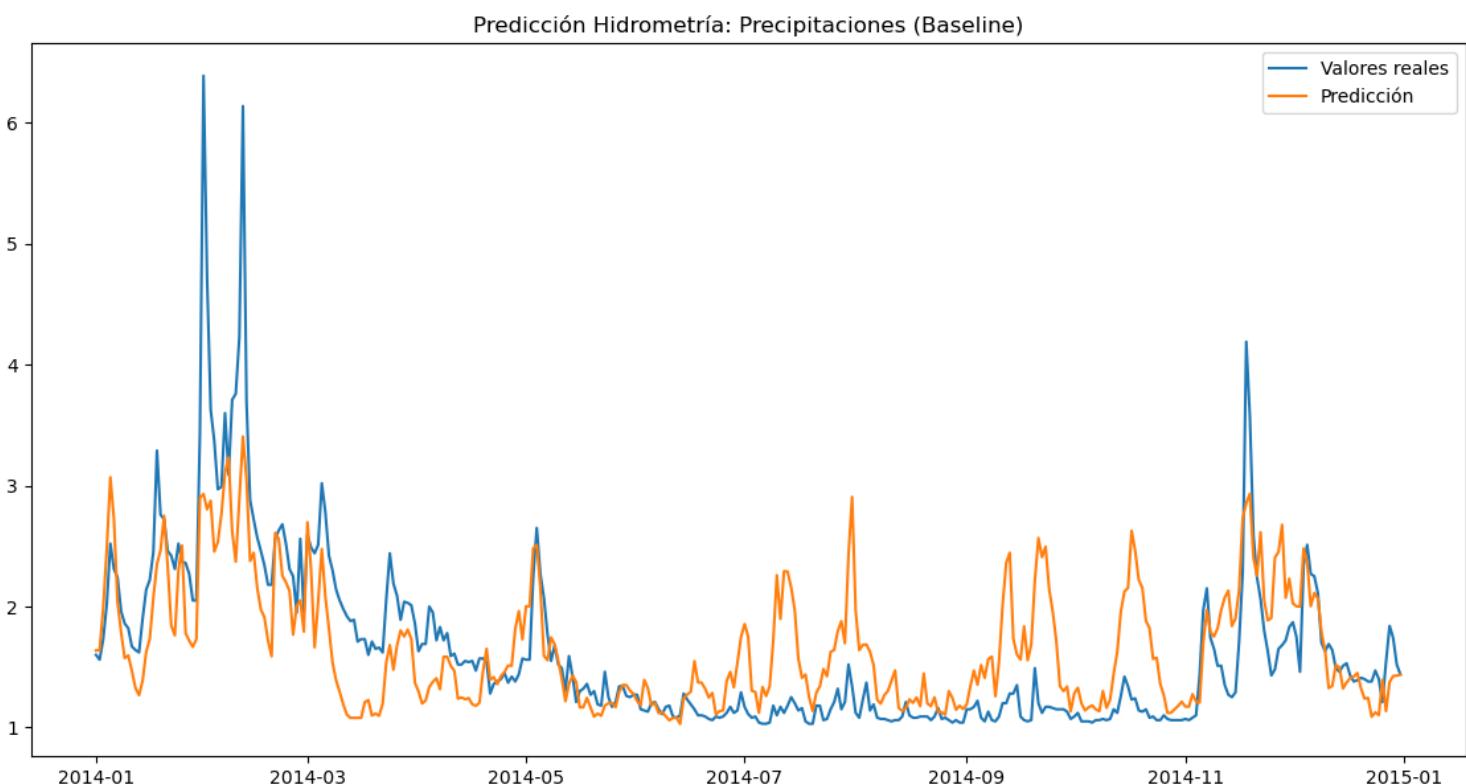
- En primer lugar tenemos la "Temperatura": hemos visto anteriormente que en los meses de mayor Temperatura, menor es la influencia de las Precipitaciones en la Hidrometría.
- En segundo lugar, podemos suponer que la capacidad de drenado del suelo está influida por lo que haya llovido en meses anteriores. Así que probaremos "Precipitaciones acumuladas" (últimos 90 días).
- En tercer lugar, tenemos el "Momento del año": es posible que la variación en la influencia de las precipitaciones dependa fuertemente del momento del año que se esté observando.
 - NOTA: explicación detallada del método que se utiliza para obtener el "Momento del año" como una función continua en Anexo 3: Obtención de la variable "Momento del año".

La metodología que se utilizará para determinar cuáles de estas variables pueden ayudarnos en nuestra predicción, es la siguiente:

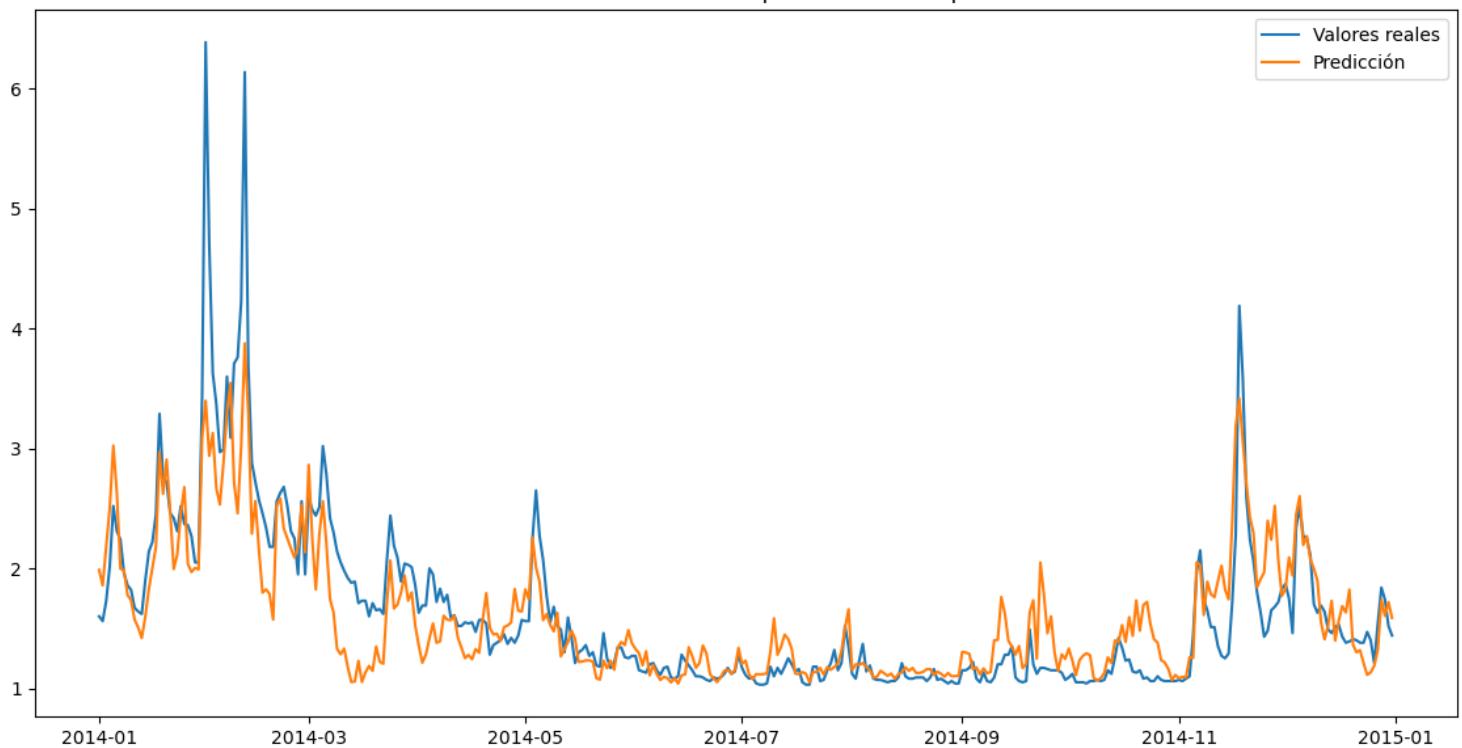
- Utilizamos un modelo de Random Forest en lugar de regresión lineal, ya que puede ser que alguna de las candidatas tenga una relación no lineal con la target
- Utilizaremos como datos de entrenamiento tres años (2011, 2012, 2013)
- Establecemos una predicción “baseline” de referencia, que será la predicción de un modelo de Random Forest utilizando como variables predictoras las Precipitaciones (las recogidas ese mismo día y los 7 días anteriores)
- Para cada una de estas tres variables que ensayaremos (Temperatura, Precipitaciones acumuladas, Momento del año)
 - Se predecirá Hidrometría para el año 2014, utilizando como variables predictoras las Precipitaciones de días anteriores (es decir, las que hemos usado en nuestra predicción “baseline”), más una de estas variables candidatas

- Visualizamos predicción vs realidad, obtenemos métrica: error absoluto medio ("mae")
- Comparamos los resultados obtenidos utilizando estas tres variables predictoras, extraemos conclusiones acerca de cuáles pueden ser de utilidad

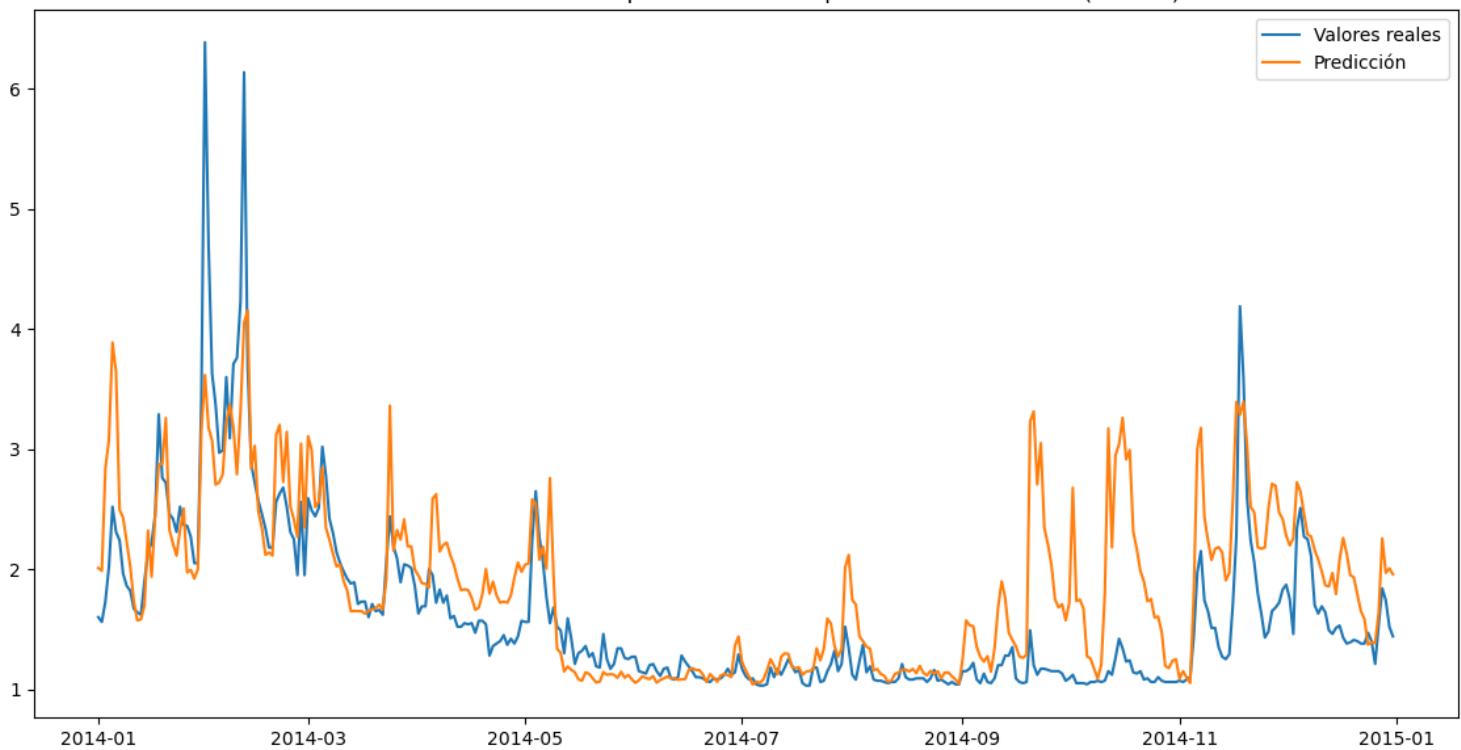
Una vez implementada esta metodología, estos son los resultados, visuales y cuantitativos, que se han obtenido:

