UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Predicción de Energía Solar con Modelos Autorregresivos

Daniel Cuesta Santos
Tutor: Ángela Fernández Pascual
Ponente: José Ramón Dorronsoro Ibero

OCTUBRE 2019

Predicción de Energía Solar con Modelos Autorregresivos

AUTOR: Daniel Cuesta Santos
TUTOR: Ángela Fernández Pascual

Dpto. Ingeniería informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Octubre de 2019

Resumen (castellano)

Este Trabajo Fin de Grado consistirá en el estudio de modelos autorregresivos para la predicción de energía solar en España. El estudio se produce debido a que actualmente la gran cantidad de contaminación producida por el uso de combustibles fósiles y la obtención de energía no renovables está afectando al planeta. La energía solar, al ser una energía renovable, aparece hoy en día como una de las principales alternativas a estos recursos y es interesante poder predecir la obtención que se podría tener en el futuro.

La obtención de energía solar depende de las condiciones meteorológicas y su predicción es fundamental para poder tenerla en cuenta como fuente de energía en la red eléctrica. En este Trabajo Fin de Grado vamos a dar una predicción de la energía solar y para ello utilizaremos los modelos autorregresivos. Determinaremos cuál de ellos podrá llegar a predecir con mas exactitud los valores reales de energía solar obtenida por las placas solares.

Los datos de los que disponemos son claramente dependientes del tiempo, podemos tratarlos como series temporales. Entre los modelos que permiten hacer predicciones en series temporales destacan los modelos autorregresivos (AR, MA, ARMA y ARIMA), que son los que utilizaremos y compararemos en este trabajo. Antes de predecir, es importante realizar un buen análisis de los datos. En el caso concreto de las series temporales, haremos un hincapié especial en conseguir series no estacionarias, para lo cual quitaremos tendencias y comportamientos estacionales.

Después de ese análisis y correcto tratamiento de los datos ya podemos empezar a usarlo para predecir, pero primero tendremos que dividir estos datos en dos conjuntos: uno para el entrenamiento y otro más pequeño para el test. Las predicciones que realizaremos con estos modelos las haremos usando horizontes, concretamente cinco horizontes. Con ellos obtendremos las predicciones de las siguientes cinco horas. Esto nos dará para poder comparar tanto los modelos como los horizontes y así determinar qué modelo es el mejor para cada horizonte.

Abstract (English)

This End of Grade Work will consist of the study of self-regressive models for solar energy prediction in Spain. The study is because the large amount of pollution produced by the use of fossil fuels and the production of non-renewable energy is affecting the planet. Solar energy, being a renewable energy, appears today as one of the main alternatives to these resources and it is interesting to be able to predict the obtaining that could be had in the future.

Obtaining solar energy depends on the weather conditions and its prediction is essential to be able to take it into account as a source of energy in the electricity grid. In this End of Degree Work we will give a prediction of solar energy and for this we will use the self-regressive models. We will determine which of them will be able to more accurately predict the actual values of solar energy obtained by solar panels.

The data we have are clearly time-dependent, we can treat them as time series. Models that allow predictions in time series include the self-regressive models (AR, MA, ARMA and ARIMA), which are the ones that we will use and compare in this work. Before predicting, it is important to perform a good analysis of the data. In the specific case of time series, we will place a special emphasis on getting non-stationary series, for which we will remove seasonal trends and behaviors.

After that analysis and correct processing of the data we can start using it to predict, but first we will have to divide this data into two sets: one for training and one smaller for the test. We will make the predictions that we will make with these models using horizons, specifically five horizons. With them we will get the predictions of the next five hours. This will give us to be able to compare both the models and the horizons and thus determine which model is best for each horizon.

Palabras clave (castellano)

Inteligencia artificial, aprendizaje automático, modelos autorregresivos, predicción por horizontes, estacionalidad, autocorrelación, energía solar.

Keywords (inglés)

Artificial intelligence, machine learning, self-regressive models, horizon prediction, seasonality, autocorrelation, solar energy.

A mi tutora que me ha permitido adquirir nuevos conocimientos, a mi familia y amigos. Especialmente a María.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

En un mundo en el que las tecnologías son tan cotidianas y el uso de electricidad es algo básico para todo el mundo, y en el cual, la obtención de ella se realiza de múltiples formas, también, existe un factor negativo: algunas de estas formas de obtener la electricidad son perjudiciales para el medio ambiente y esto provoca el tan temido cambio climático.

Diariamente podemos ver que la polución en grandes ciudades se está intentando controlar con prohibiciones de entrada a vehículos altamente contaminantes y fomentando el uso de vehículos menos contaminantes o cero contaminantes; así como bicicletas, patinetes eléctricos incluso hasta coches eléctricos.

En base a este cambio climático y al mayor uso de vehículos eléctricos con el objetivo de obtener unos resultados que puedan servir para determinar si una zona específica puede ser un mejor lugar para la explotación de energía solar, hemos hecho el estudio presentado a continuación.

En esta memoria de Trabajo de Fin de Grado, explicaremos todo lo realizado en la obtención de una predicción de energía solar en una determinada zona. En este caso, centrada en la predicción de energía solar en España. Los datos con los que trabajaremos son los correspondientes entre los años 2010 y 2015, incluyendo ambos años.[1]

1.2 Objetivos

El principal objetivo del estudio realizado para este Trabajo de Fin de Grado se centra en predecir la energía solar obtenida en España durante el año 2015 con un modelo autorregresivo.

Para llegar al objetivo principal, debemos llevar a cabo los siguientes objetivos secundarios:

- Obtener y ordenar una gran cantidad de datos sobre energía solar por fecha y hora con los que trabajar.
- Analizar y tratar los datos de diferentes formas para poder trabajar con ellos.
- Realizar una búsqueda de los valores para nuestros modelos
- Realizar un estudio de los modelos AR, MA, ARMA y ARIMA con cinco horizontes.
- Comparar todos los resultados y determinar cual es el mejor modelo y su horizonte.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **En el capítulo 2 tendremos el estado del arte, donde introduciremos todo sobre el tema que vamos a tratar y explicar el funcionamiento.**
- **En el apartado 3 abordaremos el diseño realizado para el estudio, paso a paso y explicándolo.**
- **El apartado 4 veremos las pruebas realizadas y los resultados obtenidos junto con la comparativa de modelos.**
- **En el último apartado (apartado 5) tendremos una conclusión sobre las pruebas y los resultados. Además, en este apartado tendremos un subapartado en el cual exponemos trabajos futuros sobre este tema.**

2 Estado del arte

2.1 Aprendizaje automático

En este capítulo explicaremos en qué consiste el aprendizaje automático, cómo funciona y los pasos que se llevan a cabo para una buena predicción. La estructura que posee el aprendizaje automático es la que podemos observar en la Figura 2-1. La cual contiene datos de ejemplo, un algoritmo y un modelo; con ello, y unos nuevos datos que utilizar en la máquina obtendremos los resultados de nuestra predicción.

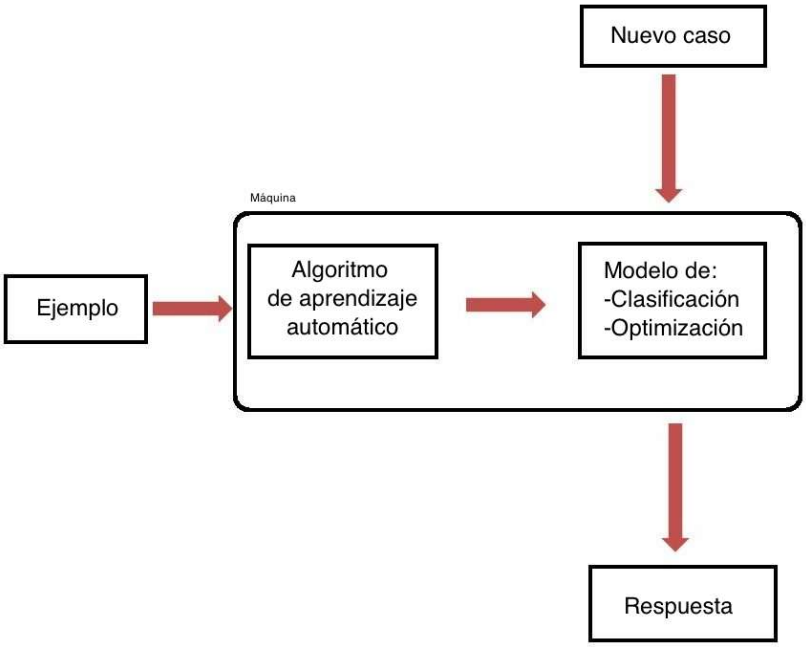

Figura 2-1: Aprendizaje automático (Fuente: *https://www.pinterest.es/grecispons/aprendizaje-autom%C3%A1tico/*)

En primer lugar, tendremos que saber qué es el aprendizaje automático. El aprendizaje automático o machine learning es un tipo de inteligencia artificial que permite a las máquinas aprender a través de unos datos proporcionados [2]. Cuantos más y diferentes sean los datos recibidos más criterio tendrán para predecir de forma más afectiva.

2.1.1 Orígenes del aprendizaje

Antes de empezar con el aprendizaje automático tenemos que saber que el aprendizaje es una cualidad en los animales. Utilizamos a los animales como ejemplo para el aprendizaje porque son los seres vivos que poseen la inteligencia más primitiva. Este aprendizaje en los animales puede ser producido o creado por diferentes factores, por ejemplo, el animal tiene hambre por lo que buscará comida, encontrará algún fruto sabroso y este alimento lo recordará para futuros momentos en los que necesite alimentarse. Hay diferentes formas de

aprendizaje: por, prueba y error, por estímulos similares, por imitación… Puede ocurrir que unos y otros se entrelacen para un mayor aprendizaje [3].

2.1.2 Funcionamiento del aprendizaje automático

Ahora ya sabemos de dónde viene el aprendizaje y es este el que se quiere transmitir a las máquinas para que ante estímulos similares a los aprendidos reaccione y siga aprendiendo de ellos. Este aprendizaje es el aprendizaje automático que buscamos.

El aprendizaje automático tiene diferentes algoritmos con los que aprender, estos algoritmos los concretamos más adelante y nos centraremos en uno de ellos que será el que usemos en la investigación. Sin embargo, aunque el algoritmo sea distinto, sigue la misma estructura para realizar el aprendizaje.

