



Ensemble Learning Assessment

MACHINE LEARNING II

GRUPO 1

Grupo formado por:

Álvaro Ezquerro Pérez
María Calvo de Mora Román
Celia Quiles Alemañ

ÍNDICE

1.	INTRODUCCIÓN.....	2
2.	ANÁLISIS EXPLORATORIO DE DATOS	2
2.1.	INTRODUCCIÓN A LOS DATOS	2
2.2.	DISTRIBUCIÓN DE LAS VARIABLES	3
2.3.	RELACIÓN ENTRE LAS VARIABLES.....	11
3.	PREPARACIÓN DE LOS DATOS Y FEATURE ENGINEERING	12
3.1.	TARGET DE NUESTRO PROBLEMA:	12
3.2.	VARIABLES EXPLICATIVAS:.....	13
3.3.	AJUSTES FINALES:	14
4.	MODELOS DE REGRESIÓN.....	14
4.1.	REGRESIÓN LINEAL:	14
4.2.	ÁRBOL DE REGRESIÓN SENCILLO	17
4.3.	BAGGED TREE	19
4.4.	RANDOM FOREST.....	21
4.5.	GRADIENTBOOSTING	24
4.5.1.	<i>Prueba 1: con todas las variables explicativas.....</i>	24
4.5.2.	<i>Prueba 2: Solo variables resultantes como significativas del modelo anterior.....</i>	26
5.	CONCLUSIONES	28
5.1.	COMPROBACIÓN FINAL.....	32
6.	RESULTADOS INTERMEDIOS NO ACEPTABLES.....	33

1. INTRODUCCIÓN

El objetivo de esta práctica es realizar un trabajo de Machine Learning en el que, a partir de datos relacionados con la irradiación solar y las utilizaciones solares fotovoltaicas, datos que serán previamente explorados y descritos, estimaremos las utilizaciones horarias de un día.

Esto se realizará utilizando técnicas de ensamblado, así como se compararán los resultados con los que obtendríamos con alguna técnica más directa.

2. ANÁLISIS EXPLORATORIO DE DATOS

El Análisis Exploratorio de Datos (EDA) es una fase crucial en cualquier proyecto de ciencia de datos o aprendizaje automático. Consiste en explorar, entender y visualizar los datos disponibles antes de aplicar cualquier modelo predictivo. En el contexto de la predicción de utilizaciones solares fotovoltaicas, el EDA desempeña un papel fundamental debido a la complejidad inherente de los datos solares y la necesidad de comprender completamente su comportamiento y relaciones.

2.1. Introducción a los datos

En primer lugar, entendamos qué variables tenemos y qué significan.

En concreto, en este trabajo contamos con 2 datasets, uno relativo a las irradiaciones solares, y otro sobre las utilizaciones solares fotovoltaicas.

La irradiación solar se refiere a la cantidad de radiación solar que llega a una superficie determinada en un período de tiempo específico, mientras que la utilización solar fotovoltaica se refiere a la cantidad de energía eléctrica generada por un sistema fotovoltaico en ese mismo período de tiempo.

Veamos primero el dataset de las irradiaciones:

```
: 1 df_orig_Irrad.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2192 entries, 0 to 2191
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   FECHA       2192 non-null    datetime64[ns]
 1   IRRADH00    2192 non-null    float64
 2   IRRADH03    2192 non-null    float64
 3   IRRADH06    2192 non-null    float64
 4   IRRADH09    2192 non-null    float64
 5   IRRADH12    2192 non-null    float64
 6   IRRADH15    2192 non-null    float64
 7   IRRADH18    2192 non-null    float64
 8   IRRADH21    2192 non-null    float64
 9   ANNO        2192 non-null    int64  
 10  MES         2192 non-null    int64  
 11  DIA         2192 non-null    int64  
 12  DIASEM     2192 non-null    int64
```

Ilustración 1. Información inicial sobre el dataset de irradaciones.

Tal y como se observa en Ilustración 1, este dataset cuenta con 12 variables.

- Por un lado, tenemos las 5 variables relacionadas con la fecha del registro en cuestión: FECHA (día, mes y año del registro), ANNO, MES, DIA y DIASEM. Estas variables nos permitirán establecer una especie de relación temporal en los registros, sin llegar a usar ningún método de forecasting ni de series temporales. Todas ellas son números enteros, salvo FECHA que es de tipo datetime. Por tanto, en este sentido, tenemos datos diarios desde enero del 2015 hasta diciembre del 2020.
- Las 8 columnas restantes, son las columnas relativas a las irradiaciones solares del día en cuestión. Tenemos valores cada 3 horas (00, 03, 06...). Esto se debe a que la irradiación solar del tramo horario h es la irradiación acumulada de las horas h, h+1 y h+2.

Además, se observa cómo no existen valores nulos en el dataset.

Seguimos con el dataset de las utilizaciones:

1 df_orig_Util.info()				
<class 'pandas.core.frame.DataFrame'>				
RangeIndex: 2192 entries, 0 to 2191				
Data columns (total 13 columns):				
#	Column	Non-Null Count	Dtype	
0	FECHA	2192	non-null	datetime64[ns]
1	UTILH00	2192	non-null	float64
2	UTILH03	2192	non-null	float64
3	UTILH06	2192	non-null	float64
4	UTILH09	2192	non-null	float64
5	UTILH12	2192	non-null	float64
6	UTILH15	2192	non-null	float64
7	UTILH18	2192	non-null	float64
8	UTILH21	2192	non-null	float64
9	ANNO	2192	non-null	int64
10	MES	2192	non-null	int64
11	DIA	2192	non-null	int64
12	DIASEM	2192	non-null	int64

Ilustración 2. Información inicial sobre el dataset de utilizaciones.

En este caso, en la Ilustración 2 se observa como este dataset vuelve a contar con 12 variables, muy similares a las anteriores.

- Por un lado, tenemos las 5 variables relacionadas con la fecha del registro en cuestión: FECHA (día, mes y año del registro), ANNO, MES, DIA y DIASEM. Estas variables nos permitirán establecer una especie de relación temporal en los registros, sin llegar a usar ningún método de forecasting ni de series temporales. Todas ellas son números enteros, salvo FECHA que es de tipo datetime. De nuevo, tenemos datos diarios desde enero del 2015 hasta diciembre del 2020.
- Las 8 columnas restantes, son las columnas relativas a las utilizaciones solares fotovoltaicas del día en cuestión. Volvemos a tener valores cada 3 horas (00, 03, 06...). Esto se debe a que la utilización del tramo horario h es la utilización media de las horas h, h+1 y h+2.

De nuevo, se observa como este dataset tampoco tiene valores ausentes.

2.2. Distribución de las variables

Dataset Irradiaciones:

A continuación, veamos qué comportamiento tienen cada una de las irradiaciones y utilizaciones. Para eso, vamos a representar su evolución temporal con gráficos de líneas.

Para comenzar, en la Ilustración 3 se representan todas las irradiaciones solares en un mismo gráfico, diferenciando con el color los distintos tramos horarios.