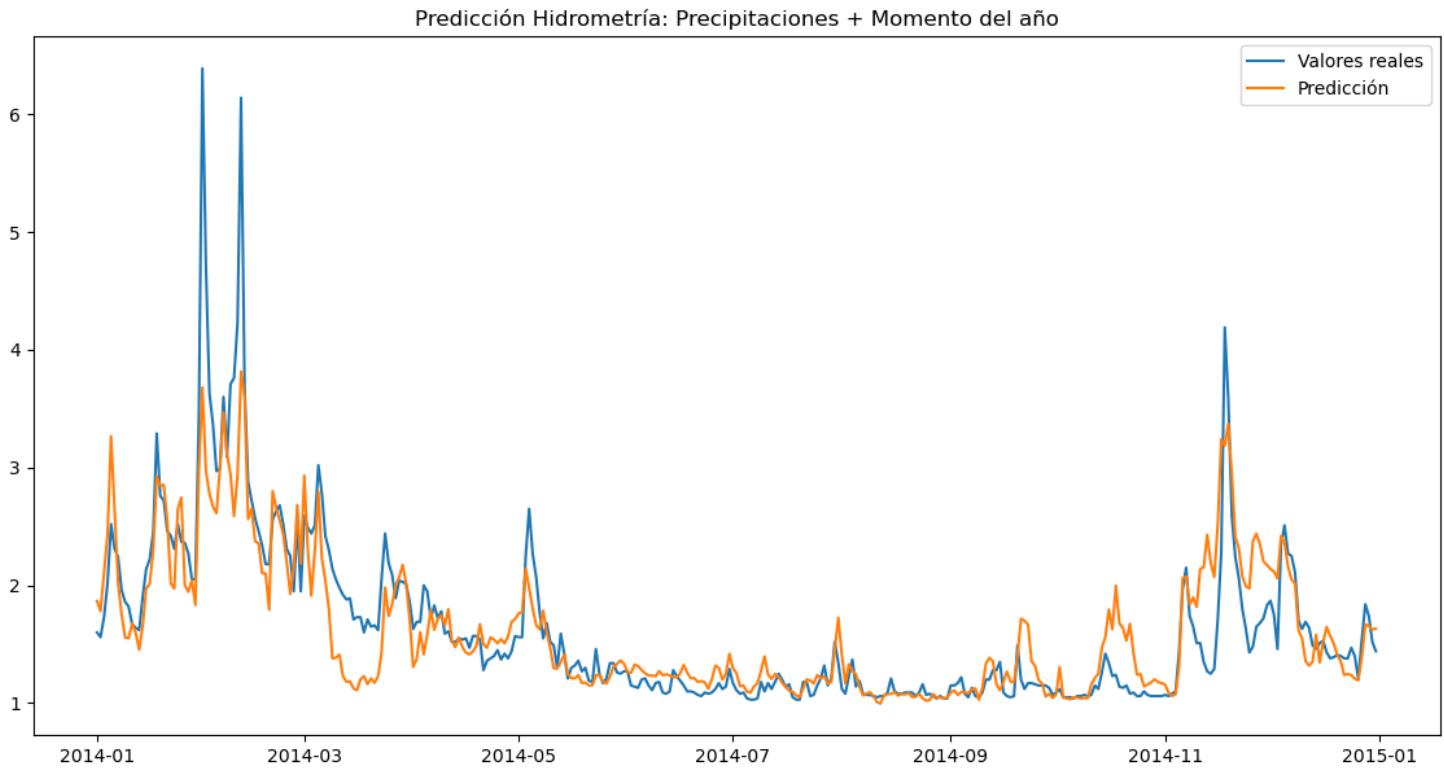


Predicción Hidrometría: Precipitaciones + Temperatura



Predicción Hidrometría: Precipitaciones + Precipitaciones acumuladas (90 días)





Cuantitativamente, estos son los errores absolutos medios de cada una de las cuatro predicciones realizadas:

metricas

```
{'Baseline': 0.38,
'Temperatura': 0.24,
'Precipitaciones acumuladas': 0.36,
'Momento del año': 0.22}
```

Vemos que la variable de ensayo "Precipitaciones acumuladas" es la que parece menos prometedora, apenas consigue mejorar la métrica que tenemos para la Baseline. Respecto a las otras dos variables ("Temperatura", "Momento del Año"), vemos que sí que consiguen bajar el error absoluto medio de la predicción.

En las gráficas observamos cómo, utilizando estas dos variables, se consigue amortiguar en cierta medida los errores en las predicciones en los que incurría el modelo Baseline, especialmente en verano. Ahora bien, estos modelos siguen teniendo margen de mejora: predicen de menos cuando ocurren grandes subidas del nivel del río, y hay determinados tramos del año donde las predicciones se desvían bastante.

CONCLUSIONES

"Temperatura" y "Momento del Año" parecen variables que ayudan a mejorar las predicciones, "Precipitaciones acumuladas" no.

Decidimos incluir en el modelado las dos variables que mejoran las predicciones:
"Temperatura" y "Momento del año".

Llegados a este punto, se adivinan dos enfoques para intentar mejorar las predicciones que tenemos ahora mismo

- Enfoque A: continuar las indagaciones para encontrar nuevas variables predictoras, que ayuden a afinar las predicciones del modelo. Implícitamente, estamos suponiendo que, con las variables que tenemos hasta el momento, no tenemos información suficiente para realizar una buena predicción de la Hidrometría.
- Enfoque B: explorar otros modelos que se adapten mejor a la resolución de este tipo de problema. Implícitamente, estamos suponiendo que, con las variables que tenemos hasta el momento (Precipitaciones, Temperatura, Momento del año) es suficiente para obtener una buena predicción de la Hidrometría. Falta encontrar un modelo que capte correctamente las relaciones entre dichas variables.

El enfoque que consideramos que es más acertado es el Enfoque B. En el siguiente capítulo, se buscará un modelo apropiado a la resolución de este problema.

5. Modelado

5.1 Consideraciones previas

Una vez concluido el análisis exploratorio de datos, llega el momento de utilizar un modelo para obtener predicciones de la Hidrometría. En este sentido, hay dos decisiones que son clave:

- Elección del horizonte temporal de la predicción: ¿Para qué necesita una compañía de aguas conocer con antelación los niveles de un río? ¿Qué uso práctico se le pretende dar a esas predicciones? La respuesta a estas preguntas condicionará los horizontes temporales (días, semanas, meses) que escojamos para realizar nuestras predicciones.
- Elección del modelo: Con todo lo que hemos aprendido de los datos en capítulos previos, ¿qué tipo de modelo puede ser el apropiado para obtener predicciones precisas?

5.1.1 Horizonte temporal de la predicción

Rescatamos una frase textual que escribe la propia compañía gestora de aguas que subió este [desafío a la web](#):

- *The time interval is defined as day/month depending on the available measures for each waterbody. Models should capture volumes for each waterbody (for instance, for a model working on a monthly interval a forecast over the month is expected).*

En nuestro caso, el intervalo de tiempo entre cada observación es un día, así que uno de los horizontes temporales que vamos a predecir, son los niveles del río al día siguiente.

Pero además, aunque no lo especifique la compañía en el desafío, vamos a realizar predicciones para un horizonte temporal más largo. Veremos que este tipo de predicciones plantean un desafío adicional, y esto ayudará a que podamos ahondar en nuestra comprensión sobre esta serie temporal.

Por tanto, elegimos un segundo horizonte temporal que sean dos semanas. Muchas predicciones meteorológicas suelen tener este horizonte temporal, y necesitaremos alimentar al modelo con las predicciones de lluvia y temperatura para poder hacer predicciones a largo plazo. De todas formas, como se verá más adelante, la metodología que se utilizará para hacer predicciones a dos semanas vista, se puede replicar

perfectamente para un horizonte temporal más largo o más corto. Bastará con disponer de predicciones meteorológicas para el horizonte temporal que escojamos.

5.1.2 Modelo utilizado

En el capítulo de análisis exploratorio de datos, ya hemos utilizado un modelo de regresión lineal, y un modelo de random forest. Para realizar el modelado final, vamos a escoger un modelo más adaptado a series temporales, y que debería de tener más potencial que los anteriores: redes neuronales recurrentes LSTM.

5.2 Modelado con Redes Neuronales Recurrentes LSTM

Para realizar las predicciones de nuestra serie temporal, se implementará un algoritmo de redes neuronales recurrentes LSTM desarrollado por Tensorflow (Website de Tensorflow. Time series forecasting tutorial, 2022). El código utilizado es una adaptación para nuestro caso particular del ejemplo que se utiliza como guía-tutorial en la propia página web de Tensorflow para predecir series temporales con redes neuronales.

https://www.tensorflow.org/tutorials/structured_data/time_series

5.2.1 Preprocesado de datos

Como ya se había adelantado en el capítulo de análisis exploratorio, se van a utilizar como variables predictoras: Precipitaciones, Temperatura, y Momento del año (Señal seno y coseno de frecuencia anual). Además, la elección de este modelo nos condiciona a tomar otras dos decisiones:

- También se va a incluir la propia Hidrometría como variable predictora de ella misma.
- No sólo se van a utilizar los valores de varios días anteriores (siete) de las Precipitaciones. Se van a utilizar los valores de siete días anteriores de todas las variables predictoras.