Primero, tenemos un conjunto de datos el cual será utilizado por el algoritmo de aprendizaje para aprender de ellos. A este conjunto de datos se le llama conjunto de entrenamiento. Cada algoritmo los usa de una manera y con ello eligen un modelo de clasificación para que con futuros datos que les proporcionamos pueda clasificar de forma correcta y aprender nuevamente con esos datos si fuera necesario. Este conjunto de datos futuros se llama conjunto de test.

Hay algoritmos de aprendizaje que pueden ser conjuntamente o que con pequeñas variaciones sean similares a otros. Un ejemplo de algoritmo sería coger los datos y dividirlos en dos partes y utilizar uno de ellos de forma que sirva de comprobación para ver que la predicción que realiza es óptima.

En esta prueba obtendremos unos resultados con los cuales elegiremos qué modelo nos permitirá una predicción más cercana a la realidad.

Una vez estudiado y analizado el modelo más adecuado para que obtengamos resultados más cercanos a la realidad, lo que haremos será predecir datos nuevos, para los que desconocemos su valor.

2.2 Diferentes tipos de Algoritmos

Como hemos visto en el anterior apartado existen diferentes tipos de algoritmos, cada uno de ellos tiene una funcionalidad diferente, ya que depende del tipo de datos con el que tratemos, lo que queramos predecir o la salida que producen.

Los tipos que explicaremos serán supervisado, no supervisado y otros tipos que son combinación de los anteriores consigo mismo o con el otro tipo de formas diferentes. Uno de ellos será el que nosotros usemos y profundizaremos más en él.

2.2.1 Supervisado

Este algoritmo de aprendizaje es del tipo que describimos en la sección 2.1.2 y será el que nosotros utilizaremos para nuestra investigación.

La forma de usar el conjunto de datos será la siguiente:

Primero, tenemos un conjunto de datos el cual dividiremos en dos: una parte, será el conjunto de datos que daremos a la máquina para que su algoritmo seleccionado aprenda de esos datos, también llamados datos de entrenamiento y el otro grupo de datos, mucho más pequeño, será nuestro conjunto de validación o conjunto de datos de prueba, el cual la máquina con el aprendizaje adquirido deberá predecir los datos que hay en este último conjunto [4]. Con ambos datos compararemos y obtendremos una media de error. Este proceso se llama validación cruzada.

Existen diferentes tipos de validación cruzada: de K iteraciones, aleatoria, dejando uno fuera. Los tipos de validación cruzada de K iteraciones y aleatoria tienen un funcionamiento igual pero lo que cambia es la forma en la que elige los datos de entrenamiento y los datos de prueba. El primero, elige por bloques de datos mientras que el otro elige los datos aleatoriamente como su mismo nombre indica. La validación cruzada dejando uno fuera utilizará como conjunto de datos de prueba un solo valor y el resto de los datos servirán de entrenamiento. En este último caso se repetirá la operación seleccionando otro dato N veces, siendo N la cantidad de datos en el conjunto [5].

En cambio, en los otros tipos este proceso se repetirá tantas veces como el valor K, siendo K el número de iteraciones que se quieren realizar, en cada iteración el conjunto de datos de prueba será distinto para predecir y comparar con datos distintos, como podemos ver en la Figura 2-1, es un ejemplo de validación cruzada con K iteraciones.

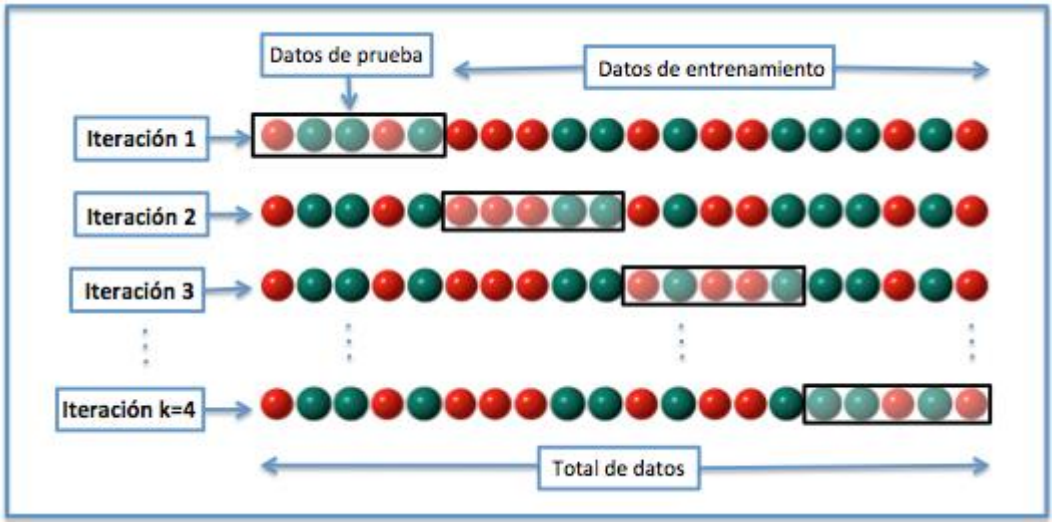

Figura 2-2: Validación cruzada de K iteraciones con K=4 (Fuente: *http://www.cotradingclub.com/2017/05/17/validacion-cruzada*)

2.2.2 No Supervisado

En el aprendizaje no supervisado no se conoce la salida que queremos predecir, por ello es todo lo contrario al anterior aprendizaje, esto quiere decir que no realiza una validación con los datos para saber si su salida es correcta o no.

Dentro de este aprendizaje hay diferentes tipos también y estos son:

- Análisis de las componentes principales. Este tipo lleva a cabo un estudio de los datos para determinar qué parte del conjunto de los datos caracterizan más y así los que menos lo hagan pueden llegar a ser eliminados sin afectar al conjunto.

- Agrupamiento. Realiza una observación de los datos para dividirlos en grupos diferentes y así clasificarlos por sus características.

- Prototipado. Utilizando la misma técnica que en el anterior tipo, pero con la diferencia que no nos dirá a la clase a la que pertenece, sino que se obtiene un prototipo de esa clase.

- Codificación. La salida de este algoritmo es el dato de forma codificada esto quiere decir que es el dato con un tamaño más pequeño y con la mayor información posible. Este tipo se podría usar para enviarlo por un canal limitado, pero al otro lado tendría que poder recuperar el dato como originalmente estaba.

- Extracción y selección de características. Consiste en coger los datos de entrada y crear un mapa topológico con ellos y determinar un conjunto nuevo de datos de menor dimensión que mantengan la misma información.

Repetimos que todas ellas no realizan ningún tipo de comprobación para saber si los resultados obtenidos son erróneos o correctos [6].

2.2.3 Otros

Explicaremos el resto de los algoritmos de aprendizaje en este apartado, entre los que destacan:

Semisupervisado. En estos algoritmos usan los dos algoritmos anteriores, supervisado y no supervisado, de forma combinada. Es una combinación porque utiliza datos etiquetados y datos no etiquetados, según los investigadores de ello si se usa un pequeño grupo de etiquetados el aprendizaje puede mejorar respecto a un aprendizaje puramente no supervisado [7].

Por refuerzo. Se basa en ensayo-error, quiere decir, que con una acción del agente el entorno devolverá un estado y eso dará una recompensa. Esto es así porque no recibe información de la salida, su única información es la recompensa que reciba. Lo que se desea con este aprendizaje es que obtenga la mayor recompensa posible [8].

Transducción. El aprendizaje de este algoritmo es parecido al aprendizaje supervisado, excepto porque este algoritmo pretende predecir las categorías de los datos nuevos guiado por los datos de entrada anteriores [9].

Multi-Tarea. Este algoritmo se basa en no solo aprender a resolver un determinado problema sino a la vez también aprender otros problemas relacionados y compartir lo aprendido entre las diferentes resoluciones para así identificarlos mejor [10].

2.3 Modelos autorregresivos

Este modelo será en el que nos centremos y llevemos acabo con el todo el proceso de aprendizaje automático. El modelo consiste en determinar que en un momento en el tiempo que valor va a tomar, para ello se observara su evolución en el tiempo y también el patrón que puedan llegar a tener esos datos. En este caso al ser autorregresivo el valor dependerá también de su valor pasado, es decir, se irán prediciendo valores y el valor que se prediga será utilizado también para la predicción siguiente [11, capitulo 2].

En estos modelos existen distintos tipos de procesos y diferentes tipos de datos que usar en ellos. En nuestro caso utilizaremos los llamados series temporales y como proceso autorregresivo ARIMA.

2.3.1 Series Temporales

En primer lugar, los datos con los que vamos a trabajar serán creados como una serie temporal. ¿Qué es una serie temporal?
Una serie temporal es un conjunto de observaciones cada una de ellas con un tiempo determinado. Tendremos, por ejemplo, una fecha y una hora relacionada con el valor que le corresponde, además los datos estarán ordenados cronológicamente [12, capitulo 1]. Estos valores pueden repetirse o ser muy similares durante periodos de tiempo, esto sería denominado estacionalidad.

La estacionalidad ocurre cuando un grupo de valores consecutivos se repiten en los siguientes periodos de tiempo. Esta repetición se realizará cada *S* estaciones, siendo *S* el número de valores contenidos en el conjunto [11, capitulo 3] [12, capitulo 2].

Muchos de los modelos funcionan bien sobre series de datos no estacionarias, pero esto es complicado ya que muchos de ellos se ven influenciados por el tiempo. Para quitar la estacionalidad y tener una serie más adecuada lo que se hace es restar los valores unos con otros, pero de forma ordenada, es decir, restar los valores con los mismos valores con una diferencia de tiempo entre ellos de *S*.

En este trabajo vamos a ver un conjunto de modelos para predecir series temporales, en concreto los modelos autorregresivos. Dentro de estos modelos destaca el modelo ARIMA que se explica a continuación.

2.3.2 Modelo ARIMA(p,d,q)

ARIMA es el acrónimo de *AutoRegressive Integrated Moving Average*, que traducido es un modelo autorregresivo integrado de medias móviles. Es un modelo estadístico que con variaciones y regresiones de los datos consigue descubrir patrones para una predicción futura, es por tanto un modelo dinámico de series temporales. Esto quiere decir que cuando prediga futuros datos, estos estarán relacionados con los pasados y no con variables aleatorias [13].