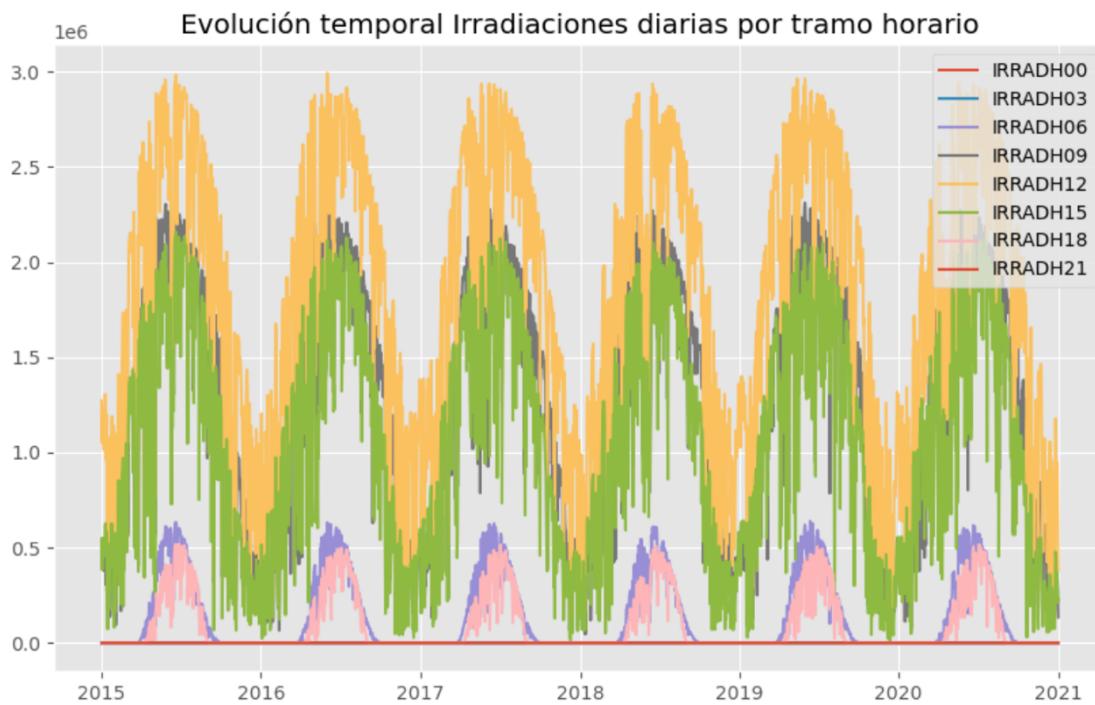


Ilustración 3. Evolución temporal de las irradiaciones diarias por tramo horario.

En este gráfico de radiación solar (Ilustración 3), es decir, cuánto sol hay a lo largo de cada día y hora, podemos ver los siguientes resultados, los cuales eran fáciles de prever:

- Cuando se registran los **valores más superiores independientemente del mes de año, es durante el tramo horario de las 12** (radiación acumulada a las 12, 13 y 14 horas, es decir, el mediodía).
- A este tramo **le siguen el de las 9am (9, 10 y 11 horas) y el de las 15 (15, 16 y 17h)**.

Todo esto ocurre en todas las épocas del año, pues son horas del día que, **independientemente de si el día es más largo o más corto, a estas horas es de día igualmente**, y debido a tener el sol al norte, con ninguna o ligera inclinación, la potencia de la radiación es mayor. No obstante, ambas 3 series sí **tienen un patrón estacional anual también**. En invierno, por ejemplo, al haber menos días soleados, la radiación solar es mucho más baja que en los meses de verano, donde alcanzamos los picos.

- **Los 2 siguientes tramos son los correspondientes a las 06 (06, 07 y 08 horas, am) y el de las 18 (18, 19 y 20 horas)**. Estos dos tramos, además de que la radiación recibida es mucho menor pues, aunque aún haga luz, debido a la inclinación la potencia de la **radiación es menor**, además, solo registran radiación en los meses de verano, final

de primavera y principios de otoño. Esto se debe a que, **en invierno, no hay luz a estas horas del día.**

- Finalmente, **en los tramos de las 21 horas, 00h y 03h, al ser de noche durante todos los meses del año, la irradiación siempre es 0.**

De igual manera, se puede también realizar una descomposición de las series temporales con el método *seasonal_decompose* de *statsmodels* y extraer la componente estacional de cada tramo para verlo de manera más clara. Esto se ve representado en la Ilustración 4.



Ilustración 4. Descomposición estacional de irradiaciones diarias por tramo horario.

- En dicha Ilustración 4 se aprecia cómo los valores para los tramos de la noche son planos.
- Para los tramos de las 06h y 18h solo hay irradiación (ligera irradiación) desde mayo hasta septiembre.
- El resto de los tramos horarios registran más volatilidad durante el año, alcanzando los valores máximos en agosto (y en verano en general), mientras que los valores mínimos son en diciembre, justamente cuando los días son más cortos.

Para terminar el análisis de las irradiaciones, vamos a extraer en esta ocasión la tendencia / Nivel de la serie, que nos va a permitir extraer 2 conclusiones: 1) en torno a qué valores se mueve cada tramo y 2) si ha habido una tendencia clara en algún tramo durante estos años.

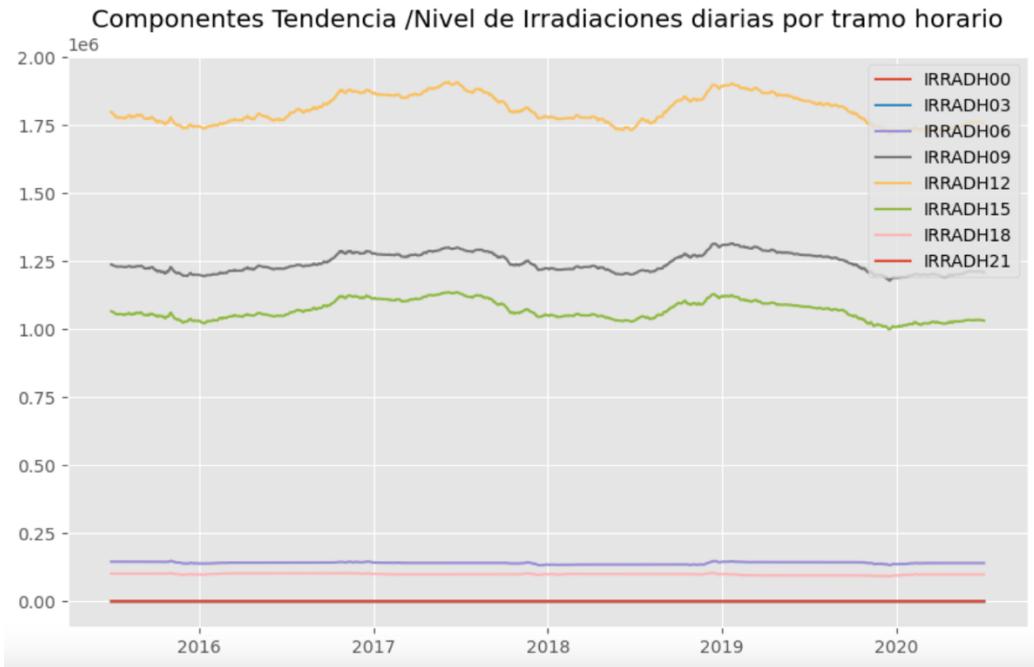


Ilustración 5. Descomposición de tendencia de irradiaciones diarias por tramo horario.

Observando la Ilustración 5, ninguno de los tramos ha sufrido una tendencia clara ni positiva ni negativa, como es normal. Es decir, la irradiación durante los años ha sido constante en cada uno de los diferentes tramos horarios.

Sin embargo, la Ilustración 5 sí nos permite ver de manera más clara la diferencia en los valores promedios de cada tramo: cuando el valor es más elevado es en el tramo del mediodía, el de las 12h (12h + 13h + 14h), seguido del de las 9h y 15h, como ya adelantábamos anteriormente.

Por último, a continuación, se representan en la Ilustración 6 los boxplots de cada tramo horario, para identificar valores atípicos por tramo horario.