El código utilizado en el tutorial de Tensorflow, está adaptado para tomar como inputs tanto los valores de la propia variable target, como los valores de días anteriores de todas las variables predictoras. Se podría hacer un análisis muy minucioso para intentar modificar con más profundidad el código, y poder ensayar sólo con las variables que más interés apuntaban en el capítulo de análisis exploratorio. Pero se considera que ese análisis se escapa del alcance de este proyecto. Además, tanto la inclusión de la Hidrometría como

de los valores de días anteriores de todas las variables, es improbable que tengan una influencia negativa en los resultados (quizá, todo lo contrario).

Tomamos como punto de partida el dataframe completo, ya no vamos a seleccionar años en concreto como en los capítulos anteriores.

Date	Temperature_Firenze	Hydrometry_Nave_di_Rosano	Mean_Rainfall
2004-01-01	8.65	1.84	0.08
2004-01-02	7.10	1.93	0.00
2004-01-03	5.50	1.61	0.00
2004-01-04	3.55	1.35	0.00
2004-01-05	2.70	1.44	0.04
...
2020-06-26	25.00	1.34	0.00
2020-06-27	26.00	1.21	0.00
2020-06-28	26.00	1.30	0.00
2020-06-29	25.00	1.19	0.00
2020-06-30	26.00	1.30	0.00

6026 rows × 3 columns

Incluimos en el dataframe el Momento del Año (utilizando las señales Seno y Coseno, tal como se explica en el Anexo 3: Obtención de la variable “Momento del año”). A continuación, dividimos los datos en los conjuntos de train, validación y test. Se hace por intervalos temporales consecutivos y en orden cronológico, las divisiones elegidas son las siguientes:

- Conjunto de Train: años 2004 a 20016
- Conjunto de Validación: años 2017 a 2018
- Conjunto de Test: año 2019 hasta final de la serie (2020-06-30)

Además, para realizar el entrenamiento de la red, se utilizarán datos normalizados para todos los conjuntos de datos (se opta nuevamente por una normalización estándar).

Hechas todas estas transformaciones, los dataframe resultantes tienen esta estructura:

```
train_df.head()
```

Date	Temperature_Firenze	Hydrometry_Nave_di_Rosano	Mean_Rainfall	Year_sin	Year_cos
2004-01-01	-1.056027	0.580601	-0.379358	3.310437e-07	1.413686
2004-01-02	-1.265387	0.730969	-0.390321	2.432714e-02	1.413477
2004-01-03	-1.481502	0.196326	-0.390321	4.864675e-02	1.412849
2004-01-04	-1.744891	-0.238072	-0.390321	7.295197e-02	1.411804
2004-01-05	-1.859701	-0.087703	-0.384839	9.723559e-02	1.410340

5.2.2 Clases y funciones

Se declaran todas las clases y funciones necesarias para llevar a cabo el entrenamiento y predicción con un modelo de Redes Neuronales Recurrentes LSTM (ver sección homónima del notebook para detalles acerca del código empleado).

5.2.3 Entrenamiento del modelo

Definimos la ventana de predicción que vamos a usar: utilizando los datos de los últimos 7 días, predecir la Hidrometría del día siguiente. Se utiliza la clase “WindowGenerator” declarada anteriormente.

```
ventana_prediccion = WindowGenerator(  
    input_width=7,  
    label_width=1,  
    shift=1,  
    label_columns=['Hydrometry_Nave_di_Rosano'])  
  
ventana_prediccion  
  
Índices de los días predictores: [0 1 2 3 4 5 6]  
Índices de los días a predecir: [7]  
Variables objetivo: ['Hydrometry_Nave_di_Rosano']
```

Definimos la arquitectura de la red e instanciamos el modelo.

```
lstm_model = tf.keras.models.Sequential([  
    tf.keras.layers.LSTM(units=32),  
    tf.keras.layers.Dense(units=1)  
])
```

Compilamos el modelo y lo entrenamos con los datos del conjunto de train, pasándole como argumento esa ventana_prediccion que creamos previamente. Es decir, el modelo

será entrenado para aprender cómo ha de asignar los pesos teniendo en cuenta las variables predictoras para los últimos 7 días, de forma que se obtenga la predicción más precisa posible para la variable target al día siguiente.

```
MAX_EPOCHS = 20

early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss',
                                                 patience=2,
                                                 mode='min')

lstm_model.compile(loss=tf.keras.losses.MeanSquaredError(),
                    optimizer=tf.keras.optimizers.Adam(),
                    metrics=[tf.keras.metrics.MeanAbsoluteError()])

history = lstm_model.fit(ventana_prediccion.train, epochs=20,
                         validation_data=ventana_prediccion.val,
                         callbacks=[early_stopping])

history

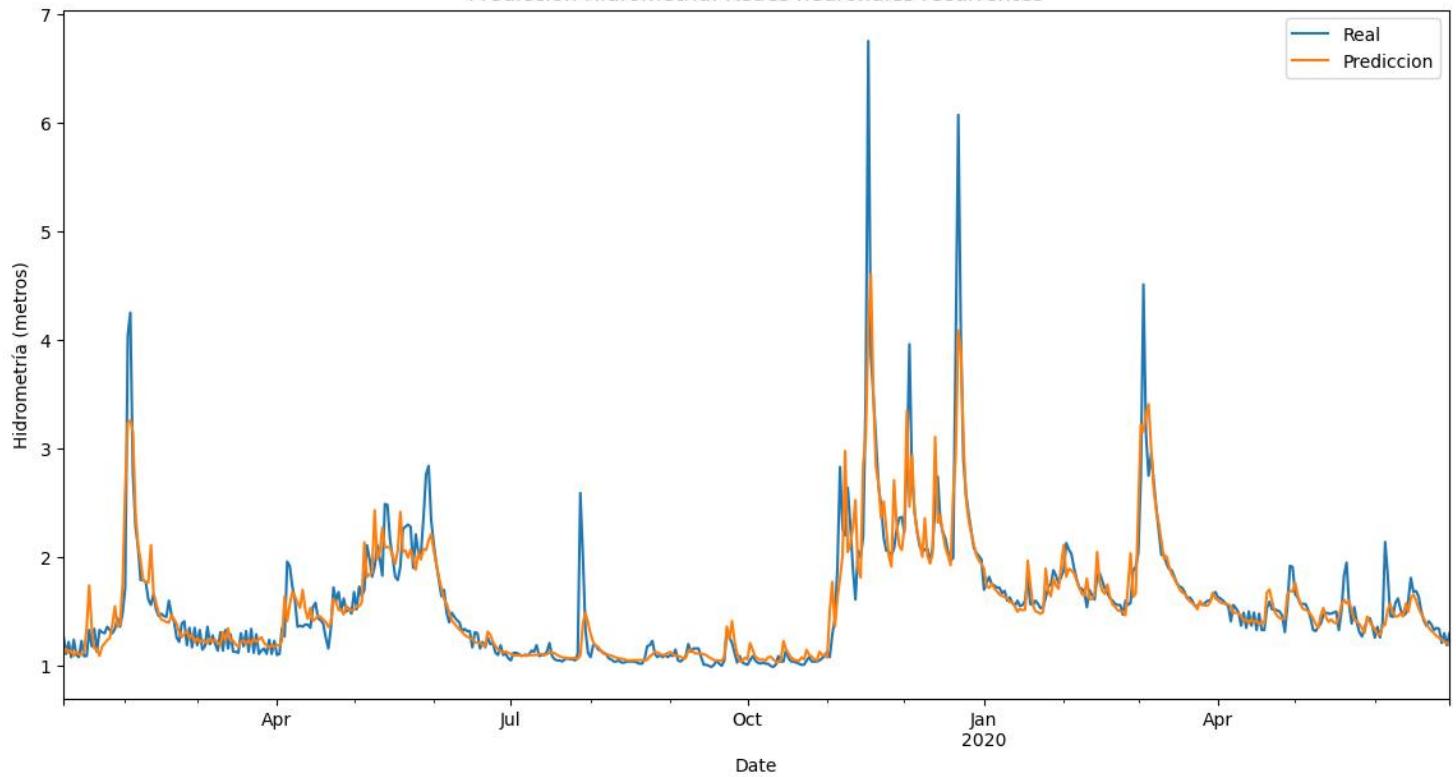
Epoch 1/20
149/149 [=====] - 5s 10ms/step - loss: 0.4022 - mean_absolute_error: 0.3560 - val_loss: 0.1972 - val_mean_absolute_error: 0.2443
Epoch 2/20
149/149 [=====] - 1s 7ms/step - loss: 0.2330 - mean_absolute_error: 0.2532 - val_loss: 0.1608 - val_mean_absolute_error: 0.2156
Epoch 3/20
149/149 [=====] - 1s 7ms/step - loss: 0.1922 - mean_absolute_error: 0.2282 - val_loss: 0.1521 - val_mean_absolute_error: 0.2074
Epoch 4/20
149/149 [=====] - 1s 7ms/step - loss: 0.1760 - mean_absolute_error: 0.2170 - val_loss: 0.1501 - val_mean_absolute_error: 0.2021
Epoch 5/20
149/149 [=====] - 1s 7ms/step - loss: 0.1673 - mean_absolute_error: 0.2104 - val_loss: 0.1498 - val_mean_absolute_error: 0.1976
Epoch 6/20
149/149 [=====] - 1s 7ms/step - loss: 0.1614 - mean_absolute_error: 0.2055 - val_loss: 0.1501 - val_mean_absolute_error: 0.1935
Epoch 7/20
149/149 [=====] - 1s 6ms/step - loss: 0.1572 - mean_absolute_error: 0.2018 - val_loss: 0.1503 - val_mean_absolute_error: 0.1897
```

Se incluye el método de EarlyStopping para evitar un sobreajuste en el entrenamiento, y el entrenamiento finaliza tras completar 7 épocas.

Una vez entrenado el modelo, se realizan predicciones para el conjunto de test (valores comprendidos entre 2019-01-01 y 2020-06-30). Se deshace la transformación (normalización estándar) que se realizó sobre los datos para devolverlos a su escala inicial.