Para la realización de este modelo se debe seguir los siguientes pasos:

- **Primer paso.** Con los datos se investigará un modelo ARIMA(p,d,q) que sea apropiado. Esto significa seleccionar un p, d, q que vayan a imitar las características de la serie estudiada. Puede que algunos modelos sean muy similares.

- **Segundo paso.** Dado el modelo se procederá a la estimación de los resultados con dicho modelo.

- **Tercer paso.** Se compararán las predicciones con los datos originales para comprobar qué modelo se ajusta más al verdadero resultado.

- **Cuarto paso.** Una vez seleccionado el modelo más idóneo para los datos y para la predicción que se necesita, se realizará el pronóstico de nuevos valores futuros. También se estudiará su capacidad de predicción.

Este ciclo se puede repetir para diferentes p, d, q, ya que dependiendo del valor que se le dé a cada uno de ellos las predicciones pueden ser más ajustadas o menos [11, capitulo 5].

En estos modelos la variable *p* indica el orden de la parte autorregresiva, *d* indica la cantidad de diferenciación y q indica el orden de la parte del promedio móvil. El modelo ARIMA lo podemos ver como la unión del modelo AR y del modelo MA, ambos modelos están englobados dentro del modelo ARIMA, ya que sus funciones realizan el mismo objetivo.

$$ARIMA(p,0,0) = AR(p)$$

$$ARIMA(0,0,q) = MA(q)$$

En los casos $d = 0$ la serie sería de tipo estacionaria y eso reduciría al modelo ARIMA en un modelo ARMA.

$$ARIMA(p,0,q) = ARMA(p,q)$$

Una vez vista la intuición de estos modelos, vamos a ver su definición matemática:

Modelo AR(p).
Modelo autorregresivo de orden p, que sigue la siguiente ecuación.

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t$$

Siendo Y_t respuesta dependiente del tiempo t.
Y_{t-1}, Y_{t-2}, …, Y_{t-p} variables de respuesta en los intervalos de tiempo t-1, t-2, …, t-p, respectivamente.
ϕ_0, ϕ_1, ϕ_2, …, ϕ_p los coeficientes a estimar.
ε_t error en el tiempo t.

Modelo MA(q).
Modelo de medias móviles con orden q, su forma es la siguiente.

$$Y_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon t_{-q}$$

Siendo Y_t respuesta dependiente del tiempo t.
μ media del proceso.
θ_1, θ_2, …, θq los coeficientes a estimar.
ε_t error en el tiempo t.
ε_{t-1}, ε_{t-2}, … ε_{t-q} errores de periodos de tiempo anteriores

Modelo ARMA(p,q).
Modelo autorregresivo de media móvil.

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi p\, Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$$

Para la obtención de los valores de *p* y *q* usaremos las funciones ACF y PACF respectivamente, sus gráficas nos harán determinar el modelo más adecuado. Las gráficas son de la función de autocorrelación de muestra (ACF) y de la función de autocorrelación parcial de muestra (PACF) [14]. Los resultados de la gráfica son posibles valores para nuestras variables y así encontrar el modelo más idóneo para la predicción. Cuando la autocorrelación es muy pequeña podemos deducir que el dato ya no es relevante y por tanto podemos tomar ese punto como valor del parámetro *p*.

2.3.3 Horizontes

Con los modelos ya elegidos y los valores para sus variables podremos empezar a predecir valores futuros, que es nuestro objetivo final.

Al hacer una predicción, se suele hacer después de haber realizado un entrenamiento para producir un aprendizaje. Pero, en este caso, lo que realizamos es un aprendizaje continuo; es decir, tenemos nuestro grupo de valores que el modelo usa para aprender y luego, predecimos el siguiente valor en el tiempo y este valor lo añadimos al conjunto de aprendizaje y volvemos a realizar el proceso anterior.

Este proceso se llama ‘horizontes’ y podemos predecir el número de horizontes que deseemos. Si solo queremos un valor tendremos un horizonte y si queremos los dos siguientes valores tendremos dos horizontes, así sucesivamente.

Los horizontes poseen un problema: la predicción es más inexacta cuanto más alejado está el horizonte del último valor de entrenamiento. Las predicciones suelen ir cogiendo una

tendencia continuada, pero si los horizontes empiezan a estar muy lejos del último valor de entrenamiento comienzan a promediarse y predicen el mismo valor. Si estos valores se muestran gráficamente veremos que llegará un punto en el que habrá una línea horizontal, ya que las predicciones tienden hacia la media de los valores.

2.3.4 Validación cruzada en series temporales

En un apartado anterior hablamos de la validación cruzada y que en nuestro caso con series temporales sería un tanto distinta debido a que los valores poseen fecha. Mientras en la validación cruzada la división en conjuntos se hace con datos aleatorios, aquí hay que mantener el orden temporal, por lo que los datos de conjunto deben ser consecutivos, quiere decir que, si el conjunto total posee datos de 3 años, nosotros debemos dividirlo en tantos conjuntos como queramos, pero los datos de cada conjunto deben ser consecutivos. De esta forma tendremos un conjunto que usaremos de validación y el resto de ellos para entrenamiento.

3 Diseño

3.1 Obtención de datos

Los valores que tenemos son datos de energía solar obtenida en toda España, estos están obtenidos cada hora dentro de cada día entre el año 2010 y el año 2015, estos incluidos. Dichos datos se localizan en un Excel .csv, el cual leemos de forma ordenada desde el 01/01/2010 00:00:00 hasta el 31/12/2015 23:00:00. Los datos los encontramos en la página Kaggle en la que hay datasets sobre diferentes temas [1].

Con la función *read_csv* de la librería de *pandas* recogemos todos los datos anteriores y se los asignamos a una variable. Esta variable será nuestra serie temporal con la que trabajaremos. Esta función, permite construir nuestra serie temporal que se llamará *"solar"*, con la serie temporal podemos obtener información a través de la función *describe ()* Figura 3-1. Veremos las estadísticas que tiene nuestra serie temporal, como la media, máximo, mínimo, etc.…

```
In [6]:  1 #print(solar['2010-01'])
         2 solar.describe()

Out[6]: count    52584.000000
        mean         0.172149
        std          0.222958
        min          0.000000
        25%          0.000000
        50%          0.016081
        75%          0.340342
        max          0.774285
        Name: ES, dtype: float64
```

Figura 3-1: Estadísticas de la serie temporal

3.2 Análisis de los datos

Los datos obtenidos los pasaremos por distintas pruebas para ver cómo están organizados y cómo se van cambiando según el tiempo. Esto lo realizamos porque los datos de energía solar en invierno no son iguales que los de verano, ya que en verano hay más horas de luz. También, porque por el día si se puede recoger energía solar y por la noche es imposible obtenerla. Todas las pruebas nos llevarán a la misma conclusión: los datos de energía solar poseen doble estacionalidad, una estacionalidad diaria y una estacionalidad anual, las cuales deberemos quitar para usar los valores y así, poder predecir sin ninguna influencia de la estacionalidad.

3.2.1 Dibujando los datos

Lo primero que haremos será observar los datos cómo cambian según trascurre cierto tiempo. Para ello, usaremos la función *plot* que nos representara los datos. Pintaremos datos de diferentes periodos de tiempo.

Para empezar, veremos los valores de un año, en este caso del año 2010. En la gráfica podemos observar que los valores más altos se sitúan en los meses más centrados y los valores bajos en los primeros meses del año y en los últimos.

```
plt.figure(figsize=(20,5))
solar['2010'].plot(title='VALORES DE 2010')
plt.show()
```

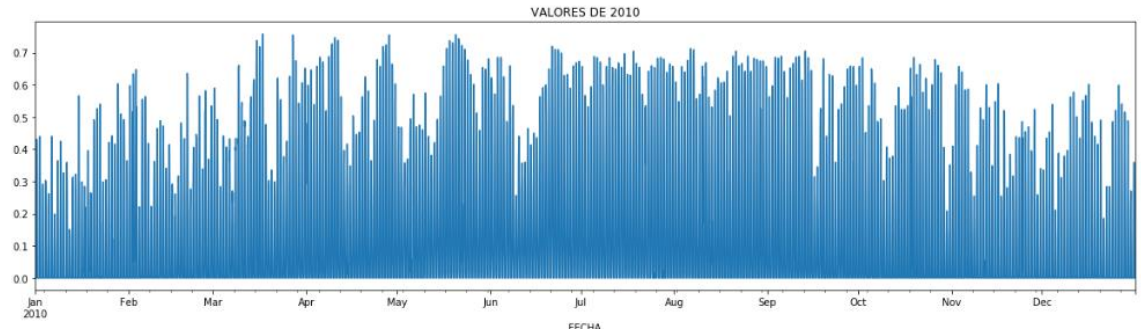

Figura 3-2: Gráfica de los valores de 2010

Otra forma de ver estas subidas y bajadas es mostrar todos los datos desde el 2010 hasta el 2015 incluido. En esta gráfica se distinguen 6 arcos uno por cada año, así podemos ver que existe una estacionalidad anual, la cual más adelante diremos cómo la quitaremos.

```
plt.figure(figsize=(20,5))
solar['2010':'2015'].plot(title='VALORES DE 2010 A 2015')
plt.show()
```

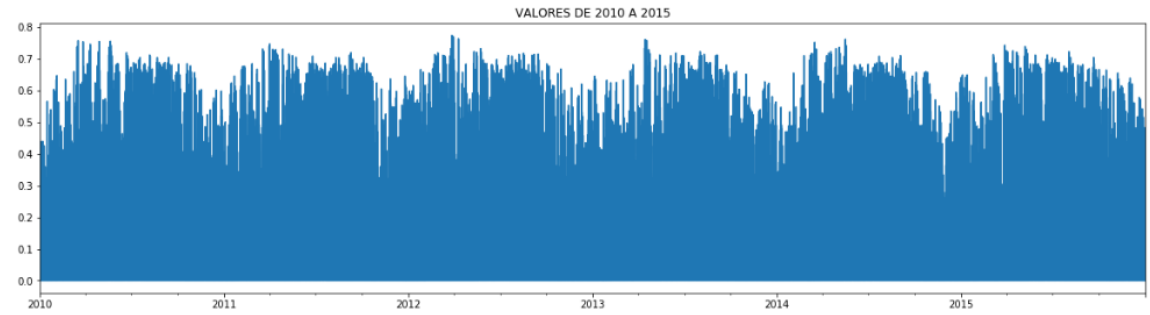

Figura 3-3: Gráfica de valores de 2010 a 2015

Ahora observaremos cómo se comportan los datos durante un mes para ver si hay estacionalidad, en este caso febrero de 2012. En un mes las gráficas son muy irregulares, por lo que no presenta ninguna estacionalidad mensual.

```
plt.figure(figsize=(20,5))
solar['2012-02'].plot(title='VALORES DE FEBRERO 2012')
plt.show()
```