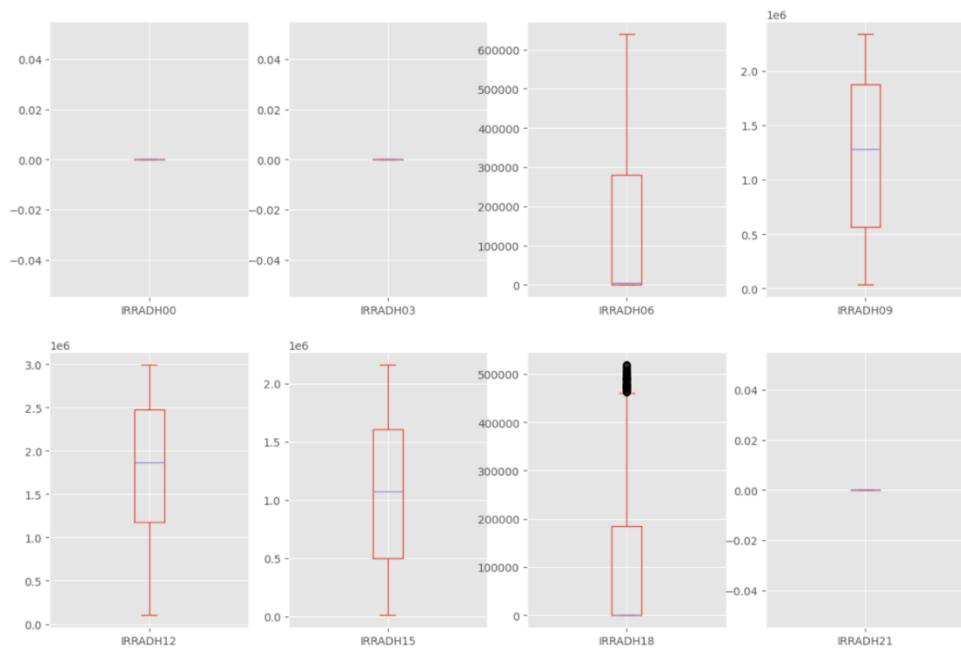


Ilustración 6. Distribuciones representadas mediante boxplots de los tramos horarios de las irradiaciones.

Los tramos de las 00h, 03h y 21h al ser una serie de 0s, no tienen ninguna variabilidad.

Los tramos centrales del día, pese a tener cierta variabilidad a lo largo del año en función de si es invierno, verano, etc. No presentan tampoco valores atípicos pues son tramos horarios donde todos los días hay cierta irradiación, sea más o menos.

No obstante, es en el tramo de las 18h donde se observan bastantes outliers debido a poseer valores elevados de irradiación. Estos serán justamente los valores registrados en junio-julio que es cuando esta serie registra anualmente sus valores máximos.

Dataset Utilizaciones:

De igual manera que antes, para comenzar con este segundo dataset, se representan a continuación todas las utilizaciones solares fotovoltaicas en un mismo gráfico (Ilustración 7), diferenciando con el color entre los distintos tramos horarios.

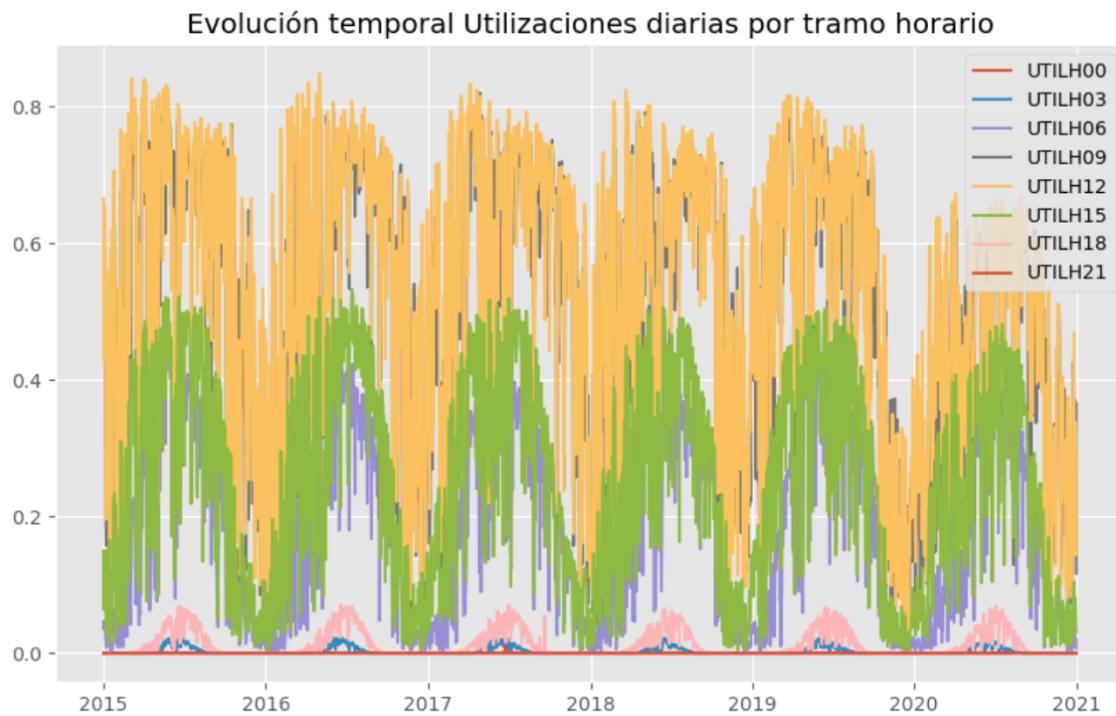


Ilustración 7. Evolución temporal de las utilizaciones diarias por tramo horario.

En este gráfico de utilización solar (Ilustración 7), es decir, qué cantidad de energía eléctrica se puede generar por un sistema fotovoltaico en un período de tiempo, lo cual dependerá principalmente de cuánta irradiación solar haya a lo largo de cada día y hora. Estudiemos los patrones de este dataset:

- Cuando se registran los **valores más superiores independientemente del mes de año, es durante el tramo horario de las 12** (irradiación acumulada a las 12, 13 y 14 horas, es decir, el mediodía. Muy seguido de los tramos de las 9am (9h, 10h y 11h).

- A este tramo le siguen el de las 15 (15, 16 y 17h) y el de las 06am (06h, 07h y 08h).

Todo esto ocurre en todas las épocas del año, pues son horas del día que, **independientemente de si el día es más largo o más corto, a estas horas es de día igualmente**, y debido a tener el sol al norte, con ninguna o ligera inclinación, la potencia de la irradiación es mayor y la utilización aumenta en consecuencia. No obstante, ambas 4 series sí **tienen un patrón estacional anual también**. En invierno, por ejemplo, al haber menos días soleados, la utilización solar es mucho más baja que en los meses de verano, donde se alcanzan los picos.

Se destacan las diferencias respecto a los patrones de irradiación estudiados antes, ahora a las 09h los valores son muy superiores a las 15h, mientras que antes eran valores similares. Esto se debe a que, las irradiaciones se calculan como un acumulado de las 3h, mientras que las utilizaciones son una media.

- Los 2 siguientes tramos son los correspondientes a las 03 (03, 04 y 05 horas, am) y el de las 18 (18, 19 y 20 horas). Estos dos tramos solo registran utilizaciones en los meses de verano, final de primavera y principios de otoño. Esto se debe a que, **en invierno, no hay luz a estas horas del día**.
- Finalmente, **en los tramos de las 21 horas y 00h, al ser de noche durante todos los meses del año, la irradiación y la utilización siempre es 0**.

Siguiendo con la metodología empleada con el dataset de irradiaciones, se realiza también una descomposición de las series temporales con el método *seasonal_decompose* de *statsmodels*, extrayendo la componente estacional de cada tramo para verlo de manera más clara.