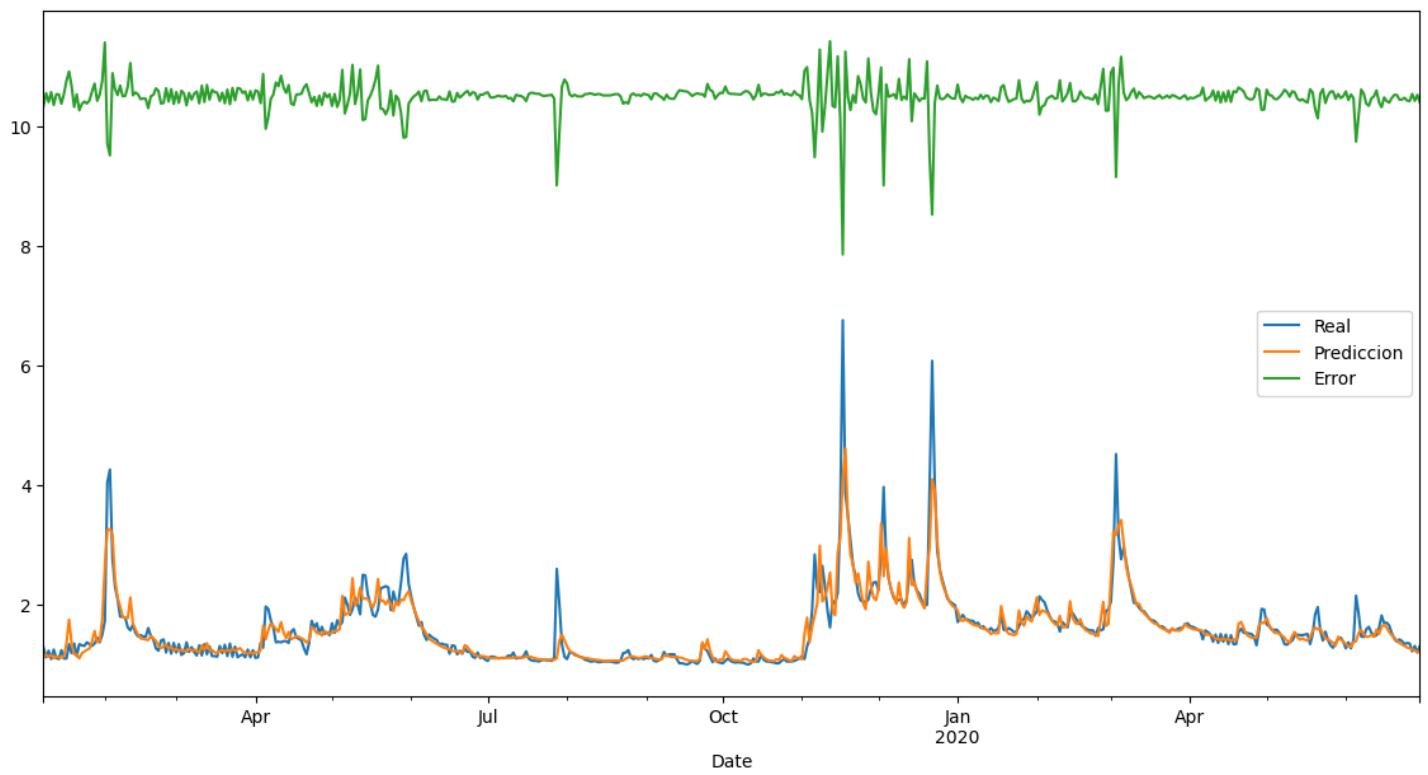
Y se comparan los valores reales de Hidrometría para el conjunto de test, frente a las predicciones que realiza nuestro modelo. Estos son los resultados obtenidos:

Predicción Hidrometría: Redes neuronales recurrentes



Visualizamos también la serie temporal de la diferencia entre valores reales y predicción.

NOTA: la escala del gráfico aplica a las series Real y Predicción, no a la serie Error (color verde). Se ha “movido” dicha serie hacia arriba para evitar un solape con las dos anteriores y facilitar la interpretación de los resultados).



Cuantitativamente, se obtiene que el error absoluto medio entre las predicciones y los valores reales es de 0,13 metros. Teniendo en cuenta que la Hidrometría media en los años 2019 y 2020 tiene valores entorno a 1,5 metros, vemos que el error relativo es bastante bajo (inferior al 10%).

Tanto a nivel visual como cuantitativo, vemos cómo las predicciones del modelo de redes neuronales recurrentes LSTM mejoran ostensiblemente a aquellas obtenidas con los modelos de regresión lineal y random forest utilizados en el anterior capítulo.

Se han cambiado diferentes parámetros con respecto a aquellos ensayos, además del modelo escogido: hemos entrenado con los datos de todo el histórico, hemos utilizado un conjunto de test diferente, hemos utilizado como variables predictoras los valores de días anteriores de todas las variables y no sólo de las Precipitaciones... Ya sea por uno o varios de estos motivos, las predicciones obtenidas han mejorado mucho.

Vemos cómo las predicciones (línea naranja) y los valores reales (línea azul) siempre son muy similares. Sea cual sea la época del año, el modelo capta correctamente la influencia de las Precipitaciones. Ya no ocurre aquello observado para regresión lineal y random

forest, que en algunos tramos del año ambas líneas se "despegaban" mucho la una de la otra.

En la segunda visualización, donde se incluye la serie temporal de los errores (línea verde, diferencia entre Real y Predicción), se ve cómo éstos siempre coinciden con los momentos en los que aumenta y disminuye bruscamente el nivel del río, independientemente de la época del año en la que esta ocurra (evidentemente, cuanto más bruscas sean las subidas y bajadas, más error en las predicciones). Los mayores errores se producen justamente los días en los que se producen grandes crecidas del río, en las cuales el modelo tiende a predecir niveles más bajos de lo real (picos hacia abajo de la línea verde).

6. Obtención de predicciones a futuro - diferentes horizontes temporales

Una vez ya se ha entrenado el modelo y se ha validado su comportamiento, el paso final consiste en realizar predicciones de valores futuros, simulando cómo una compañía gestora de aguas podría sacar provecho de este modelo. Como ya se había comentado, se ofrecerán predicciones para dos horizontes temporales diferentes: estimaremos los niveles del río para el día siguiente, y para dentro de dos semanas.

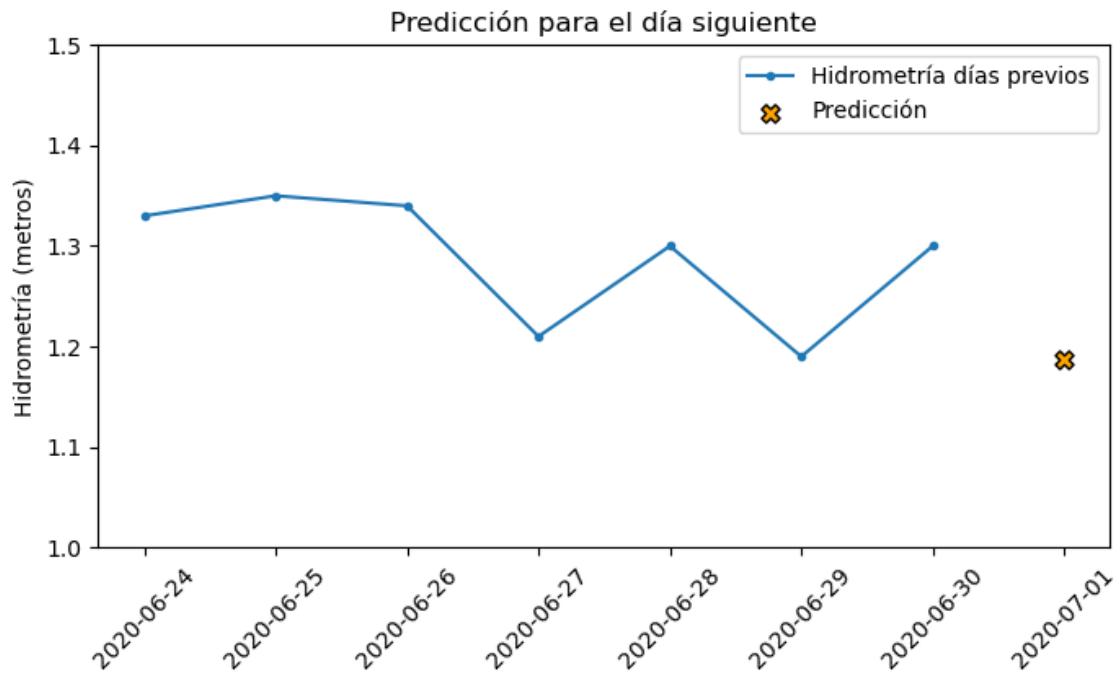
6.1 Predicción al día siguiente

Ya que nuestro conjunto de datos finaliza el 2020-06-30, se va a obtener la predicción de cómo estará el nivel del río al día siguiente: 2020-07-01.

Para ello, se utiliza una dinámica similar a la que se ha relatado en la fase de validación, en el capítulo de Entrenamiento del modelo: se genera el conjunto de variables predictoras: valores de Hidrometría, Precipitaciones, Temperatura y Momento del año, para los 7 días anteriores al día que se quiere ofrecer la predicción. Se les aplica una normalización estándar.

Temperature_Firenze	Hydrometry_Nave_di_Rosano	Mean_Rainfall	Year_sin	Year_cos	
2020-06-24	1.287461	-0.271487	-0.390321	0.181985	-1.402435
2020-06-25	1.152389	-0.238072	-0.390321	0.157833	-1.405357
2020-06-26	1.152389	-0.254779	-0.390321	0.133635	-1.407864
2020-06-27	1.287461	-0.471978	-0.390321	0.109397	-1.409954
2020-06-28	1.287461	-0.321610	-0.390321	0.085127	-1.411627
2020-06-29	1.152389	-0.505393	-0.390321	0.060832	-1.412882
2020-06-30	1.287461	-0.321610	-0.390321	0.036519	-1.413719

Se utiliza el modelo entrenado y validado en el capítulo anterior para obtener una predicción para la Hidrometría en el día 2020-07-01. Se desnormaliza la predicción, y se visualizan los resultados:



6.2 Predicción para dentro de dos semanas

NOTA: Recordemos que, aunque en este caso se realice la predicción para dentro de dos semanas, el procedimiento que se verá a continuación se puede aplicar para otros horizontes temporales más cortos o largos. Basta con obtener las predicciones de una agencia meteorológica para las Temperaturas y Precipitaciones de los días previos.

Supongamos que quisiésemos conocer la predicción de la Hidrometría dos semanas más allá de la fecha hasta la que disponemos datos: es decir, queremos una predicción para el 2020-07-14.

A diferencia de la predicción para el día siguiente, las predicciones para horizontes temporales más largos presentan una complicación adicional: a priori, no poseemos de todos los datos necesarios para poder hacer la predicción. Nuestro modelo necesita conocer los valores de Temperatura, Precipitaciones e Hidrometría de los 7 días anteriores al día que se quiere hacer la predicción. Se valoran dos enfoques diferentes para resolver este problema:

- Enfoque A: Se puede entrenar la red neuronal para que aprenda a hacer predicciones para dentro de 14 días, en vez de hacerlas para dentro de 1 día. Habría que especificar una ventana diferente a la que se ha utilizado, y rehacer el entrenamiento.

A continuación, se indica cómo se generaría una ventana para predecir la hidrometría dentro de 14 días, utilizando un histórico de 7 días de las variables predictoras.

```
ventana_alternativa = WindowGenerator(  
    input_width=7,  
    label_width=1,  
    shift=14,  
    label_columns=['Hydrometry_Nave_di_Rosano'])  
  
ventana_alternativa  
  
Índices de los días predictores: [0 1 2 3 4 5 6]  
Índices de los días a predecir: [20]  
Variables objetivo: ['Hydrometry_Nave_di_Rosano']
```

Este enfoque presenta un gran inconveniente, y es que, como vimos en el capítulo de análisis exploratorio de datos, las Precipitaciones influyen sobre el nivel del río unos pocos días después de ocurrir, pasados esos días su influencia se disipa. Para predecir cómo serán los niveles del río el día 20, no parece muy importante conocer lo que haya llovido los días 0 a 6, sería mucho más determinante conocer lo que ha llovido los días 17, 18 o 19.