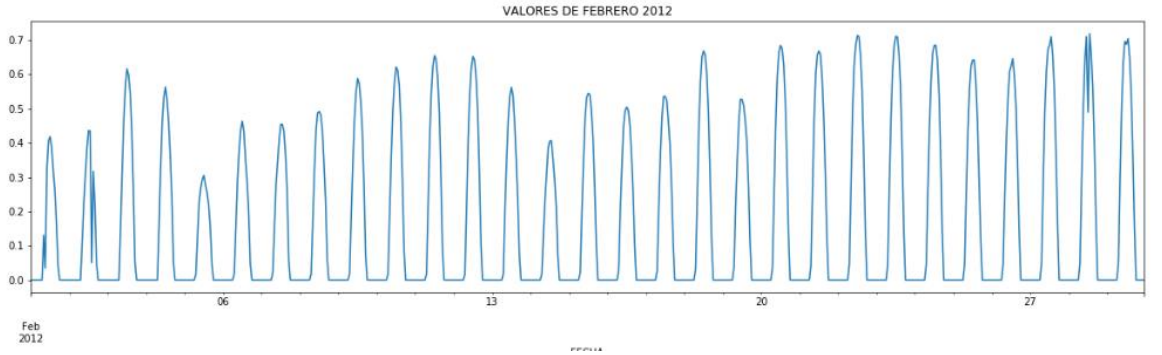

Figura 3-4: Gráfica de valores de febrero de 2012

Por último, comprobaremos si hay estacionalidad diaria, aunque con la anterior gráfica podemos hacernos a la idea de que seguramente los días tengan estacionalidad.
Esta vez pintaremos desde el 5 de mayo de 2015 hasta el 15 de mayo de 2015 y vemos la misma reacción de los datos que en el caso de los años, por ello también tendrá estacionalidad diaria.

```
plt.figure(figsize=(20,5))
solar['2015-05-05':'2015-05-15'].plot(title='VALORES DEL 5 DE MAYO 2015')
plt.show()
```

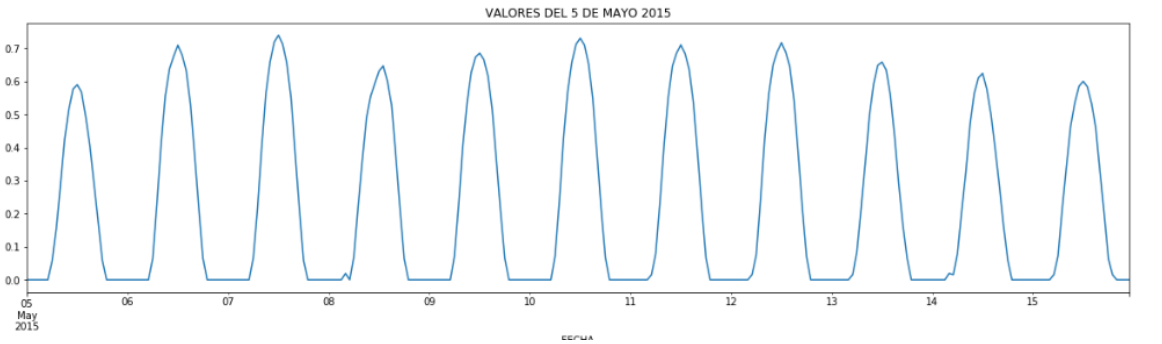

Figura 3-5: Gráfica de valores del 5 al 15 de mayo de 2015

Estas gráficas nos proporcionan mucha información sobre cómo están los datos y sus oscilaciones, aunque veamos que existe la doble estacionalidad (diaria y anual) por lo que se pintan en las gráficas, se necesitan más pruebas para determinar definitivamente que existe la doble estacionalidad.

3.2.2 Usando seasonal_descompose

En la librería *statsmodels.tsa.seasonal* tenemos una función llamada seasonal_descompose, que nos permite la descomposición estacional utilizando datos móviles. Para ello, haremos dos pruebas con diferentes frecuencias. La primera, con una frecuencia de 24 horas para determinar la estacionalidad diaria y la otra, con frecuencia 365*24 para determinar la estacionalidad anual.

La función añadiéndole la función plot() nos pinta 4 gráficas diferentes: observed, trend, seasonal y residual.

La gráfica Observed nos muestra los valores de los datos que se contienen en la variable solar, al igual que en las gráficas del apartado anterior.

La gráfica Trend nos indica la tendencia que tienen los datos a lo largo del tiempo, si van teniendo valores mayores o menores.

La gráfica Seasonal nos enseña la estacionalidad de los datos. Es la gráfica que más nos demuestra que los datos poseen estacionalidad sea diaria o anual, eso dependerá de la frecuencia.

Por último, Residual muestra los datos residuales que hay en el conjunto de datos. Estos, serán los valores que se registran por la noche ya que siempre serán con un valor de cero.

En las gráficas con frecuencia 24 se observa mucho más fácil la estacionalidad de los datos debido a que hay menos carga de datos, estos datos se cogen del año 2010 del mes de enero. Tanto en la gráfica Observed como en la Seasonal de frecuencia 24, se distingue mejor la estacionalidad diaria que en las otras dos gráficas.

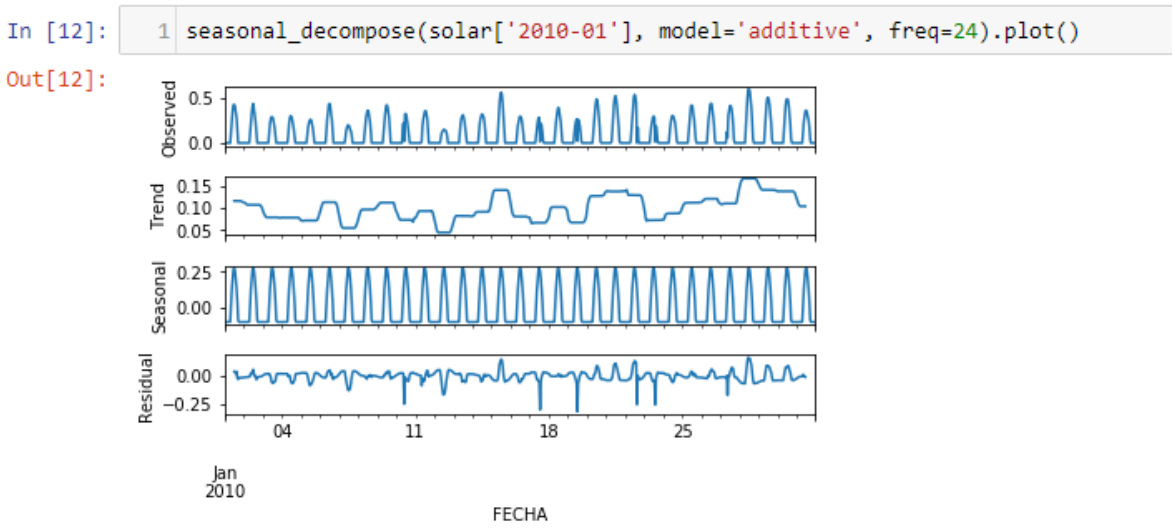

Figura 3-6: Gráfica seasonal_descompose con frecuencia 24

Las gráficas de frecuencia 24*365 entre los años 2010 y 2012 son más complicadas de ver su estacionalidad ya que poseen muchos más datos que las anteriores gráficas. Aun así, se pueden observar en las gráficas Observed y seasonal las 3 ondulaciones de los datos una por cada año, esto nos indica la estacionalidad anual de los datos. También, se puede ver en la gráfica Trend que los valores van aumentando según el trascurso de los años, lo cual tiene sentido pues cada vez hay más paneles solares instalados.

```
In [13]:  1 seasonal_decompose(solar['2010':'2012'], model='additive', freq=24*365).plot()
```

Out[13]:

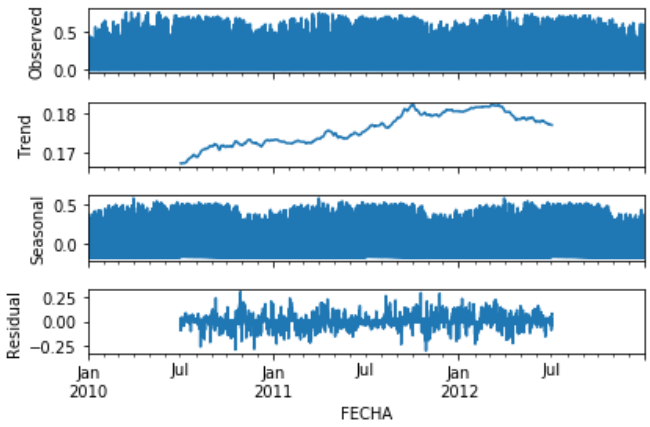

Figura 3-7: Gráfica seasonal_descompose con frecuencia 24*365

Todas las gráficas nos dan información necesaria para saber con qué datos vamos a trabajar y qué debemos hacer con ellos para una predicción más correcta.

Después de este análisis, podemos decir que los datos tienen una estacionalidad diaria y una estacionalidad anual. Por lo que deberemos quitar esa estacionalidad doble para poder trabajar con los datos y que las predicciones que realicemos se encuentren sin ninguna influencia de la estacionalidad, ya que con ella pueden salir datos erróneos o poco cercanos a la realidad de los valores que se recogerían en una fecha y hora determinada.

3.2.3 Análisis de Autocorrelación

En esta prueba usaremos la autocorrelación de datos para determinar que los valores poseen una doble estacionalidad (diaria y anual). La función que usaremos será plot_acf. Esta función tiene dos parámetros de entrada: uno de ellos, los valores con los que queremos trabajar, en este caso la variable solar; y el otro parámetro es lags, que será el que determine cuántos retrasos se mostrarán en el eje horizontal. Mientras que en el eje vertical se trazarán las correlaciones.