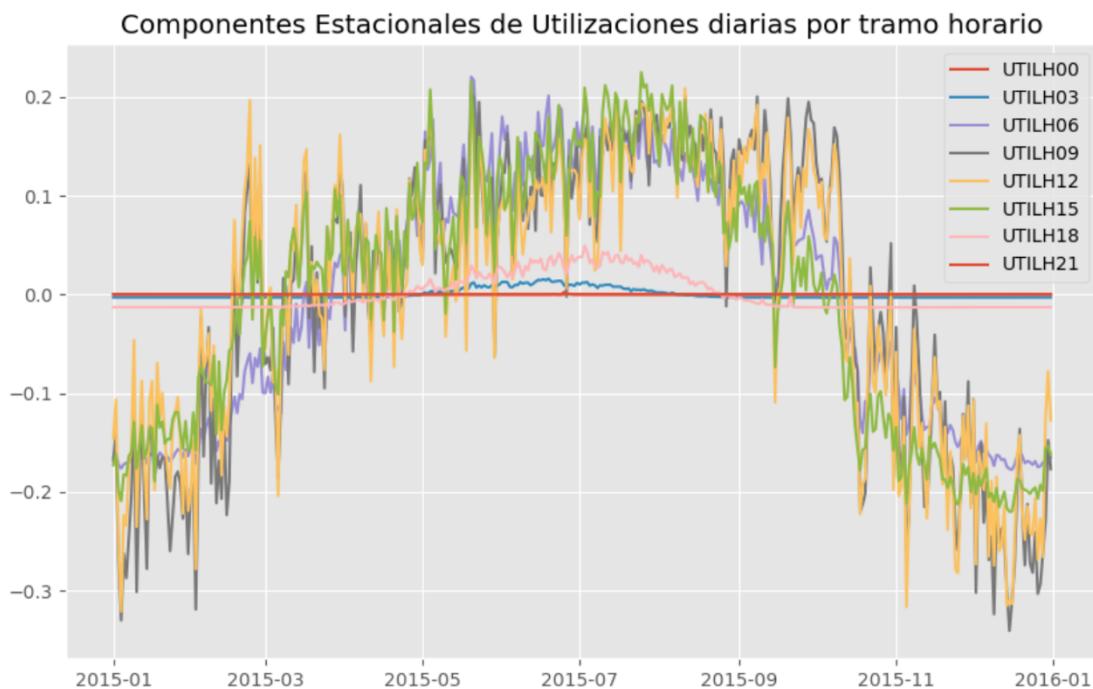


Ilustración 8. Descomposición estacional de utilizaciones diarias por tramo horario

- En Ilustración 8 se aprecia cómo los valores para los tramos de la noche son planos.

- Para los tramos de las 03h y 18h solo hay irradiación (ligera irradiación) desde mayo hasta agosto/septiembre.
- El resto de los tramos horarios registran más volatilidad durante el año, alcanzando los valores máximos en agosto (y en verano en general), mientras que los valores mínimos son en diciembre, justamente cuando los días son más cortos.

Para terminar el análisis de las irradiaciones, se procede a extraer en esta ocasión la tendencia / Nivel de la serie, lo cual va a permitir extraer 2 conclusiones idénticas a las extraídas con el dataset de irradiaciones: 1) en torno a qué valores se mueve cada tramo y 2) si ha habido una tendencia clara en algún tramo durante estos años.

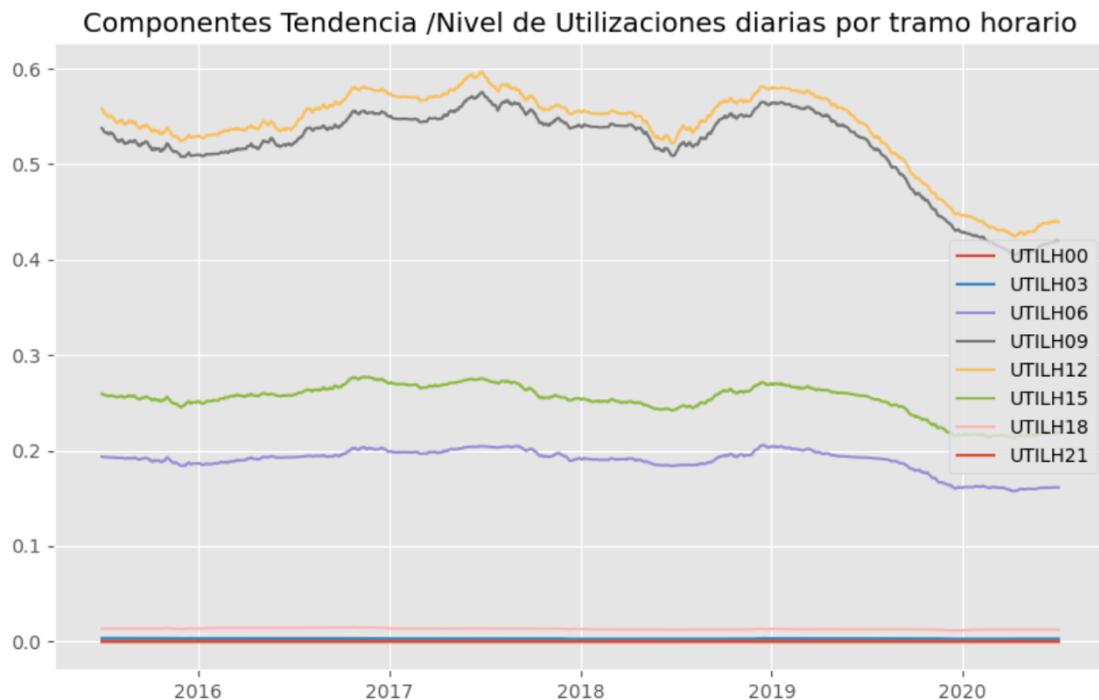


Ilustración 9. Descomposición de tendencia de utilizaciones diarias por tramo horario.

- Tal y como anticipaba antes, la Ilustración 9 muestra como los valores máximos se alcanzan a las 12h, muy seguido de las 9h.
- A dichos valores horarios les sigue el tramo de las 15h y el de las 06h.
- Después se observa cierto valor de utilización a las 18h y algo, casi nada a las 03. Mientras que los valores a las 21 y a las 00h es siempre igual a 0.

Por último, se representan las distribuciones de cada tramo horario, para identificar valores atípicos por tramo horario.

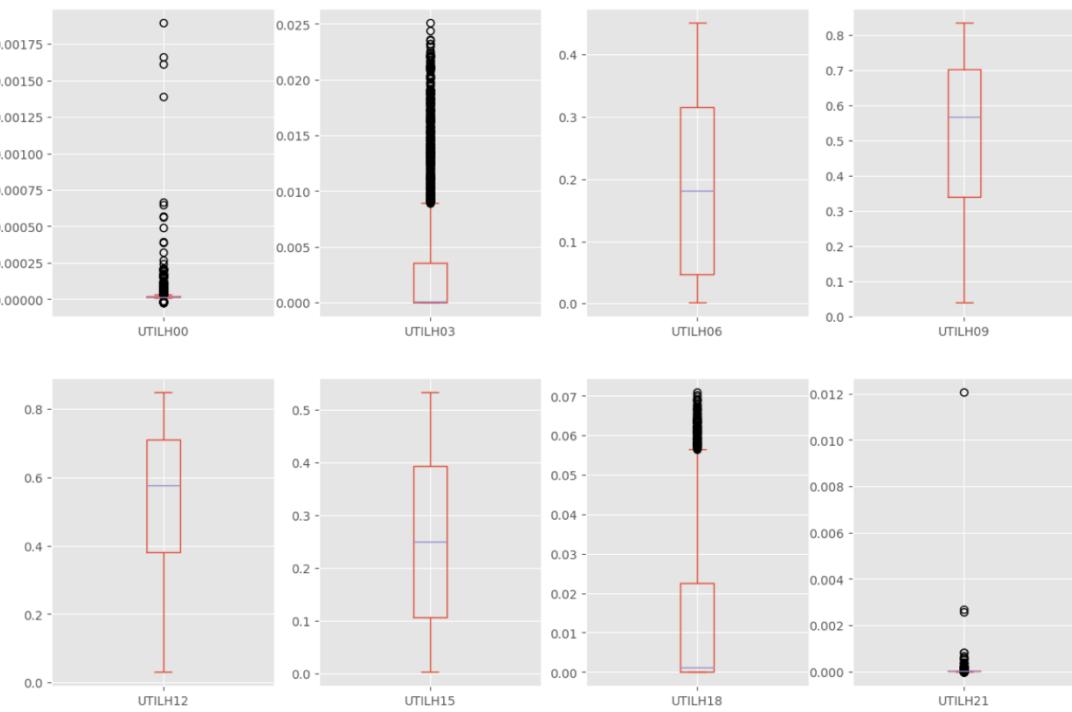


Ilustración 10. Distribuciones representadas mediante boxplots de los tramos horarios de las utilizaciones.