Así que se utilizará un enfoque alternativo, que se considera que se adapta mucho mejor a las peculiaridades de nuestro problema:

- Enfoque B: Mantener el modelo que ya hemos entrenado: predecir para el día siguiente, utilizando los datos de los 7 días anteriores. Reutilizar esta predicción de la Hidrometría del día siguiente, más las predicciones de una agencia meteorológica para Temperaturas y Precipitaciones, para obtener una predicción de la Hidrometría para dentro de dos días. Realizar sucesivas iteraciones hasta obtener la predicción para dentro de 14 días.

Este enfoque presenta una gran ventaja, y es que así sí que se están utilizando las Precipitaciones de los días inmediatamente anteriores como variables predictoras. Además, para los valores futuros de Precipitaciones y Temperatura, no hace falta utilizar las predicciones de este mismo modelo, el cual no está específicamente diseñado para predecirlas. En su lugar, utilizamos las predicciones de una agencia meteorológica, mucho más adaptadas para predecir correctamente Precipitaciones y Temperatura.

Inicialmente, para poner este enfoque en práctica, se consideró ofrecer la predicción para el 2020-07-14. Ya que, como hemos indicado, es la fecha que coincide con dos semanas después del final de los datos de nuestro dataset. Sin embargo, en esa época del año las precipitaciones son escasas y los niveles del río son estables y fácilmente predecibles. Así que se va a realizar la predicción para una época del año más lluviosa (en concreto, la época en la que se redacta este documento, otoño de 2022).

Suponemos que estamos a fecha de 16 de Noviembre de 2022. La compañía gestora necesita una predicción de la Hidrometría para dentro de 14 días, es decir el 30 de Noviembre de 2022. Para ello, vamos a necesitar los siguientes datos externos:

- Valores reales de fechas previas: Necesitamos conocer Hidrometría, Precipitaciones y Temperaturas de los 7 días anteriores al momento actual. Es decir, el intervalo del 10 al 16 de Noviembre
- Valores estimados para Precipitaciones y Temperatura desde el momento actual hasta el día anterior al que queremos predecir. Es decir, el intervalo del 17 al 29 de Noviembre

Cargamos estos datos que hemos almacenado en un fichero .csv

NOTA: Los valores estimados de Precipitaciones y Temperatura (del 2022-11-17 al 2022-11-29) han sido obtenidos de la siguiente página web:

<https://www.ilmeteo.it/meteo/Firenze>

Para los valores de Precipitaciones y Temperatura de fechas previas (2022-11-10 a 2022-11-16), no se disponía de información en esa página web, y no se encontró ninguna otra página web que ofreciese tanto Precipitaciones y Temperatura de fechas previas, como de fechas futuras. Para la Hidrometría en Nave di Rosano se encontró esta web:

<https://www.cfr.toscana.it/monitoraggio/stazioni.php?type=idro>

Pero hay dos puntos de medición para este lugar, y desconocemos cuál de los dos es el que coincide con el que tenemos en nuestro dataset.

Así que, para no complicar esta parte en demasía, se decide generar datos dummy para Hidrometría, Temperatura y Precipitaciones de fechas previas. Simplemente se replican los valores que tuvimos en ese mismo intervalo en un año del que sí disponemos datos: del 2019-11-10 a 2019-11-16.

Así, el .csv que se carga en el entorno es el siguiente:

Date	Temperature_Firenze	Hydrometry_Nave_di_Rosano	Mean_Rainfall
2022-11-10	9.0	2.30	8.00
2022-11-11	12.0	1.90	19.04
2022-11-12	12.0	1.61	5.28
2022-11-13	10.0	2.06	0.20
2022-11-14	11.0	2.00	34.00
2022-11-15	12.0	2.19	30.88
2022-11-16	11.0	3.47	59.48
2022-11-17	11.0	NaN	25.44
2022-11-18	15.2	NaN	4.10
2022-11-19	11.4	NaN	6.10
2022-11-20	10.6	NaN	0.50
2022-11-21	8.3	NaN	3.00
2022-11-22	9.7	NaN	47.00
2022-11-23	7.6	NaN	0.50
2022-11-24	9.2	NaN	0.00
2022-11-25	9.3	NaN	2.00
2022-11-26	11.9	NaN	2.00
2022-11-27	9.8	NaN	0.00
2022-11-28	5.6	NaN	0.00
2022-11-29	8.9	NaN	0.50
2022-11-30	8.0	NaN	0.00

Generamos la variable Momento del año y normalizamos los datos.

```
df_2022.head(10)
```

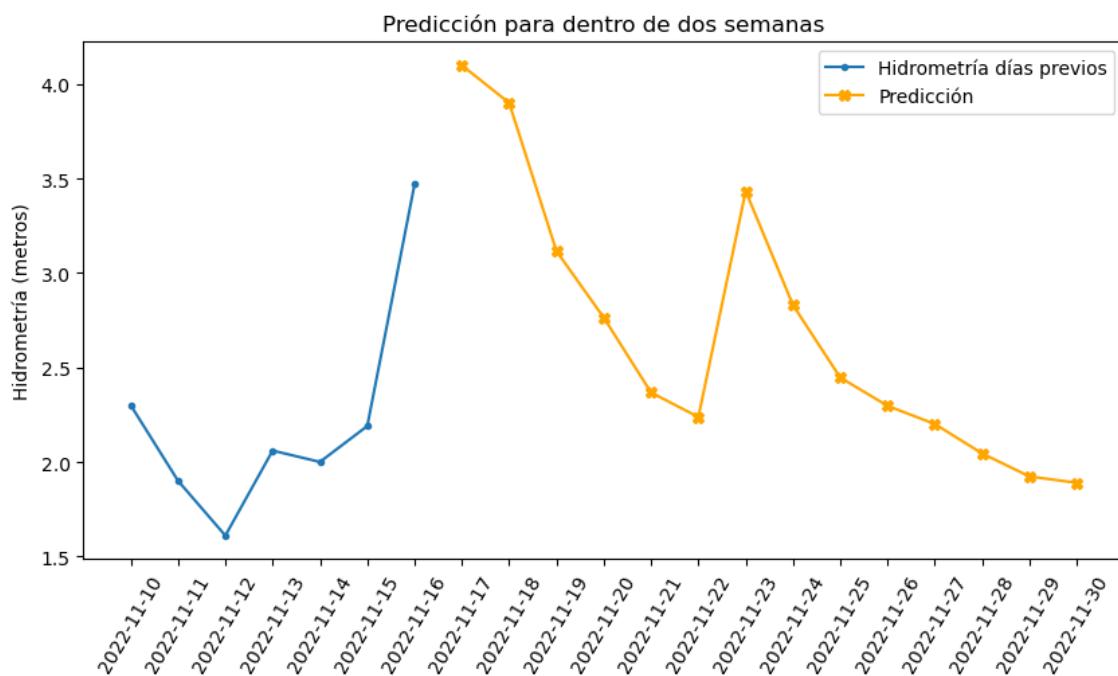
Date	Temperature_Firenze	Hydrometry_Nave_di_Rosano	Mean_Rainfall	Year_sin	Year_cos
2022-11-10	-1.008752	1.349150	0.706064	-1.101132	0.886970
2022-11-11	-0.603538	0.680846	2.219075	-1.085704	0.905777
2022-11-12	-0.603538	0.196326	0.333293	-1.069955	0.924316
2022-11-13	-0.873681	0.948168	-0.362912	-1.053890	0.942581
2022-11-14	-0.738609	0.847922	4.269315	-1.037512	0.960567
2022-11-15	-0.603538	1.165367	3.841725	-1.020828	0.978269
2022-11-16	-0.738609	3.303939	7.761302	-1.003841	0.995681
2022-11-17	-0.738609	NaN	3.096183	-0.986558	1.012799
2022-11-18	-0.171310	NaN	0.171576	-0.968982	1.029617
2022-11-19	-0.684581	NaN	0.445672	-0.951120	1.046130

Utilizamos el modelo para obtener la predicción de la Hidrometría para 2022-11-17, usando como variables predictoras aquellas para las fechas entre 2022-11-10 y 2022-11-16.

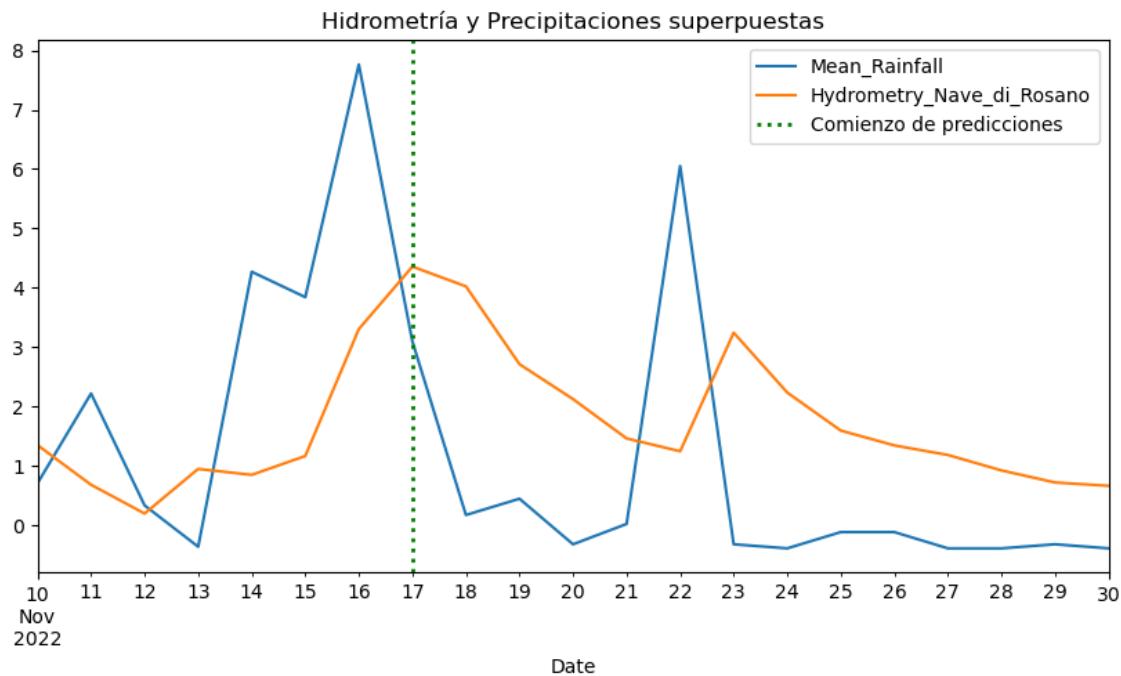
En la siguiente iteración, predecimos la Hidrometría para 2022-11-18, usando como variables predictoras aquellas para las fechas entre 2022-11-11 y 2022-11-17.

Continuamos iterando hasta alcanzar la predicción objetivo: Hidrometría de 2022-11-30.