Primero, ejecutaremos la función para determinar la estacionalidad diaria. Para ello, usaremos un retraso superior a 24, no usaremos un valor igual a 24 ya que correlacionaría valores muy similares y no veríamos muy bien la estacionalidad. En este caso, probaremos con los valores 30, 40 y 50 para lags, y así ver cómo los valores de correlación descienden y ascienden.

Como podemos observar, en la gráfica con lags=30 la correlación empieza con valor 1.00 y va descendiendo casi hasta -0.75 cuando el valor en el eje horizontal es 12, y vuelve a ascender hasta volver a tener el valor 1.00 cuando el eje horizontal tiene el valor 24. Nuevamente, vuelven a descender los valores hasta que se llega a 30 en el eje horizontal, que es el máximo lags introducido en la función.

```
1 sm.graphics.tsa.plot_acf(solar['2010-05'], lags=30)
```

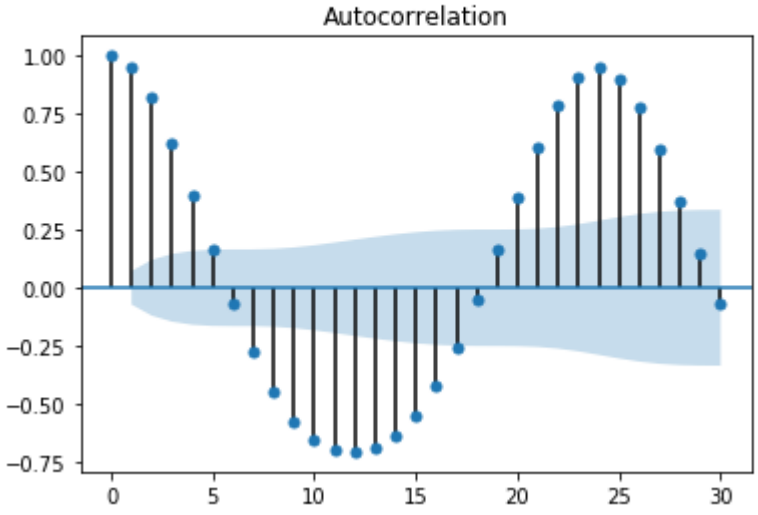

Figura 3-8: Gráfica autocorrelación con lags=30

A continuación, aumentaremos el lags máximo a 40 para que la gráfica muestre más retrasos, veamos más valores y observemos mejor las oscilaciones de la correlación. Con este aumento, se añaden 10 valores más y con ellos se ve que los valores vuelven a bajar nuevamente, se repiten como en el primer descenso y en el momento que llega a su punto mínimo empieza a ascender.

```
1 sm.graphics.tsa.plot_acf(solar['2010-05'], lags=40)
```

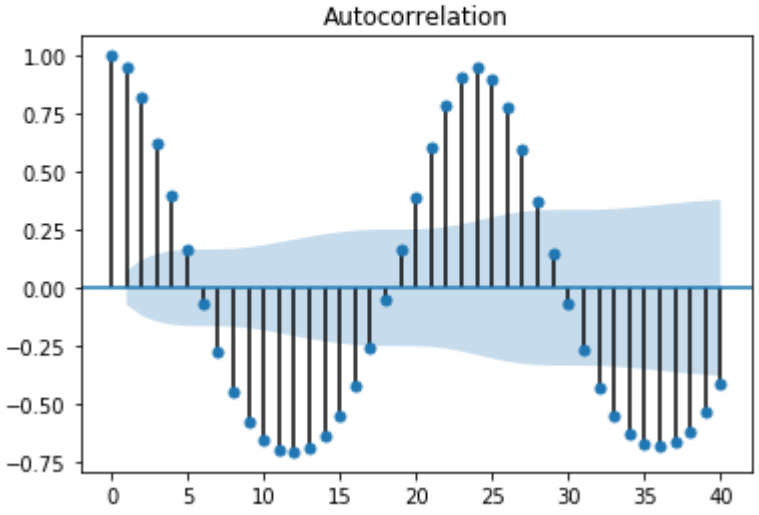

Figura 3-9: Gráfica autocorrelación con lags=40

Aumentamos, nuevamente, hasta 50 el valor máximo de lags y vemos que se repite otra vez la misma secuencia de valores en la gráfica. Además, ahora se pueden observar más

claramente las oscilaciones; ya que, tenemos tres picos superiores con valor 1.00 y otros dos picos inferiores cercanos a -0.75.

```
1 sm.graphics.tsa.plot_acf(solar['2010-05'], lags=50)
```

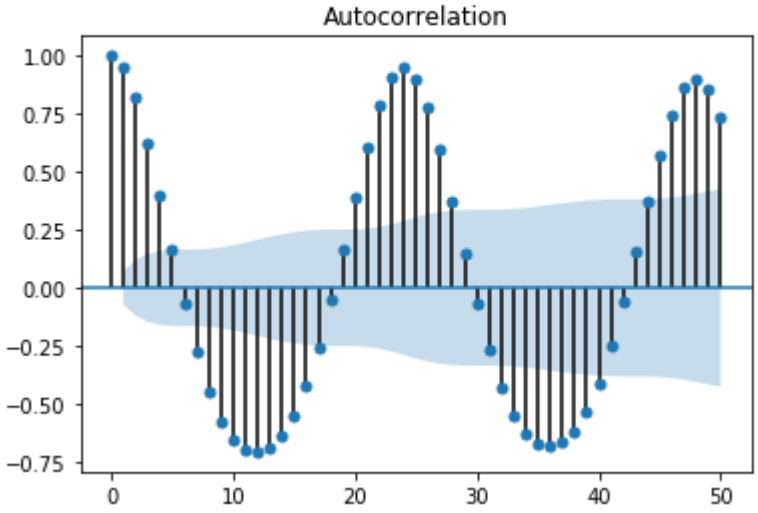

Figura 3-10: Gráfica autocorrelación con lags=50

En este punto, podemos observar que ya existe la estacionalidad diaria, empezando en un valor y durante varios retrasos vuelve al mismo valor que tenía al principio. Estas oscilaciones siguen repitiéndose a lo largo de la gráfica hasta el valor determinado en lags. Podríamos seguir aumentando el valor de lags y seguiría ocurriendo lo mismo, ascenderían hasta 1.00 y luego descenderían hasta casi -0.75 hasta el valor de lags que nosotros decidamos.

En la última gráfica, utilizaremos un lags=400 para poder ver como se autocorrelacionan los datos durante un lags muy grande y comprobaremos que sus valores seguirán siendo los mismos que en las anteriores gráficas.

```
1 x, ax = plt.subplots(figsize=(40,10))
2 plot_acf(solar, lags=400, ax=ax)
3 plt.show()
```

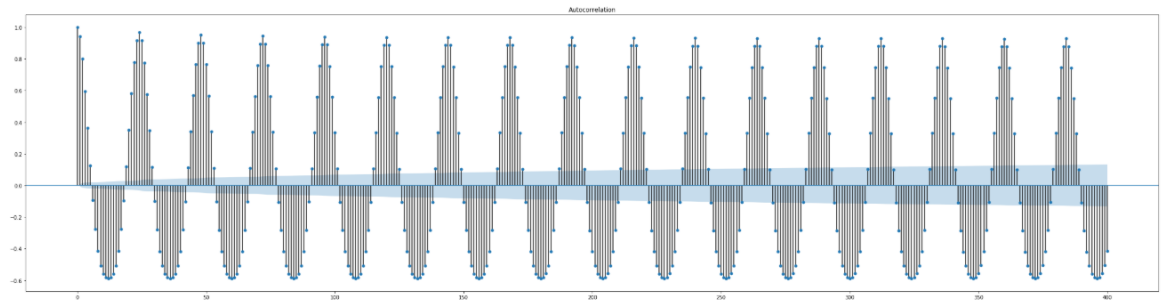

Figura 3-11: Gráfica autocorrelación con lags=400

Despues de este estudio de los datos llegamos a la conclusión que existe la estacionalidad diaria. Ahora para trabajar con ellos tendremos que quitar esa estacionalidad, que lo explicaremos y haremos en el siguiente apartado.

3.3 Tratamiento de los datos

En las anteriores secciones hemos tanto explicado de donde hemos obtenido los datos y del periodo de tiempo que comprenden, como el estudio de ellos y cómo van variando con el paso del tiempo. Ahora lo que haremos con ellos será modificarlos y trabajar con ellos para finalmente obtener un buen esquema de predicción y con ello poder predecir valores futuros que podrían dar.

3.3.1 Quitar Estacionalidad

El primer paso para el tratamiento de los datos será quitar la doble estacionalidad (diaria y anual), para ello restaremos valores unos con otros siguiendo varios criterios. Realizaremos dos restas una por la estacionalidad diaria y otra por la estacionalidad anual.