A diferencia de lo que observábamos en la Ilustración 6 del dataset de Irradiaciones, la Ilustración 10 superior muestra como sí existen más valores atípicos en el dataset de utilizaciones.

Por un lado, se observan valores atípicos a las 21h y 00h, esto se debe a que, pese a que la serie es igual a 0 en la mayoría de los casos, también hay ciertas fechas con valores distintos a 0, lo que hace que se conviertan en outliers.

Por otra parte, en las utilizaciones de las 03h y las 18h vuelve a ocurrir como pasaba en el caso de las irradiaciones. Hay bastantes outliers debido a poseer valores elevados de utilización. Estos serán justamente los valores registrados en junio-julio que es cuando esta serie registra anualmente sus valores máximos.

A continuación, ponemos el foco en el caso de las 00 y 21, con el objetivo de identificar los outliers:

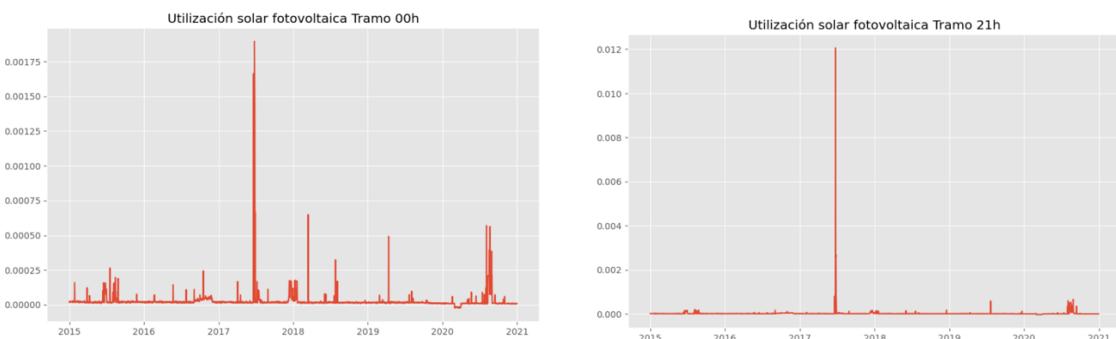


Ilustración 11. Utilización solar fotovoltaica en los tramos de las 00 hrs y 21 hrs.

Tal y como se observa en la figura superior (Ilustración 11) ambas series son en casi todos los casos igual a 0, salvo en el caso de mediados de verano de 2017 en ambos tramos horarios.

2.3. Relación entre las variables

Para finalizar con el análisis exploratorio, se procede a representar la relación entre variables mediante una matriz de correlaciones para cada uno de los dataset (irradiaciones y utilizaciones).

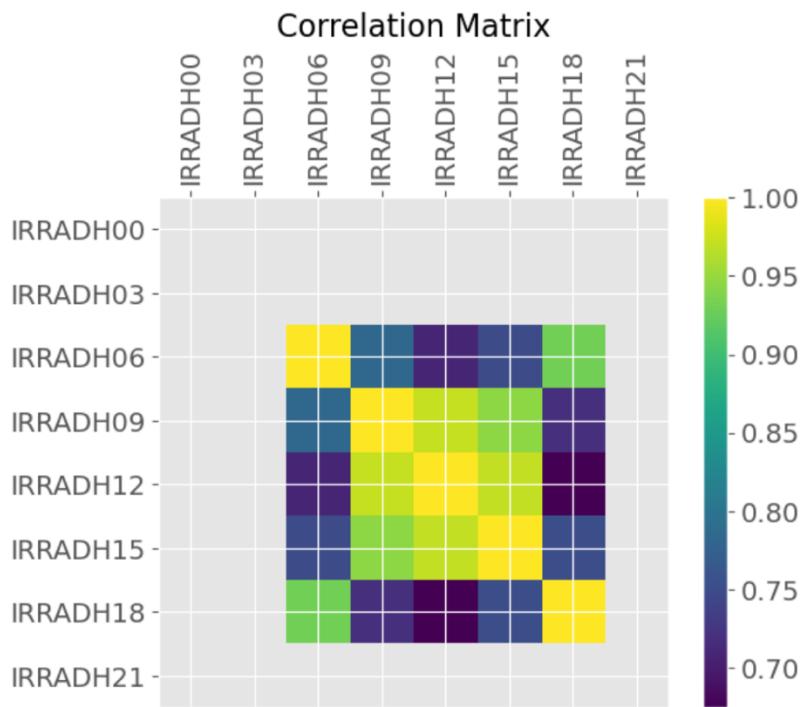


Ilustración 12. Matriz de correlaciones lineales de todas las variables del dataset de irradiaciones solares.

En este primer gráfico de correlaciones (Ilustración 12), se observan valores constantes iguales a 0 en las columnas 'IRRADH00', 'IRRADH03' e 'IRRADH21', por lo que, en consecuencia, sus columnas y filas aparecen “blancas” implicando una correlación nula (valores NaN originados por la no variabilidad de las variables).

En cuanto al resto, se aprecia una correlación muy alta entre las irradiaciones de las 12h y las 09h, siendo una correlación cercana a 1. A esta, le sigue la correlación entre las irradiaciones de las 15h y las 09h, así como la de las 18h y las 06h, ambas correlaciones tienen valores superiores a 0.9.

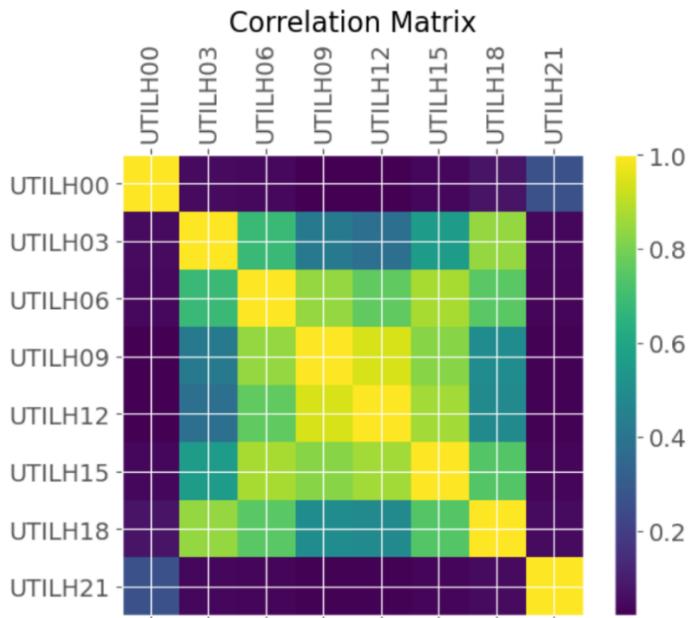


Ilustración 13. Matriz de correlaciones lineales de todas las variables del dataset de utilización fotovoltaica.