Desnormalizamos los datos, y visualizamos la predicción:



Visualizamos valores de Hidrometría conjuntamente con las Predicciones (ambas variables reales hasta el día 16, predichas a partir de entonces).



Vemos como, tanto en el intervalo de datos reales como de datos predichos, la Hidrometría sigue ese comportamiento que hemos venido observando desde el capítulo de análisis exploratorio: está muy influenciada por las Precipitaciones, de varios días anteriores.

7. Conclusiones

7.1 Recapitulación

En primer lugar, tras el pertinente preprocesado de los datos, se ha realizado un análisis para tratar de discernir cuáles son los factores que más influyen en la variable objetivo, la Hidrometría. Se concluyó que uno de los factores más influyentes son las Precipitaciones en días anteriores, ahora bien, hemos visto que la capacidad de influencia de las Precipitaciones varía según el momento del año que consideremos.

A continuación, se ha tratado de entender la naturaleza de esa variación. Al tratarse de una variación que se repite de forma estacional a lo largo del año, estaba claro que la variable Momento del año debía de participar en el modelado. Sin embargo, el Momento del año nos ayuda a entender cómo se comporta esta variación, pero no explica la causa de la misma. Para mejorar la explicabilidad del modelo, se trató de buscar otras variables que sí fuesen causa directa de esa variación (Temperatura, Precipitaciones de los últimos 3 meses), los resultados fueron parcialmente satisfactorios en este sentido.

Concluido el análisis, el desafío consistía en encontrar un modelo que consiguiese captar estas relaciones tan singulares que se dan entre nuestras variables de estudio. Se ensayó con un modelo de Redes Neuronales Recurrentes LSTM, y los resultados han sido bastante satisfactorios, tanto a nivel métricas como a nivel visual. Quizá el área de mejora más clara para nuestro modelo, sería que predice de menos cuando ocurren lluvias especialmente cuantiosas.

Finalmente, se ha ofrecido la predicción que se demandaba por parte de la empresa que propuso el desafío (horizonte temporal de un día), y adicionalmente hemos visto cómo se puede utilizar un modelo preparado para predecir horizontes temporales de un día, para extender nuestras predicciones a horizontes más lejanos.

7.2 Sugerencias para ampliar este estudio

Las sugerencias que se incluirán a continuación persiguen dos objetivos: mejorar la comprensión que se tiene acerca del modelo, y mejorar las predicciones.

- Entender mejor cuáles han sido las variables clave en las predicciones

Hemos determinado de forma empírica que, si utilizamos Precipitaciones, Temperatura e Hidrometría de los 7 días anteriores, podemos obtener predicciones bastante satisfactorias de la Hidrometría utilizando un modelo LSTM. Sería interesante conocer más acerca de

cuáles de estas variables han tenido más peso en la predicción. Además, como ya hemos visto, un modelo LSTM se diferencia de un modelo estándar de RNN por su capacidad de "recordar" patrones a largo plazo. Sería interesante intentar realizar predicciones con un modelo RNN estándar y comparar la precisión de ambos modelos, para comprobar si, efectivamente, hay una componente a largo plazo que hace necesario que se utilicen los modelos LSTM.

- Encontrar el factor que explica que las lluvias influyan de forma diferente en diferentes épocas del año

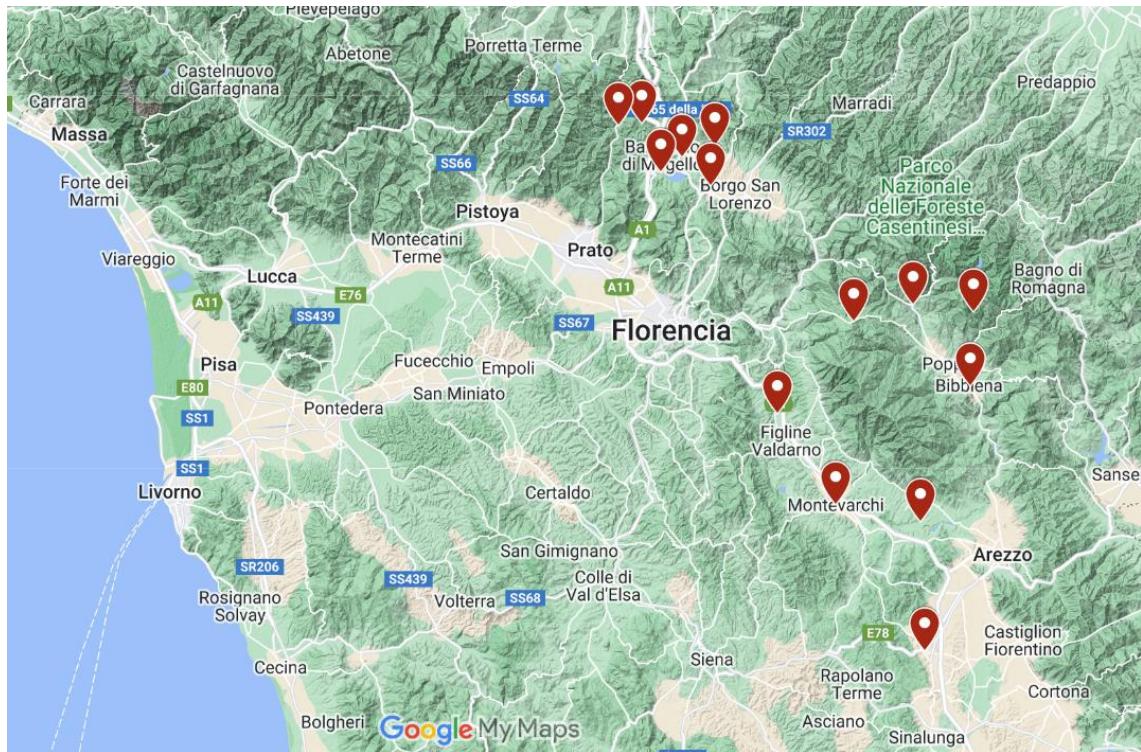
Como ya hemos visto, no influye de la misma forma en el Hidrometría una precipitación de la misma cuantía en invierno o en verano. Incluir la variable Momento del año nos ayuda a modelizar este efecto, pero ello no quiere decir que sea la causa raíz de este comportamiento. Se intentó encontrar la causa en las variables Temperatura y Precipitaciones acumuladas en los últimos 3 meses, pero los resultados no fueron del todo satisfactorios. Quizá también haya que considerar la Temperatura de los últimos meses, y no sólo la del día en cuestión (ya que la capacidad de drenado del suelo dependerá no sólo de si el día que estamos considerando es cálido, sino también de lo cálidos que hayan sido los días anteriores). Quizá se deba probar con lo ocurrido en 1, 2 , 4, etc meses anteriores, o quizá haya que ensayar con una combinación de ambas variables, y no las dos por separado. Y por último, se podría buscar la explicación en alguna variable que no esté incluida en el dataset.

- Considerar las medidas de los pluviómetros como variables separadas, y no la media de las mismas

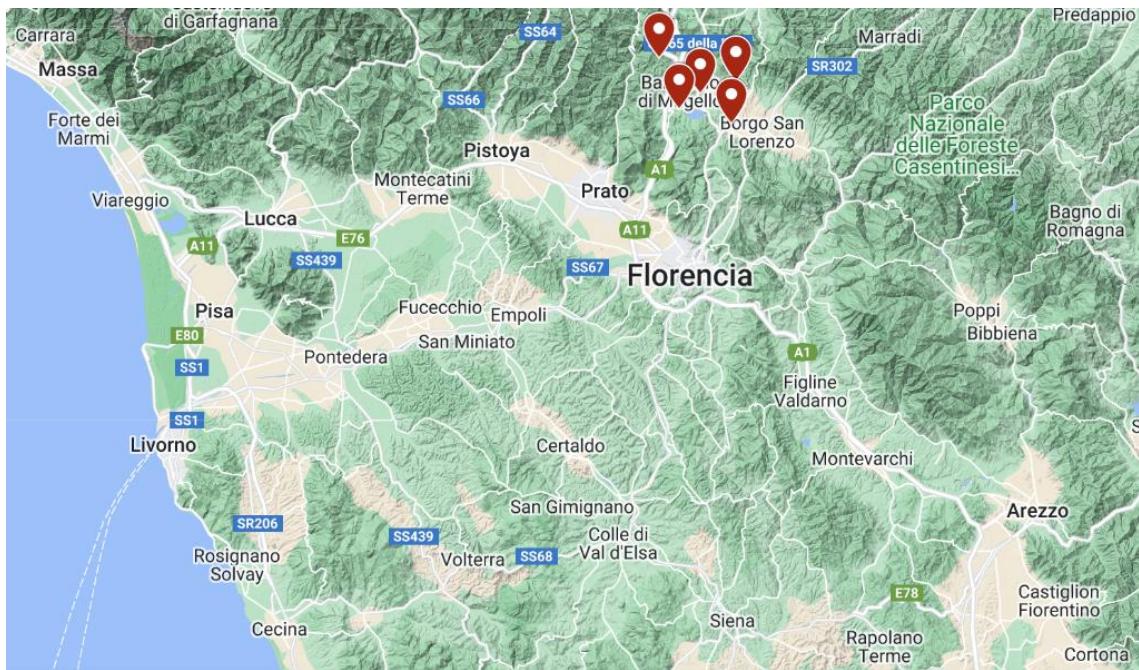
Aunque hemos visto en el capítulo 3.3 Análisis exploratorio: Precipitaciones que parece que todos los pluviómetros considerados para el estudio nos dan una información similar, se podría profundizar en el análisis para comprobar si realmente es así. Supongamos que hay puntos geográficos donde las Precipitaciones influyen más fuertemente en la Hidrometría (por ejemplo, por encontrarse en una región muy cercana al punto de medición, o en un valle que desagua directamente al río). Si llueve mucho en esas regiones, y llueve menos en otros puntos donde el agua caída no influye tanto en que aumente el nivel del río, cabría esperar un aumento grande en la Hidrometría. Pero en nuestro modelo, al hacer la media de las Precipitaciones de diferentes pluviómetros, ese efecto se ve amortiguado.

- Conseguir los datos faltantes de ciertos pluviómetros

Recuperamos el mapa con la localización de los pluviómetros que estaban presentes en el dataset:



Sin embargo, por falta de datos en ciertos intervalos, se tuvo que prescindir de las medidas de muchos de estos pluviómetros (Vernio, Stia, Consuma, Incisa, Montevarchi, S_Saviro, Laterina, Bibbiena, Camaldoli). Así, el mapa con los pluviómetros que se han podido utilizar en nuestro estudio, es este:



Vemos que no hemos podido utilizar información de los pluviómetros que están más río arriba. Sin embargo, es posible que las lluvias en las zonas aledañas río arriba, tengan mucha influencia en los niveles del río. Por tanto, si es posible, sería interesante recuperar esas mediciones y tenerlas en cuenta para la realización del estudio.