En la primera de ellas restamos un valor con el siguiente valor con veinticuatro horas más de diferencia. Realizamos esta operación con todos los valores excepto con los veinticuatro últimos ya que no tienen con quien restarse. Estos últimos veinticuatro valores no serán utilizados ya que no tienen eliminada la estacionalidad diaria.

```
1  solar_dias = solar
2  aux = solar[0]
3
4  for i in range(0, len(solar)-24):
5      solar_dias[i] = aux - solar[i+24]
6      aux = solar[i+1]
7
8  #for(s:solar):
9  #   s = solar_aux-s
10
```

Figura 3-12: Código quitando estacionalidad diaria

Para ver los cambios en el conjunto mostraremos los datos en gráfica, al igual que en el apartado 3.2.1. En la gráfica podemos observar que los valores de los datos ya no son solo positivos, y además de eso se ve que no hay tantas oscilaciones como en graficas anteriores.

```
plt.figure(figsize=(20,5))
solar_dias['2010':'2014'].plot(title='VALORES DE 2010 A 2015')
plt.show()
```

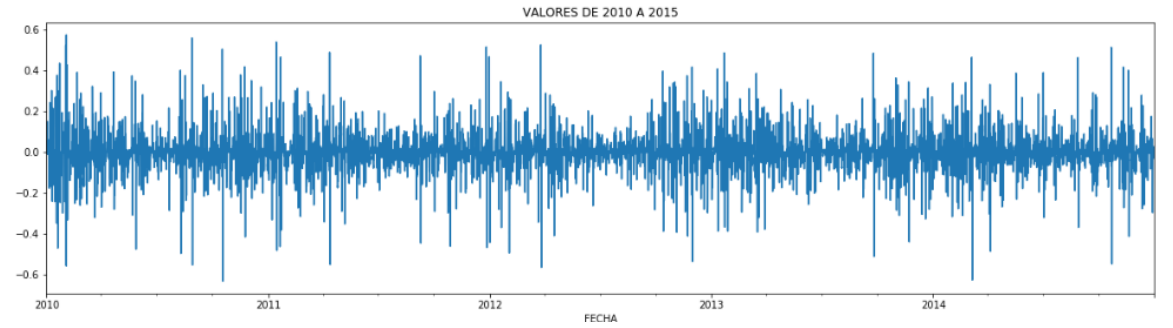

Figura 3-13: Datos sin estacionalidad diaria

En la segunda resta en vez de restar valores con veinticuatro horas de diferencia, restaremos valores con un año de diferencia, osea se, 365*24 valores de diferencia, y así quitar la estacionalidad anual. La resta la haremos sobre los resultados de la resta anterior. En el caso anterior descartamos los últimos veinticuatro valores, es decir, el último día de valores, en este caso descartaremos el último año, por no tener otro año más de datos después, por lo que el año 2015 será el que se omita.

```
solar_año = solar_dias
aux = solar_dias[0]

for i in range(0, len(solar_dias)-(365*24)):
    solar_año[i] = aux - solar_dias[i+(365*24)]
    aux = solar_dias[i+1]
```

Figura 3-14: Código quitando estacionalidad anual

Una vez más mostramos los valores en la gráfica para asegurarnos que hemos quitado esa estacionalidad. Podemos ver que los valores están mucho más cercanos al cero y no hay oscilaciones continuadas y de los mismos valores como había al principio. Los valores mostrados son hasta el año 2014 incluido, el 2015 no lo mostramos por lo explicado anteriormente.

```
plt.figure(figsize=(20,5))
solar_año['2010':'2014'].plot(title='VALORES DE 2010 A 2015')
plt.show()
```

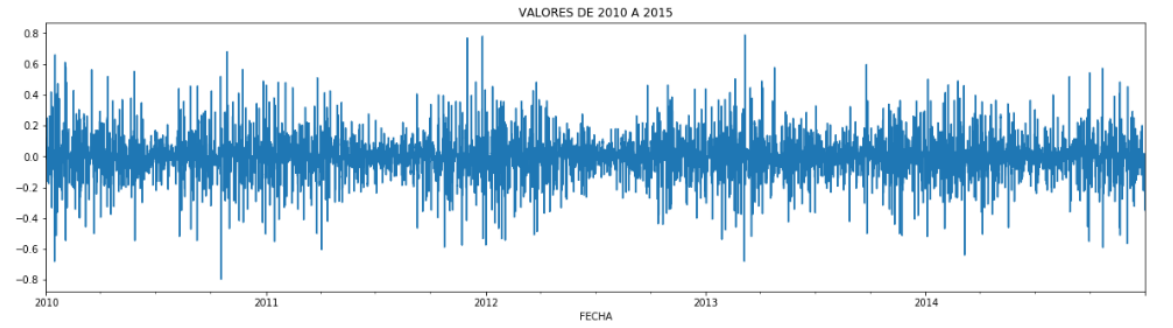

Figura 3-15: Datos sin estacionalidad anual

Como podemos observar en las gráficas, cuando quitamos la estacionalidad diaria indirectamente estamos quitando también la estacionalidad anual, por eso las gráficas son tan parecidas. Para más sencillez utilizaremos los datos quitando solo la estacionalidad diaria y se ha visto empíricamente que funciona mejor con los datos sin estacionalidad diaria, pero sin hacer la segunda diferenciación.

Con estas modificaciones en los datos ya podemos empezar a realizar nuestro trabajo de crear un método de predicción. En los siguientes apartados empezaremos a crearlo y a utilizar ciertas funciones nuevas.

3.3.2 Usando ARIMA

En este apartado, tendremos un primer contacto con el modelo ARIMA, con el cual analizaremos y pronosticaremos datos de nuestra serie temporal. La función ARIMA recibe dos valores como entrada, el primero serán los valores que hay en la serie temporal con la estacionalidad quitada, la variable solar_dias desde el año 2010 hasta el 2014 incluido. El segundo valor será el tipo de modelo que queremos probar, en este caso 5, 1, 1 para p, d y q respectivamente. Más adelante explicaremos cómo seleccionamos estos valores y cuáles seleccionamos.

Una vez ejecutada la función ARIMA, realizamos un fit sobre el resultado obtenido para ajustar los datos y después con la función summary() mostraremos las estadísticas del modelo con esos datos.

```
In [5]: arima_res = ARIMA(solar_dias['2010':'2014'], order=(5,1,1))
        res_fit=arima_res.fit(disp=0)
        print(res_fit.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                   D.ES   No. Observations:                43823
Model:                 ARIMA(5, 1, 1)   Log Likelihood               85708.090
Method:                       css-mle   S.D. of innovations              0.034
Date:                Thu, 26 Sep 2019   AIC                        -171400.179
Time:                        23:57:20   BIC                        -171330.676
Sample:                  01-01-2010   HQIC                       -171378.273
                       - 12-31-2014
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.981e-10   5.49e-08      0.011      0.991   -1.07e-07    1.08e-07
ar.L1.D.ES     0.6891      0.005    144.615      0.000       0.680       0.698
ar.L2.D.ES     0.2125      0.006     36.733      0.000       0.201       0.224
ar.L3.D.ES    -0.0082      0.006     -1.392      0.164      -0.020       0.003
ar.L4.D.ES    -0.0574      0.006     -9.922      0.000      -0.069      -0.046
ar.L5.D.ES    -0.0719      0.005    -15.096      0.000      -0.081      -0.063
ma.L1.D.ES    -1.0000   5.66e-05  -1.77e+04      0.000      -1.000      -1.000
```

Figura 3-16: Estadísticas ARIMA

En dichas estadísticas tenemos datos como Dep. Variable, que es la variable que estamos estimando, Model, el modelo que se está utilizando, Sample, dentro de los datos la fecha por la que empieza y por la que acaba.

Luego con respecto a los datos tenemos varias columnas: coef, el valor estimado del coeficiente, std err, el error estándar de la estimación del coeficiente.

Ya tenemos la función ARIMA, ahora bien, debemos determinar el modelo exacto que vamos a utilizar. Como hemos dicho antes tendremos que elegir qué valores les damos a las variables de *order (p, d, q).* Para ello usaremos la función de autocorrelación vista en el apartado 3.2.3 (ACF) y conseguir el valor de *p*. El valor de *d* será 0 o 1 y el valor de *q* está determinado por la función PACF.

En la Figura 3-17 podemos ver la gráfica de ACF y con ella determinamos que el valor para *p* sea 7 porque es un valor que está muy cerca del cero en la autocorrelación y lejos del uno. Podríamos haber elegido el 8 perfectamente también, pero queríamos un valor cercano a cero, pero no demasiado ya que afectaría a la predicción del modelo.

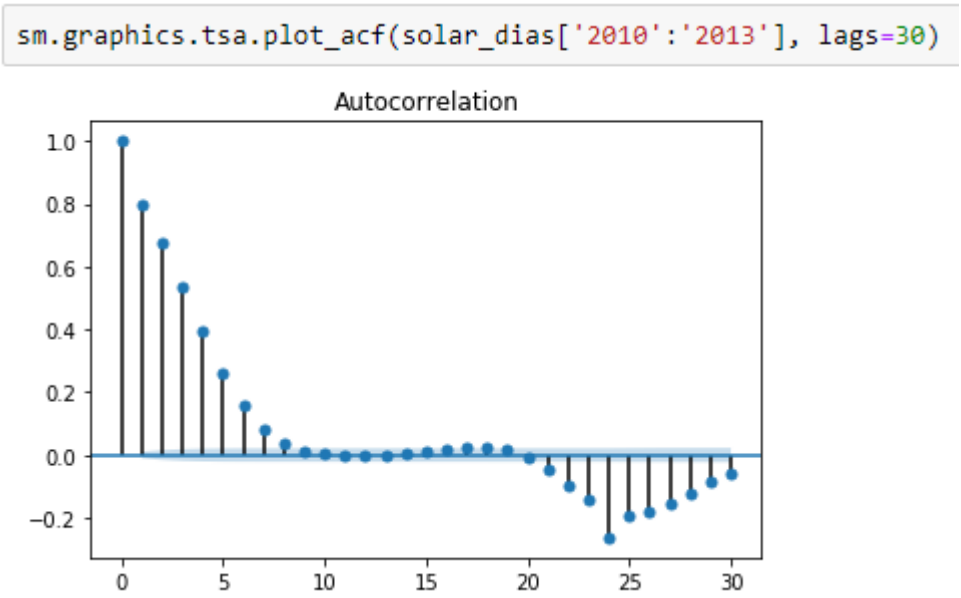

Figura 3-17: Gráfica autocorrelación con lags=30 sobre solar_dias

En la Figura 3-18 tenemos la gráfica producida por la función PACF y al igual que en el anterior caso elegiremos un valor que este lejos del uno y cerca del cero. Ese valor es el 2, el 1 no es elegido porque realiza la autocorrelación consigo mismo y el 3 tampoco porque es el último valor antes de pasar a valores negativos.

```
sm.graphics.tsa.plot_pacf(solar_dias, lags=30)
```

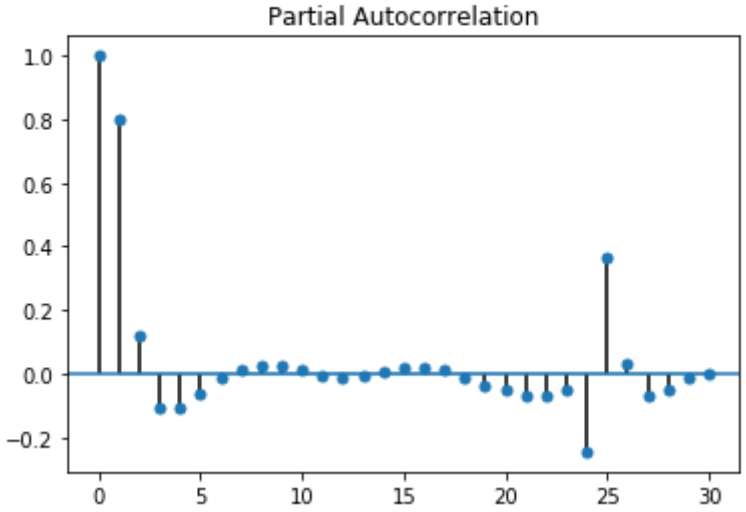

Figura 3-18: Gráfica parcial autocorrelación con lags=30 sobre solar_dias

Con la elección de estos valores ya podemos realizar las predicciones de los modelos que vamos a usar. Los modelos que estudiaremos serán AR (7), MA (2), ARMA (7, 2) y ARIMA (7, 1, 2), todos ellos utilizando también la técnica de los horizontes.