En este segundo gráfico de correlaciones (Ilustración 12), de nuevo la correlación más elevada es aquella que se da entre las utilizaciones de las 09h y las 12h, seguida muy de cerca de la correlación entre las 06h y 15h, junto con la de las 03h y 18h y la de las 12h y 15h.

Todas ellas son superiores a 0.85, y se relacionan bastante con lo mencionado anteriormente al estudiar los patrones estacionales y de tendencia de cada uno de los tramos horarios.

3. PREPARACIÓN DE LOS DATOS Y FEATURE ENGINEERING

Previamente a llevar a cabo cualquier tipo de modelo, se necesita definir cuál va a ser la columna que se pretende predecir (tarea de regresión), así como identificar qué variables pueden ser útiles para esta tarea, bien sean columnas ya incluidas en alguno de los datasets, o columnas a las que les hagamos los ajustes necesarios.

3.1. Target de nuestro problema:

Nuestro objetivo es predecir cuál va a ser la utilización solar fotovoltaica para un tramo horario determinado, un día, mes y año concreto. No obstante, en nuestro caso ahora mismo se tiene una columna para tramos horarios de utilización.

Si dejáramos así el dataset, necesitaríamos entrenar los modelos 8 veces distintas, asumiendo cada vez que el target es un tramo horario de utilización concreto (iríamos cambiando la columna considerada target del problema).

Para no tener que hacer esto, y poder directamente entrenar el modelo de manera que pueda predecir la utilización de cualquier tramo horario, vamos a recoger en una columna UTILHtodas las utilizaciones (en lugar de tener 1 registro por día, ahora tendremos 8, uno para cada tramo horario).

3.2. Variables explicativas:

Ahora bien, para no perder la capacidad explicativa que tienen los valores de los otros tramos horarios sobre nuestro target (como se demostraba en la matriz de correlación, la utilización de las 9h, por ejemplo, es casi la utilización de las 12h), añadimos las siguientes columnas como variables explicativas:

- 'UTILH_TramoAnterior', será la utilización registrada ese mismo día 3 horas antes
 - 'UTILH_TramoAnterior_x2', será la utilización registrada ese mismo día 6 horas antes (2 tramos horarios antes)
 - 'UTILH_TramoAnterior_x3', será la utilización registrada ese mismo día 9 horas antes (3 tramos horarios antes)
-
- 'UTILH_TramoPosterior', será la utilización registrada el día anterior, 3 horas después
 - 'UTILH_TramoPosterior_x2', será la utilización registrada el día anterior, 6 horas después (2 tramos horarios después)
 - 'UTILH_TramoPosterior_x3', será la utilización registrada el día anterior, 9 horas después (3 tramos horarios después)
-
- 'UTILH_DiaAnterior', será la utilización registrada a esa misma hora el día anterior
 - 'UTILH_DiaAnterior_x2', será la utilización registrada a esa misma hora 2 días anteriores
 - 'UTILH_AñoAnterior', será la utilización registrada a esa misma hora el mismo día un año antes
-
- 'IRRAD_misma_hora', será la irradiación registrada ese mismo día a esa misma hora
-
- 'IRRADH_TramoAnterior', será la irradiación registrada ese mismo día 3 horas antes
 - 'IRRADH_TramoAnterior_x2', será la irradiación registrada ese mismo día 6 horas antes
 - 'IRRADH_TramoAnterior_x3', será la irradiación registrada ese mismo día 9 horas antes
-
- 'IRRADH_TramoPosterior', será la irradiación registrada el día anterior, 3 horas después
 - 'IRRADH_TramoPosterior_x2', será la irradiación registrada el día anterior, 6 horas después
 - 'IRRADH_TramoPosterior_x3', será la irradiación registrada el día anterior, 9 horas después

-
- 'DIA', 'MES', 'DIAMES' y 'TramoHorario' para complementar en lo relativo a la fecha-hora del registro

Cabe destacar que todos los datos se cogen sobre el mismo día o sobre días pasados, pues en la práctica no tendremos los datos de mañana o de horas más tarde, es decir, datos futuros.

3.3. Ajustes finales:

En primer lugar, lo que hacemos es unir ambos datasets con el objetivo de tenerlos ya todos unidos por fecha: para un día concreto, tendremos todas las irradiaciones y utilizaciones.

Una vez ya tenemos un único dataset, realizamos el cambio de target y la adición de las variables explicativas que hemos mencionado anteriormente en los apartados 3.1y 3.2.

Ya con las nuevas columnas generadas, dividimos entre conjunto de entrenamiento y conjunto de prueba. Como no vamos a actuar como un modelo de forecasting, esta división será de carácter aleatoria (no mantenemos estructura temporal haciendo una división secuencial).

Importante: Vamos a realizar una primera división 90-10, donde reservamos el 10% de los datos para su uso exclusivo al final del proceso, con el fin de verificar al 100% la capacidad generativa del modelo final seleccionado. Una vez completada esta separación inicial, procedemos a dividir el 90% restante en conjuntos de entrenamiento y prueba, asignando el 80% de los datos para el entrenamiento del modelo y reservando el 20% restante para evaluar su desempeño.

Esta estrategia nos permite entrenar el modelo con una cantidad significativa de datos y luego validar su capacidad predictiva de manera rigurosa, garantizando así su eficacia en aplicaciones futuras.

4. MODELOS DE REGRESIÓN

Para realizar el problema de regresión que se nos ha planteado, usaremos tanto modelos de ensamblado, como modelos más clásicos.

En concreto, estos serán los modelos que van a ser entrenados (todos ellos tuneando sus hiperparámetros y obteniendo métricas de error para compararlos posteriormente)

- Regresión lineal
- Árbol de regresión simple
- Bagged Tree
- Random Forest
- Gradient Boosting

4.1. Regresión Lineal:

Antes de entrar en los modelos de ensamblado, vamos a entrenar un modelo simple e interpretable como es una regresión lineal. Este modelo nos servirá como baseline, y también

nos aportará información sobre la importancia de las variables gracias a los coeficientes estimados.

Así pues, comenzamos entrenando el modelo **con todas las variables explicativas que tenemos**.

En la Ilustración 14 se puede observar cómo pese a que la gran mayoría de variables sí son significativas, algunas de ellas no:

var	coef	std err	t	P> t
num_ANNO	0.187	0.000	401.896	0.000
num_MES	-0.002	0.001	-3.934	0.000
num_DIA	-0.001	0.000	-1.149	0.251
num_DIASEM	-0.000	0.000	-0.274	0.784
num_Tramo_Horario	0.000	0.000	0.901	0.367
num_UTILH_DiaAnterior	0.005	0.001	4.247	0.000
num_UTILH_DiaAnterior_x2	0.018	0.002	8.877	0.000
num_UTILH_AñoAnterior	0.018	0.001	12.318	0.000
num_UTILH_TramoAnterior	0.056	0.002	33.771	0.000
num_UTILH_TramoPosterior	0.114	0.002	50.225	0.000
num_UTILH_TramoAnterior_x2	0.081	0.002	34.592	0.000
num_UTILH_TramoPosterior_x2	-0.051	0.003	-19.464	0.000
num_UTILH_TramoAnterior_x3	-0.038	0.002	-16.257	0.000
num_UTILH_TramoPosterior_x3	0.013	0.002	5.403	0.000
num_IRRADH_misma_hora	0.014	0.002	7.091	0.000
num_IRRADLH_AñoAnterior	0.169	0.002	81.163	0.000
num_IRRADDH_TramoAnterior	-0.064	0.002	-35.691	0.000
num_IRRADDH_TramoPosterior	-0.115	0.002	-47.673	0.000
num_IRRADH_TramoAnterior_x2	-0.018	0.002	-8.019	0.000
num_IRRADH_TramoPosterior_x2	0.079	0.003	30.033	0.000
num_IRRADH_TramoAnterior_x3	0.002	0.002	0.699	0.485
num_IRRADH_TramoPosterior_x3	-0.040	0.002	-19.816	0.000

Ilustración 14. Coeficientes estimados y p-valores de un modelo de regresión lineal simple entrenado con todas las variables explicativas.