- Entender por qué el modelo predice de menos cuando hay crecidas bruscas del río

En la fase de validación del modelo, comprobamos que sistemáticamente se predicen Hidrometrías inferiores a las reales, cuando los niveles del río incrementan súbitamente. A fin de mejorar la precisión del modelo, sería interesante analizar con detenimiento esa casuística. Quizá parametrizando el modelo de forma diferente, podemos conseguir mitigar ese efecto. Incluso, si conseguimos datos adicionales de pluviómetros u otras variables, el modelo podría tener más información para predecir mejor esta casuística.

Aunque en principio este modelo se utilizaría para gestionar mejor los activos de una compañía de aguas, también podría ser utilizado para predecir crecidas peligrosas del río. En ese caso, cobraría aún más importancia una predicción correcta de dichas crecidas.

Referencias

Fulton, J. ARIMA Models in Python (Curso web). Recuperado el día 31 de Enero de 2023

<https://app.datacamp.com/learn>

Holbrook, R. Time Series (Tutorial web). Recuperado el día 3 de Febrero de 2023

<https://www.kaggle.com/learn/time-series>

Jansen, S. Manipulating Time Series Data in Python (Curso web). Recuperado el día 31 de Enero de 2023

<https://app.datacamp.com/learn>

Kulkarni, A., Booz, R., Toth, A., (9 de Agosto de 2022). What Is Time-Series Data?.

<https://www.timescale.com/blog/time-series-data/>

Mahanta, J. (10 de Julio de 2017). Introduction to Neural Networks, Advantages and Applications.

<https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207>

Siami Namin, S; Siami Namin, A. (15 de Marzo de 2018). Forecasting Economics and Financial Time Series: ARIMA vs. LSTM

<https://arxiv.org/ftp/arxiv/papers/1803/1803.06386.pdf>

Website de Tensorflow (versión en inglés). TensorFlow > Learn > TensorFlow Core > Tutorials > Time series forecasting. Recuperado el 23 de Enero de 2023.

https://www.tensorflow.org/tutorials/structured_data/time_series

Wikipedia (versión en inglés). Feature Scaling. Recuperado el 22 de Enero de 2023.

[https://en.wikipedia.org/wiki/Feature_scaling#Rescaling_\(min-max_normalization\)](https://en.wikipedia.org/wiki/Feature_scaling#Rescaling_(min-max_normalization))

[https://en.wikipedia.org/wiki/Feature_scaling#Standardization_\(Z-score_Normalization\)](https://en.wikipedia.org/wiki/Feature_scaling#Standardization_(Z-score_Normalization))

Anexo 1: Recabar datos de Temperatura desde una fuente de datos externa

Se han localizado en la web los datos históricos de temperatura en la ciudad de Florencia, incluidos aquellos del intervalo que nos interesa (desde finales de 2017 hasta la fecha final de las observaciones recogidas en el dataset, 2020-06-30).

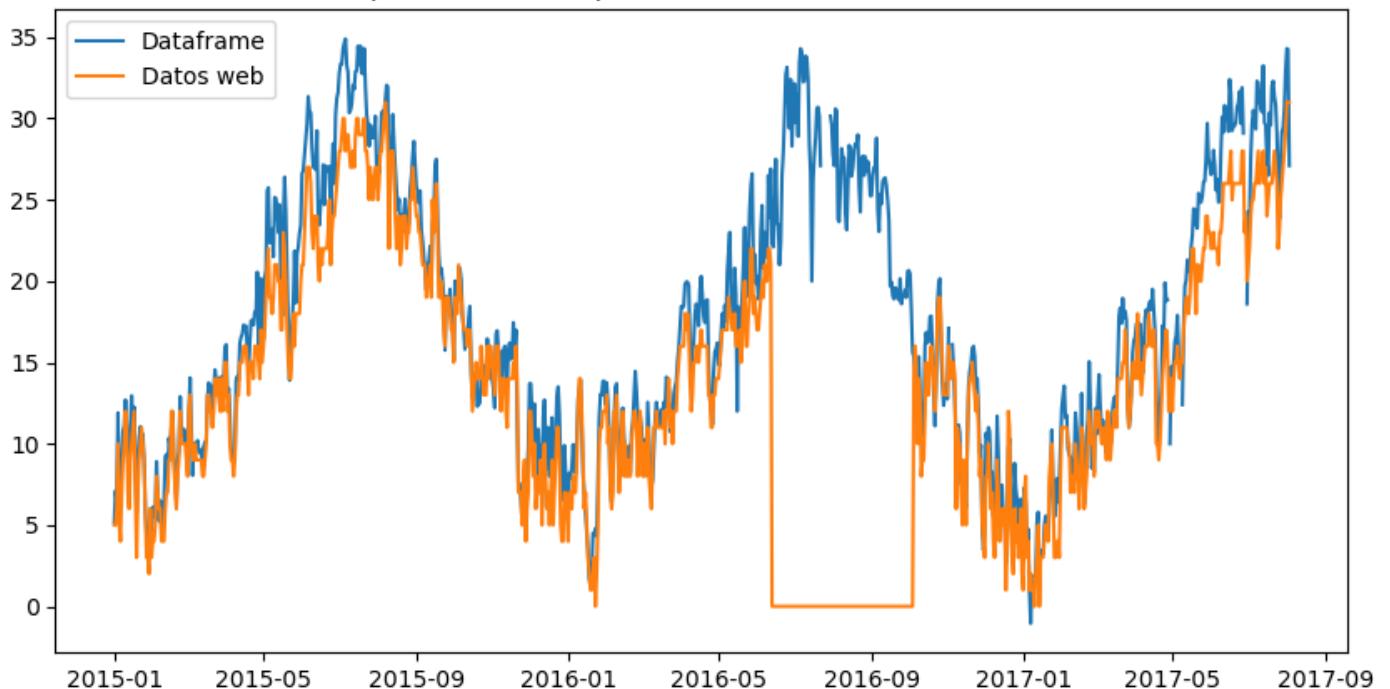
https://www.ilmeteo.net/meteo_Firenze-Europa-Italia-Firenze--sactual-29832.html

Nótese que las temperaturas están medidas en el aeropuerto de Florencia, mientras que desconocemos en qué estación de medición se tomaron las medidas que tenemos en nuestro dataframe. Tendremos que cerciorarnos de que los valores son equiparables.

Para ello, se copian a un csv no sólo las temperaturas medias en Florencia para el intervalo de fechas en el que no tenemos datos (2017-08-04 en adelante), sino que se cargarán datos desde el inicio de 2015, para estudiar la correspondencia entre las medidas que ofrece esta web, y las que tenemos en nuestro dataframe en el intervalo de solape (2015-01-01 a 2017-08-03).

En la siguiente visualización, se observan los valores de la Temperatura para la ciudad de Florencia en dicho intervalo de solape.

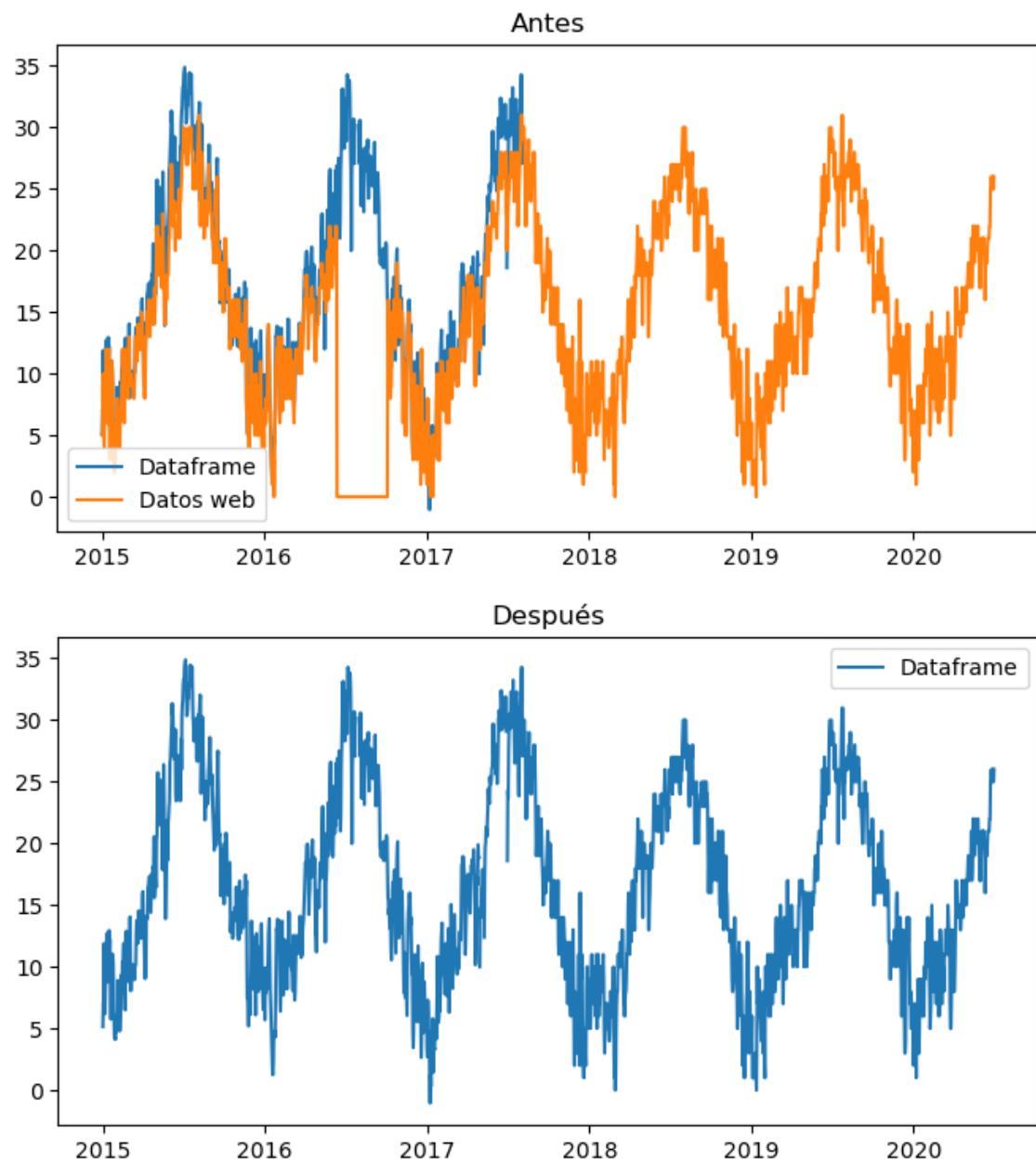
Comparativa de Temperatura entre dos fuentes de datos