Los horizontes que vamos a realizar en la predicción serán cinco, es decir, predeciremos los valores de las siguientes cinco horas. A continuación, introduciremos el valor del horizonte uno en nuestros datos de aprendizaje y predeciremos otros cinco más, continuaremos así durante todo un año.

Con esto tendremos por cada modelo la predicción de un año por cinco horizontes, con esto haremos la media de error por cada horizonte para descubrir qué modelo y horizonte predice de forma más efectiva. Los resultados de ello los veremos en el próximo apartado.

4 Integración, pruebas y resultados

Después de recoger todas las predicciones de cada modelo con los cinco horizontes, realizamos la media de error de ellos y obtenemos la Tabla 4-1. Ahora explicaremos en detalle estos resultados y el porqué de estos.

	H+1	H+2	H+3	H+4	H+5
AR (7)	0.00989	0.00571	0.00711	0.00597	**0.00445**
MA (2)	**0.00242**	0.00360	0.00960	0.00964	0.00966
ARMA (7,2)	**0.00100**	0.00147	0.00232	0.00244	0.00270
ARIMA (7,1,2)	0.00410	0.00366	0.00352	0.00343	**0.00328**

Tabla 4-1: Medias de error de cada modelo y horizonte

En ella podemos observar que el modelo ARMA con horizonte uno es el modelo con menos media de error y se acerca más al dato real. ¿Por qué es el mejor?

ARIMA no sería el mejor debido a que cómo ya hemos quitado la estacionalidad de los datos y este modelo funciona con estacionalidad, no es el modelo más adecuado. Los otros dos modelos, AR y MA, no serán mejores debido a que ambos combinados hacen el modelo ARMA. Por estas razones el modelo ARMA es el más adecuado.

Si vemos las medias de error, también se puede apreciar que, tanto en MA como en ARMA, a partir del horizonte tres son malas medias y que poseen valores muy similares unos horizontes con otros dentro del mismo modelo.

Según lo estudiado anteriormente, cuando aumentáramos de horizonte aumentaría el error. En MA y ARMA vemos que se da correctamente, pero en el caso de AR y ARIMA no. ¿Por qué? Vamos a analizar sus predicciones.

Analizaremos las predicciones de AR con sus horizontes para ver qué ocurre y por qué no siguen este criterio.

Primero veremos cómo se comportan sus predicciones en un par de días. En este caso, los días 2 y 3 de enero de 2014, cómo podemos ver en la Figura 4-1.

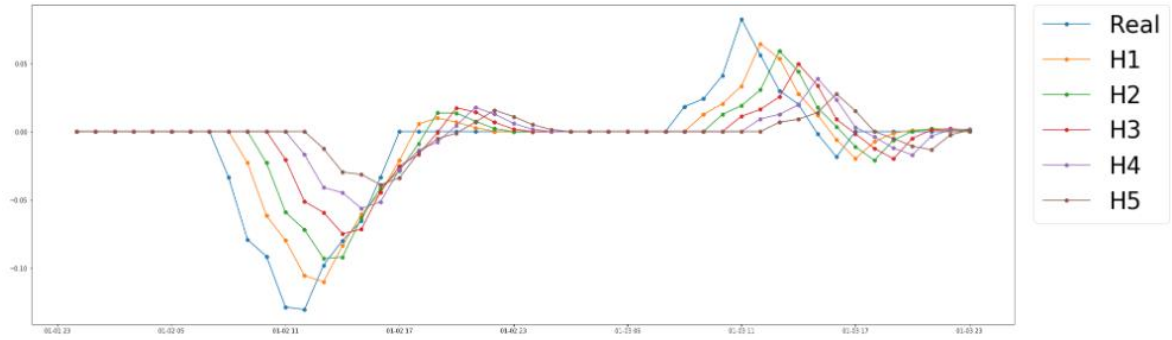

Figura 4-1: Gráfica modelo AR (7) y sus horizontes durante dos días

Los valores que se muestran en la gráfica parecen razonables, aunque se aprecia un desplazamiento que más adelante comentaremos porqué ocurre. Las predicciones de estos dos días no presentan ningún índice del porqué de las medias de error. Vamos a realizar la media de error sobre estos días para confirmarlo.

MODELO	H+1	H+2	H+3	H+4	H+5
AR (7)	**0.00029**	0.00065	0.00103	0.00140	0.00163

Tabla 4-2: Medias de error de AR por horizontes

Como vemos en la Tabla 4-2, el modelo se comporta como se espera, el horizonte uno es el que menor media de error tiene y según se avanza de horizonte el error aumenta.

Probaremos a mostrar todas las predicciones del modelo con sus horizontes y así quizás obtengamos el porqué de esas medias de error. La Figura 4-2 aunque un poco amontonada muestra lo necesario para descubrir el problema.

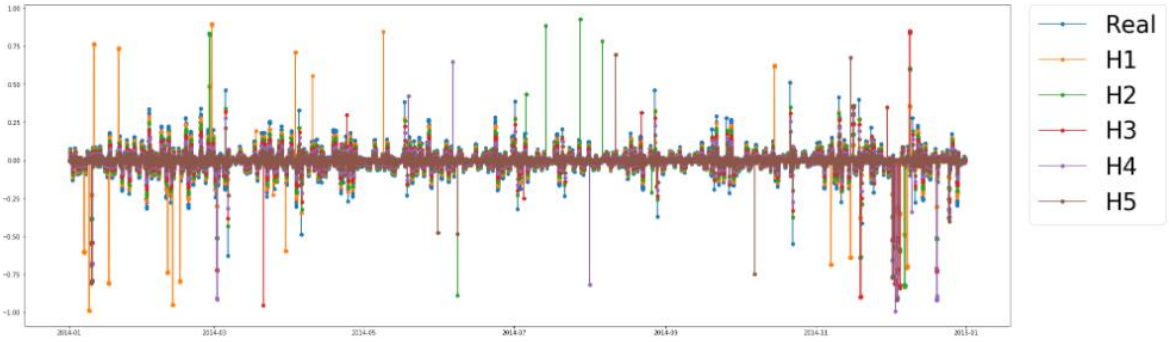

Figura 4-2: Gráfica modelo AR (7) y sus horizontes durante un año

Como hemos dicho antes, el horizonte uno debería ser el que tenga menor media de error, pero en la Figura 4-2 vemos que el horizonte uno posee unos picos en los valores que desestabilizan la media de error. En un trabajo futuro se podría determinar porque salen esos picos y además quitarlos del error.

Este fenómeno de los picos también se produce en modelo ARIMA de la misma forma, por ello ambos modelos su media de error mínima es la del horizonte cinco.

Ahora realizaremos una comparación de modelos por horizonte y lo haremos sobre 2 días para ver detalles. Estos días serán los mismos que la anterior vez, 2 y 3 de enero de 2014.

Para conseguir esto mostraremos los datos reales comparándolos con cada horizonte para todos los modelos y así a la vez comparar unos modelos con otros.

Empezaremos con el horizonte uno como en la Figura 4-3, y vemos que las predicciones realizan un dibujo muy similar excepto ARIMA por lo anterior dicho. También vemos un ligero desplazamiento de las predicciones hacia la derecha.

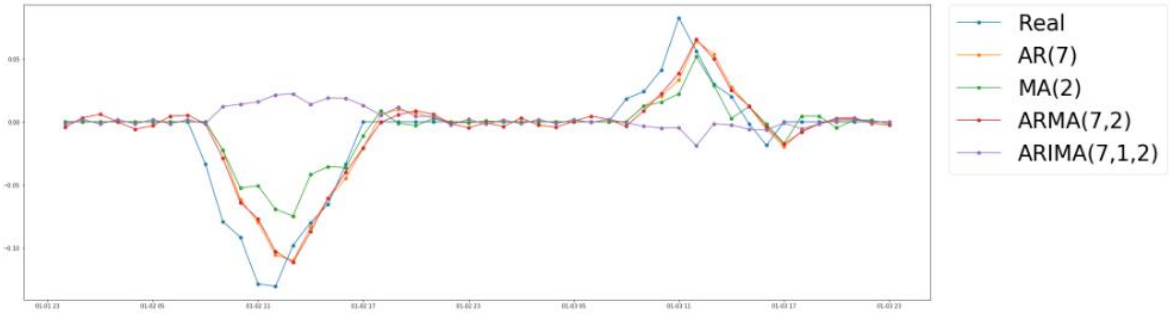

Figura 4-3: Gráfica horizonte uno para los modelos

El siguiente horizonte será el dos (Figura 4-4) y podemos ver que los valores empiezan a tener mayor distancia con los reales tanto vertical como horizontal.