Fijándonos en el p-valor de cada una, se ve que las columnas 'DIA', 'DIASEM', 'TramoHorario' e 'IRRADH_TramoPosterior_x3' presentan p-valores muy elevados (superiores a 0.05) y son, por tanto, no muy significativas para el modelo.

El hecho de que las variables relacionadas con la fecha, sobre todo 'DIAMES' y 'TramoHorario' no resulten significativas, implica que probablemente las regresiones se están obteniendo a partir de las variables de utilizaciones e irradiaciones.

A continuación, se representan gráficamente en la Ilustración 15 las métricas de error MSE y MAE para el modelo de regresión lineal simple en el conjunto de entrenamiento.

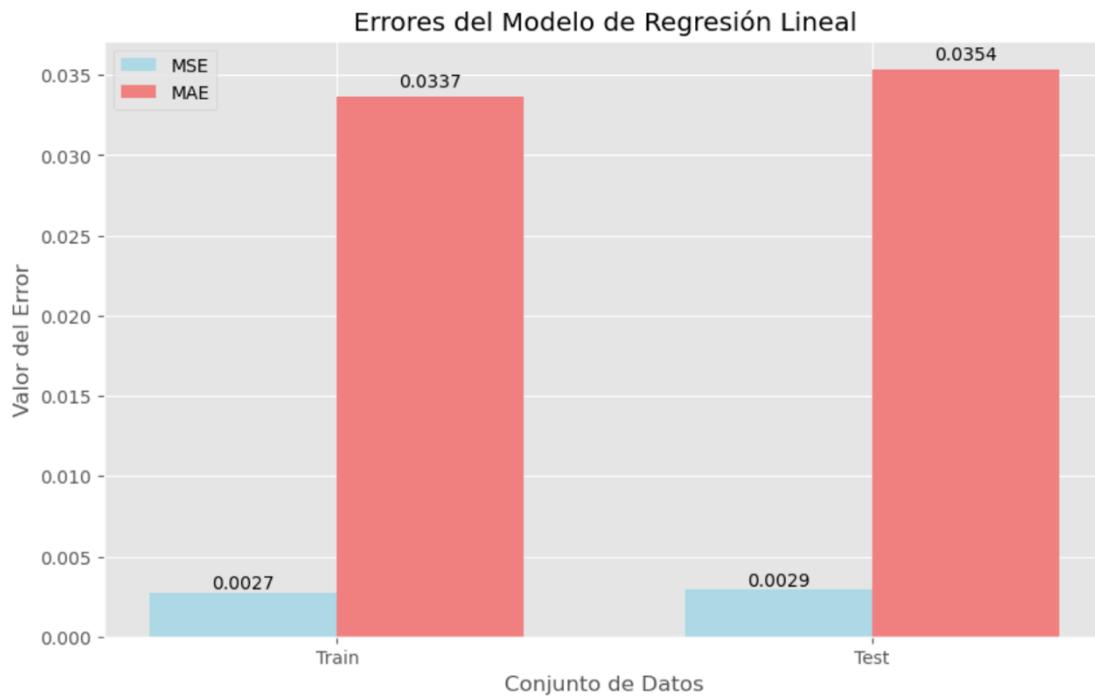


Ilustración 15. Métricas de error para el modelo de regresión lineal simple entrenado con todas las variables explicativas.

Observando la figura superior, se obtiene un MSE igual a 0.0027 y un MAE igual a 0.034 en el conjunto de entrenamiento. En cuanto a los errores en el test, no hay prácticamente diferencia, por lo que no hemos caído en sobreentrenamiento.

A continuación, se prueba ahora sin las variables que han resultado no significativas:

var	coef	std err	t	P> t
num_ANNO	0.188	0.000	415.204	0.000
num_MES	-0.003	0.000	-5.738	0.000
num_UTILH_DiaAnterior	-0.000	0.000	-0.503	0.615
num_UTILH_DiaAnterior_x2	0.028	0.002	15.147	0.000
num_UTILH_AñoAnterior	0.018	0.001	12.601	0.000
num_UTILH_TramoAnterior	0.057	0.002	35.687	0.000
num_UTILH_TramoPosterior	0.100	0.002	47.120	0.000
num_UTILH_TramoAnterior_x2	0.067	0.002	30.759	0.000
num_UTILH_TramoPosterior_x2	-0.024	0.002	-11.095	0.000
num_UTILH_TramoAnterior_x3	-0.023	0.002	-13.242	0.000
num_UTILH_TramoPosterior_x3	-0.018	0.002	-11.211	0.000
num_IRRAD_misma_hora	-0.001	0.001	-0.767	0.443
num_IRRADLH_AñoAnterior	0.167	0.002	82.234	0.000
num_IRRADLH_TramoAnterior	-0.066	0.002	-38.249	0.000
num_IRRADLH_TramoPosterior	-0.103	0.002	-44.965	0.000
num_IRRADLH_TramoAnterior_x2	-0.016	0.002	-8.704	0.000
num_IRRADLH_TramoPosterior_x3	0.052	0.002	24.379	0.000

Ilustración 16. Coeficientes estimados y p-valores de un modelo de regresión lineal simple entrenado excluyendo las variables que han resultado no significativas.

Ahora aparecen 2 nuevas variables no significativas. Igualmente, a la hora de obtener de nuevo el error en las predicciones del conjunto de entrenamiento y de prueba, este ha permanecido igual (Ilustración 17).

Dado que solo se trata de una simple regresión lineal, y que no hemos conseguido mejoras significativas, de primeras vamos a dejar todos los inputs en los siguientes modelos.

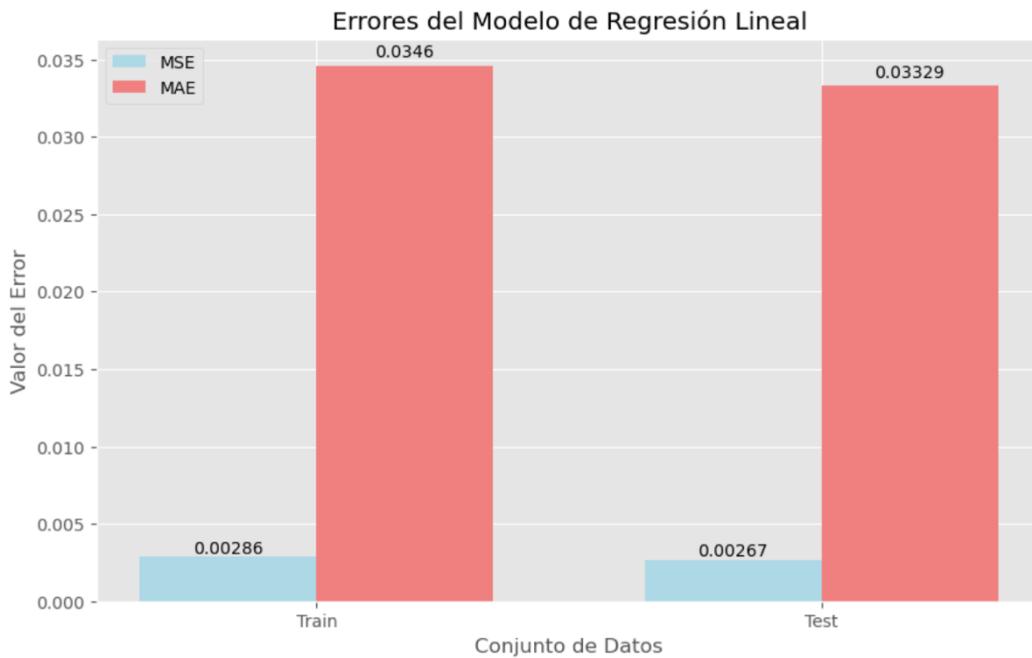


Ilustración 17. Métricas de error para el modelo de regresión lineal simple entrenado excluyendo las variables que han resultado no significativas.