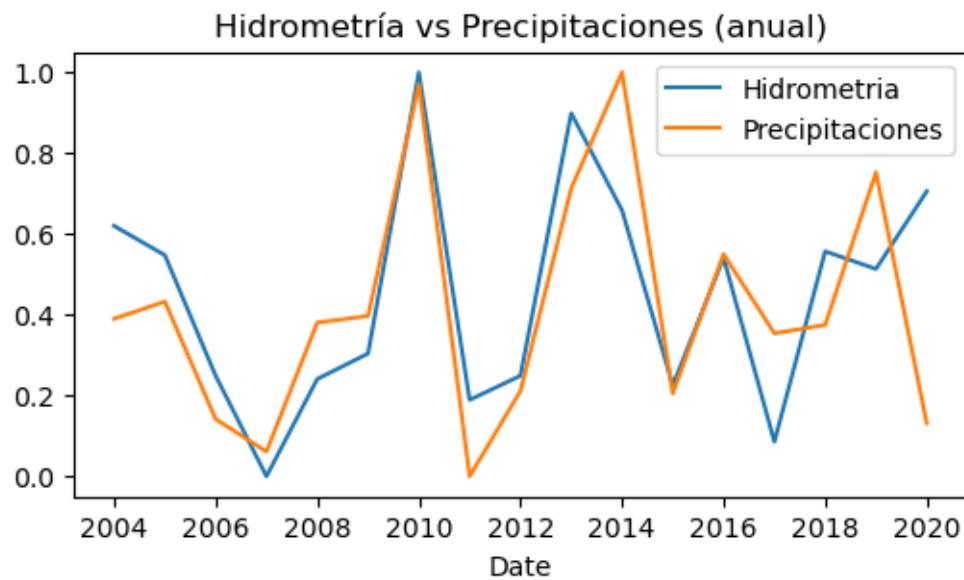
Más allá de los valores anómalos a mediados del año 2016 para los Datos web (que no nos afectarían puesto que no pretendemos utilizar este intervalo de medidas para nuestro

estudio), se observa que la temperatura medida en la estación que tenemos en el Dataframe es ligeramente mayor. En términos cuantitativos, la diferencia media entre ambas series es de 1,7 °C. La diferencia es apreciable, pero quizás sean lo suficientemente parecidas para no desvirtuar el estudio que pretendemos realizar.

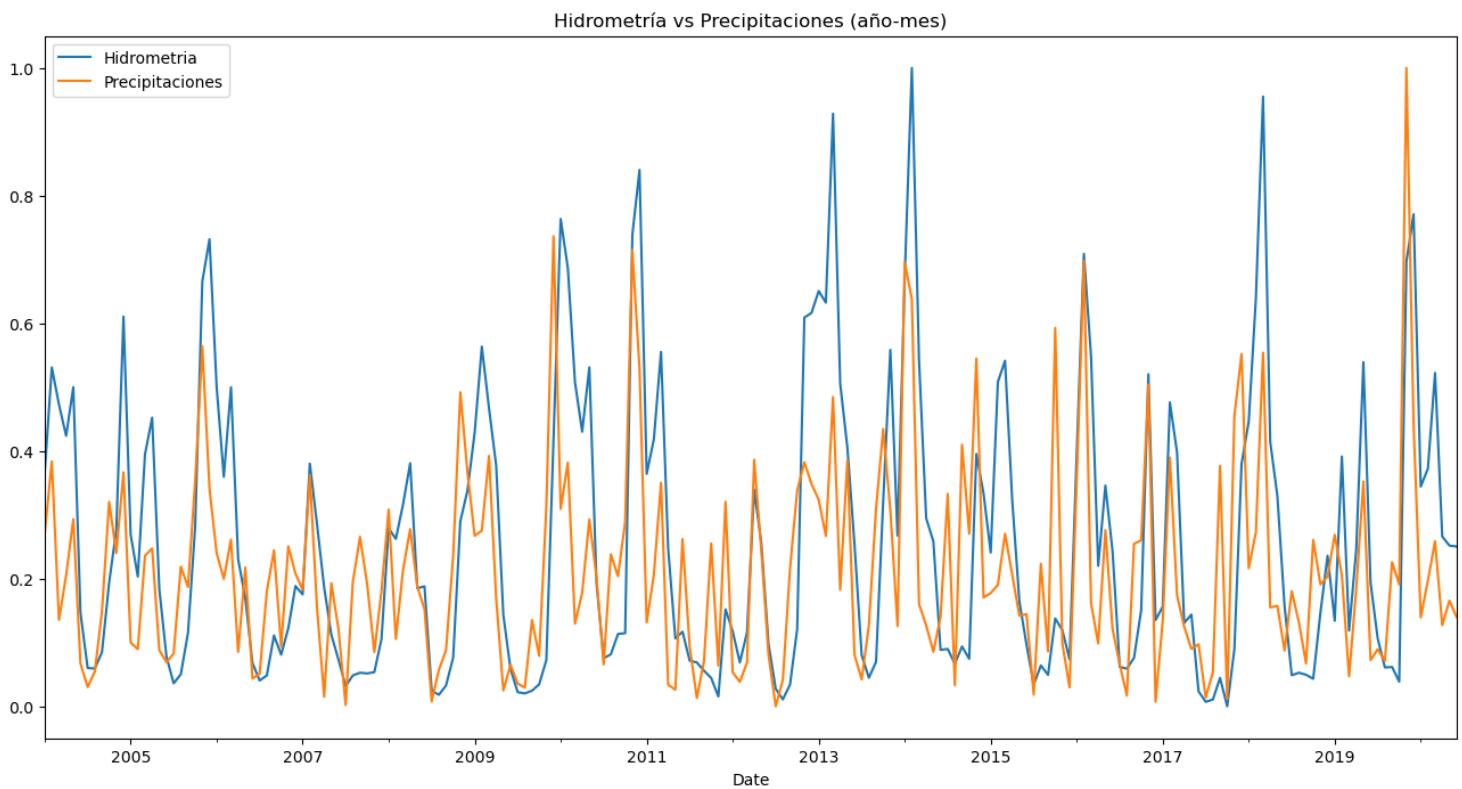
Así, decidimos incorporar las medidas de temperatura del aeropuerto de Florencia para el intervalo en el que no teníamos datos (2017-08-04 a 2020-06-30). Las siguientes visualizaciones resumen a nivel gráfico la incorporación de estos nuevos datos al dataframe:



Anexo 2: Análisis de valores anómalos en la correlación Hidrometría-Precipitaciones



Para la anomalía en cuestión, estamos extrayendo información a partir de datos agrupados por año, vamos a bajar a la granularidad de mes para observar este comportamiento anómalo.



En esta granularidad, no parece que haya una pérdida grande de correlación a partir de 2017.

Si computamos los coeficientes de correlación (bajando a la granularidad de día):

- Coeficiente de correlación Hidrometría-Precipitación de 2004 a 2016: 0.290
- Coeficiente de correlación Hidrometría-Precipitación de 2017 a 2020: 0.147

Sigue habiendo diferencia, pero mucho menos acusada que para el cálculo de la granularidad año. Además, hay que recordar que para el año 2020 no tenemos la serie completa, los datos terminan el 2020-06-30, y puede ser que en la primera mitad del año la relación entre ambas variables sea diferente a la relación que se tiene computando el año entero.

Haría falta una análisis más profundo para determinar si se trata de una anomalía introducida a la hora de procesar los datos (granularidad año, normalización min-max), o se trata de un patrón que genuinamente está en los datos. Por ahora, anotaremos esta anomalía y la tendremos en mente si surgen problemas en la fase de modelado.

Anexo 3: Obtención de la variable “Momento del año”

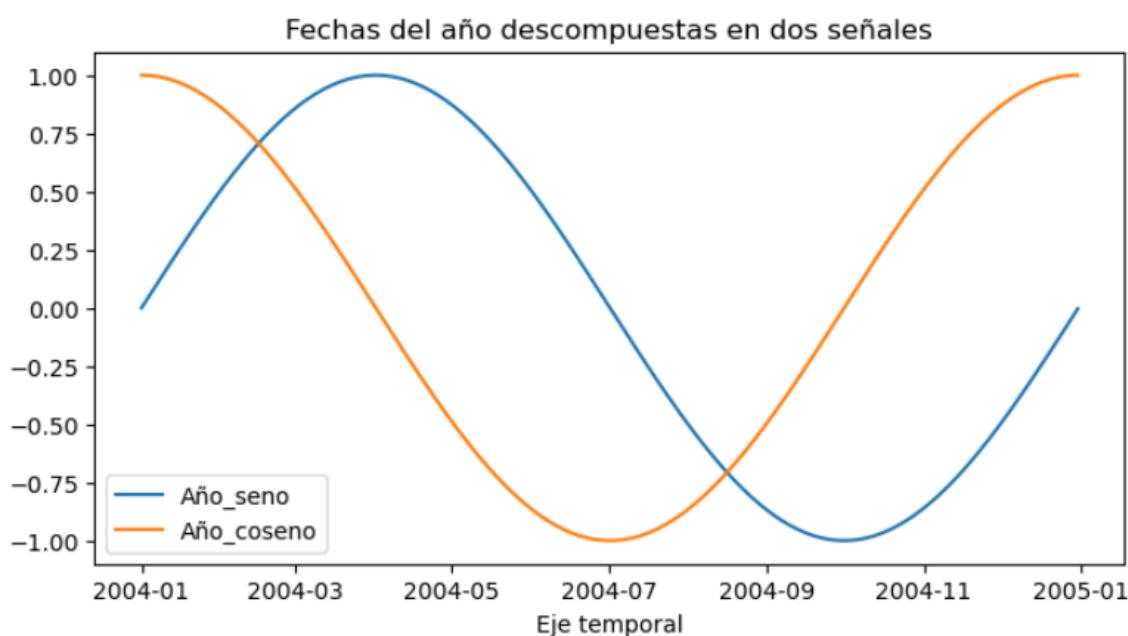
A la hora de introducir la variable temporal en un modelado, hay varias formas en que se puede abordar la periodicidad (Website de Tensorflow. Time series forecasting tutorial, 2022).

Efectivamente, en nuestro caso particular nos interesa introducir la variable temporal de forma periódica, con un periodo de un año de duración. Como hemos visto en los capítulos de análisis exploratorio, nuestra variable objetivo sigue un ciclo anual. Por tanto, a la hora de entrenar el modelo y realizar las predicciones, nos interesa informar la variable temporal como una señal que se repite cíclicamente año tras año. Así, el modelo podrá captar patrones que se vayan repitiendo en las mismas fechas de años distintos.

Para introducir el Momento del año como variable de estudio, se va a transformar cada fecha del año en dos variables: Año_seno, y Año_coseno. Se incluye la siguiente visualización para ilustrar cómo funcionaría esta transformación, tomando un año en concreto.

```
plt.figure(figsize=(8,4))
axes = plt.plot(df.loc['2004'].index.to_list(), np.sin(np.arange(len(df.loc['2004'])) * (2 * np.pi / 365.2425)))
axes[0].set_label('Año_seno')
axes = plt.plot(df.loc['2004'].index.to_list(), np.cos(np.arange(len(df.loc['2004'])) * (2 * np.pi / 365.2425)))
axes[0].set_label('Año_coseno')
plt.xlabel('Eje temporal')
plt.title('Fechas del año descompuestas en dos señales')
plt.legend()

plt.show()
```



Vemos como, utilizando estas dos señales, se puede identificar periódicamente, y de forma única, cada momento del año (puede ser que dos momentos del año tengan valores de Año_seno iguales, pero sus valores de Año_coseno serán diferentes, y viceversa).