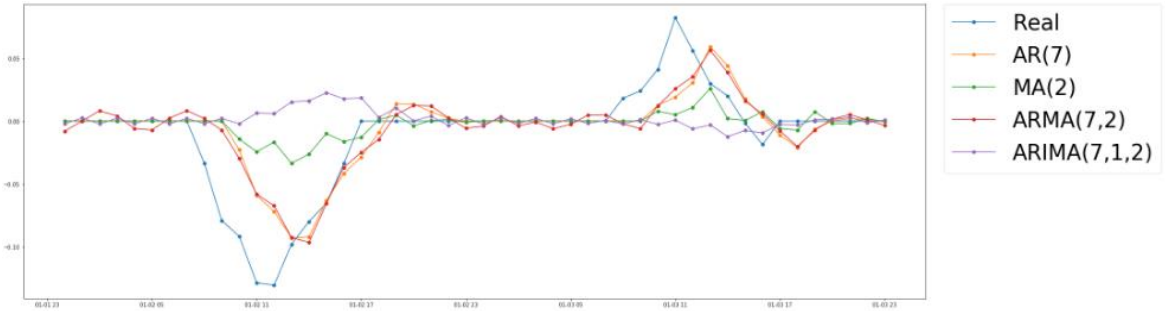

Figura 4-4: Gráfica horizonte dos para los modelos

Continuamos con el horizonte tres en la Figura 4-5, en el cual se vuelve a detectar que los valores aumentan su distancia con los reales en vertical y horizontal.

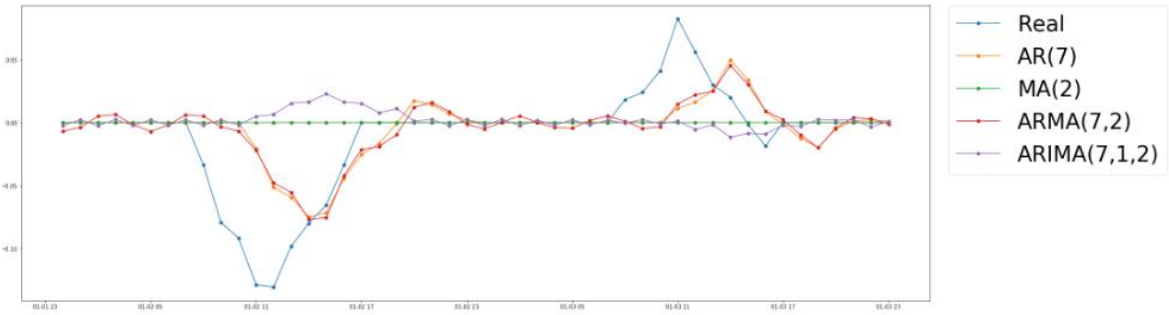

Figura 4-5: Gráfica horizonte tres para los modelos

Ahora tenemos en la Figura 4-6 el horizonte cuatro, seguimos viendo en ella que las predicciones están cada vez más cerca de cero y también vemos otro desplazamiento hacia la derecha.

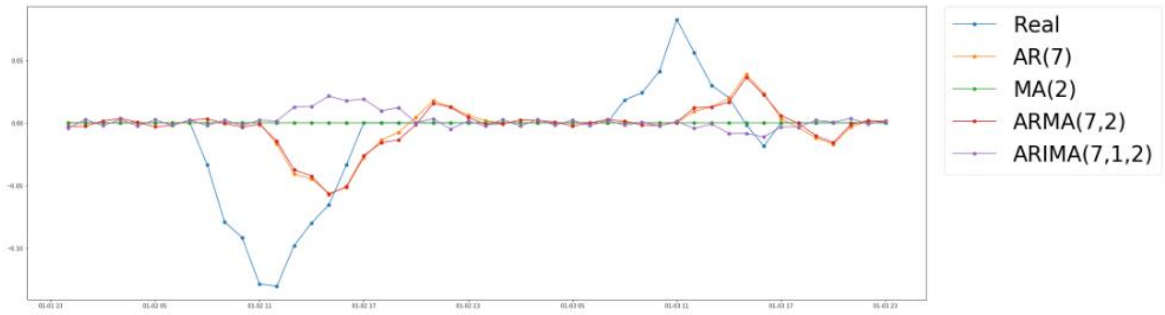

Figura 4-6: Gráfica horizonte cuatro para los modelos

Por último, veremos el horizonte cinco en la Figura 4-7, sus valores están ya muy lejos de los reales y casi no tienen ni la misma forma, tanto por el desplazamiento vertical como por el horizontal.

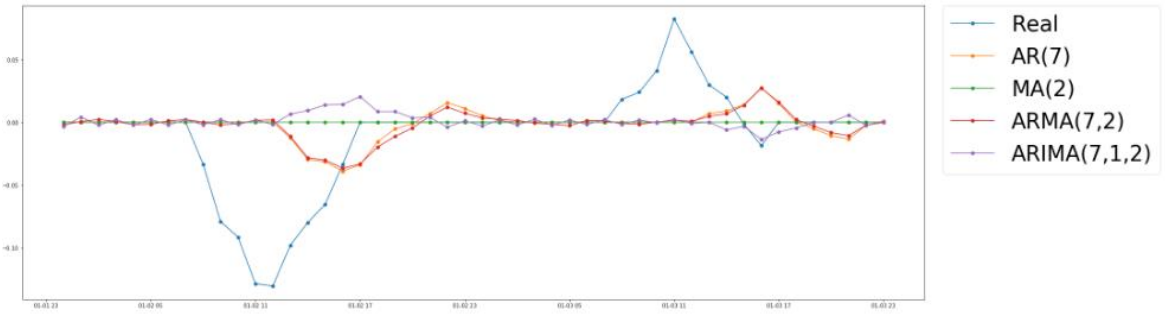

Figura 4-7: Gráfica horizonte cinco para los modelos

Con todas estas comparativas podemos determinar varios detalles acerca de los modelos y los horizontes:

1. Confirmamos que ARMA parece el mejor modelo para predecir ya que se acerca bastante a los datos reales, muy pegado a este modelo está también AR, aunque globalmente su comportamiento no sea el adecuado.

2. Otro detalle, es que según va aumentando el horizonte la predicción es peor y se acerca a la media, que está muy cerca del cero.

3. Por último, como ya hemos ido comentando se aprecia un desplazamiento horizontal en las predicciones según aumenta el horizonte. Esto es producido porque los modelos autorregresivos hacen la media ponderada de los valores de energía solar inmediatamente anteriores. Esto es un efecto muy similar a una persistencia, que es repetir el valor justo anterior, en trabajos futuros se podría intentar evitar.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

Para finalizar la investigación de los modelos autorregresivos y según las pruebas y resultados obtenidos, podemos concluir que el modelo ARMA con horizonte uno es el mejor modelo a usar para la predicción de energía solar obtenida en una determinada zona.

Los otros modelos como hemos explicado anteriormente poseen ciertos problemas en sus predicciones y no son adecuados. En el caso de ARIMA porque es un modelo que realiza predicciones de series temporales con tendencias y nuestra serie temporal al no tener, sus resultados no tienen sentido. En AR sus predicciones en ciertos momentos tenían picos que producían una desestabilización de la media de error. El modelo MA se comportaba correctamente pero no era tan cercano a los datos reales como ARMA. Además, tiene lógica que ARMA sea mejor que AR y que MA, debido a que ARMA es la combinación de ambos modelos.

Respecto a los horizontes exceptuando los casos de AR y de ARIMA por sus picos y el tipo de modelo, en el caso de ARIMA, se comportan según lo previsto cuanto mayor es el horizonte mayor será la media de error, ya que el valor a predecir esta cada vez mas lejos del ultimo valor obtenido.

5.2 Trabajo futuro

La continuación de esta investigación podría ser arreglar el desplazamiento horizontal que se produce en las predicciones. Esto se empezaría optimizando los parámetros cogiendo p y q más grandes (que cojan más de un día) para que el modelo autorregresivo vea el comportamiento general de la curva y no las horas, que muchas veces son solo noche. Lo siguiente seria quitar los valores de la noche que son todo ceros y así los valores no se acercarían a cero y la media de error será menor. Estos valores de la noche no es necesario predecirlos pues siempre son ceros y se pueden dejar fijos a este valor. También para su mejora se le puede añadir los valores de radiación solar como variable externa, enriqueciendo el modelo.

Otro tema que tratar seria averiguar porque se producen esos picos en el horizonte 1 de AR. Como en el anterior caso quitando los valores de la noche quizá pueda mejorar la predicción ya que esos datos son repetitivos y pueden afectar a la predicción.

Referencias

[1] Dane, S. “30 Years of European Solar Generation”, Kaggle, Obtenido de: https://www.kaggle.com/sohier/30-years-of-european-solar-generation

[2] Rouse, M. “Aprendizaje automático (machine learning)”, TechTarget, Obtenido de: https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-automatico-machine-learning

[3] Moreno, A. (1994). Aprendizaje automático, 1-5.

[4] Nguyen Cong, B., Rivero Pérez, J. L., & Morell, C. (2015). Aprendizaje supervisado de funciones de distancia: estado del arte. Revista Cubana de Ciencias Informáticas, 9(2), 14-28.

[5] Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. Encyclopedia of database systems, 532-538.

[6] APLICADAS, L. E. M. (2012). Aprendizaje no supervisado y el algoritmo wake-sleep en redes neuronales (Doctoral dissertation, UNIVERSIDAD TECNOLOGICA DE LA MIXTECA), 27-41

[7] Abney, S. (2007). Semisupervised learning for computational linguistics. Chapman and Hall/CRC.

[8] Boada, M. J. L., Boada, B. L., & López, V. D. (2005). Algoritmo de aprendizaje por refuerzo continuo para el control de un sistema de suspensión semi-activa. Revista Iberoamericana de Ingeniería Mecánica, 9(2), 77.

[9] Ávila, E., Garay, P., Grefa, V., Montúfar, E., & Revelo, M. Diseño y Construcción de un Robot Móvil que Aprenda a Detectar Diferentes Señales de Tránsito Mediante Inteligencia Artificial.

[10] Crespo, A. B. (2013). Aprendizaje máquina multitarea mediante edición de datos y algoritmos de aprendizaje extremo (Doctoral dissertation, Universidad Politécnica de Cartagena).

[11] Casimiro, M. P. G. (2009). Análisis de series temporales: Modelos ARIMA. Lejona, España: Universidad del País Vasco.

[12] Brockwell, P. J., & Davis, R. A. (2016). Introduction to time series and forecasting. Springer.

[13] McCullagh, P. (2002). What is a statistical model?. Annals of statistics, 1225-1267.

[14] Nochai, R., & Nochai, T. (2006, June). ARIMA model for forecasting oil palm price. In Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and applications (pp. 13-15).

Glosario

AR	AutoRegressive
MA	Moving Average
ARMA	AutoRegressive Moving Average
ARIMA	AutoRegressive Integrated Moving Average
ACF	AutoCorrelation Function
PACF	Partial AutoCorrelation Function