4.2. Árbol de Regresión sencillo

Para continuar, probamos con otro modelo previo a las técnicas de ensamblado, como puede ser un árbol de regresión sencillo, cuyos hiperparámetros tuneamos empleando un *GridSearch* (*min_impurity_decrease*, *min_samples_leaf* y *min_samples_split*).

El árbol de decisión resultante es tan profundo y complejo que no se puede apreciar. Es decir, tiene una enorme cantidad de cortes, visibles en la Ilustración 18.

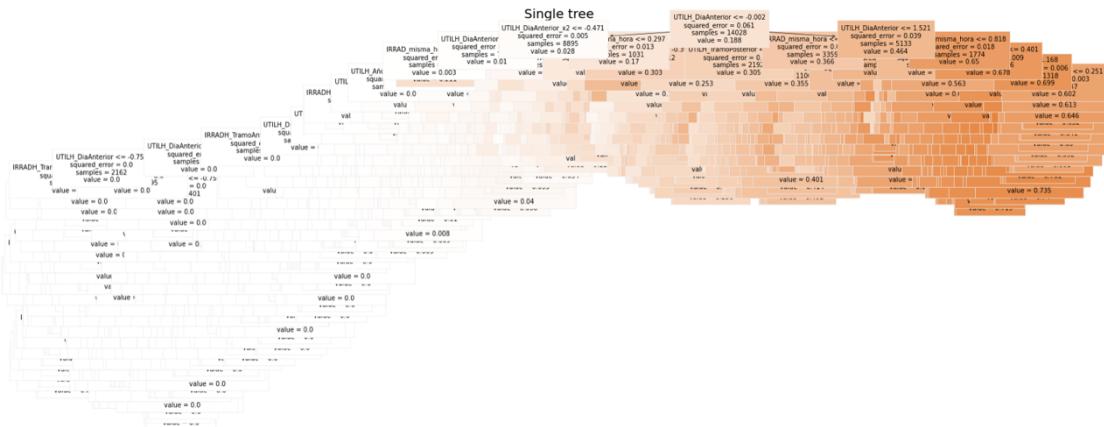


Ilustración 18. Representación del árbol de regresión sencillo entrenado.

No obstante, se procede a extraer la importancia de las variables explicativas para este modelo (*feature importance*) para saber qué variables son más relevantes a la hora de obtener las particiones. Dichas importancias, quedan representadas en la siguiente Ilustración 19.

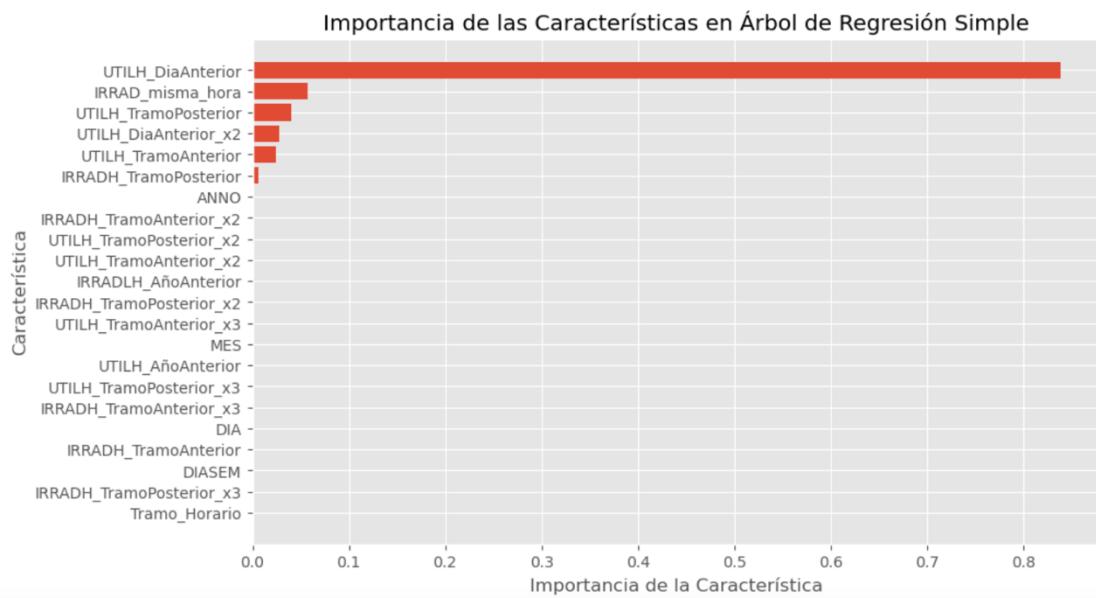


Ilustración 19. Representación de la importancia de las características en el árbol de regresión simple.

De las 21 variables explicativas originales, solo 6 aparecen como significativas en el modelo tal y como queda representado en Ilustración 19. Estas son: la utilización a la misma hora el día anterior como primera variable explicativa, con mucha diferencia. Y, a continuación, se encuentran la irradiación registrada ese mismo día a la misma hora, las utilizaciones registradas en el tramo anterior o el posterior, junto con la irradiación del tramo posterior.

Continuamos el análisis comparando los valores predichos con los reales. Esto se representa a continuación en Ilustración 20.

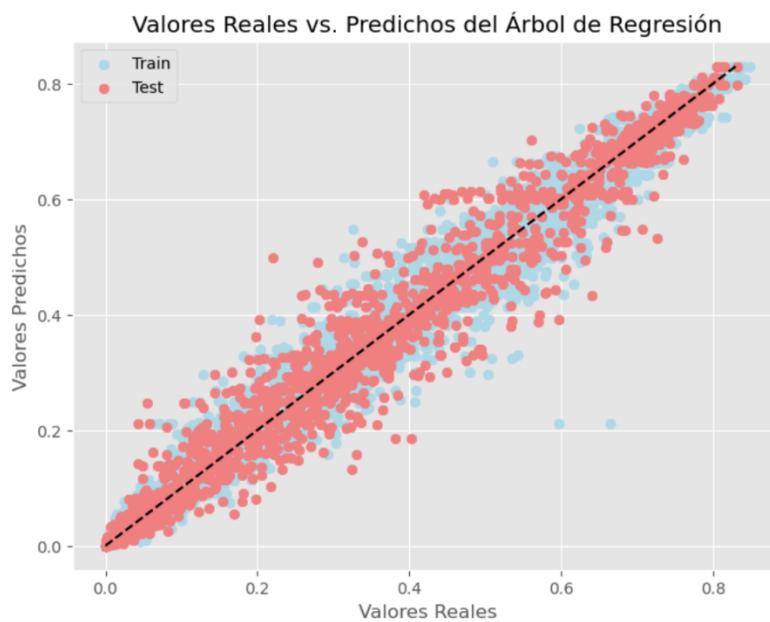


Ilustración 20. Representación de los valores reales vs. los valores predichos del árbol de regresión simple.

Observando el gráfico superior (Ilustración 20) parece que estamos cometiendo errores muy similares en el conjunto de entrenamiento y de prueba (puntos azules y rojos, respectivamente). Todos ellos están aproximadamente sobre la línea diagonal discontinua, aunque se observa

cierto aumento en la desviación de las predicciones, y mayor variabilidad en utilizaciones intermedias.

Esto podemos complementarlo con las métricas de error propias de una regresión. Estas son de nuevo el MSE y MAE, obtenidos anteriormente para la regresión lineal simple. De igual modo que para el modelo anterior, la representa dichas métricas de error para el modelo que nos atañe actualmente, es decir, el árbol de regresión sencillo.

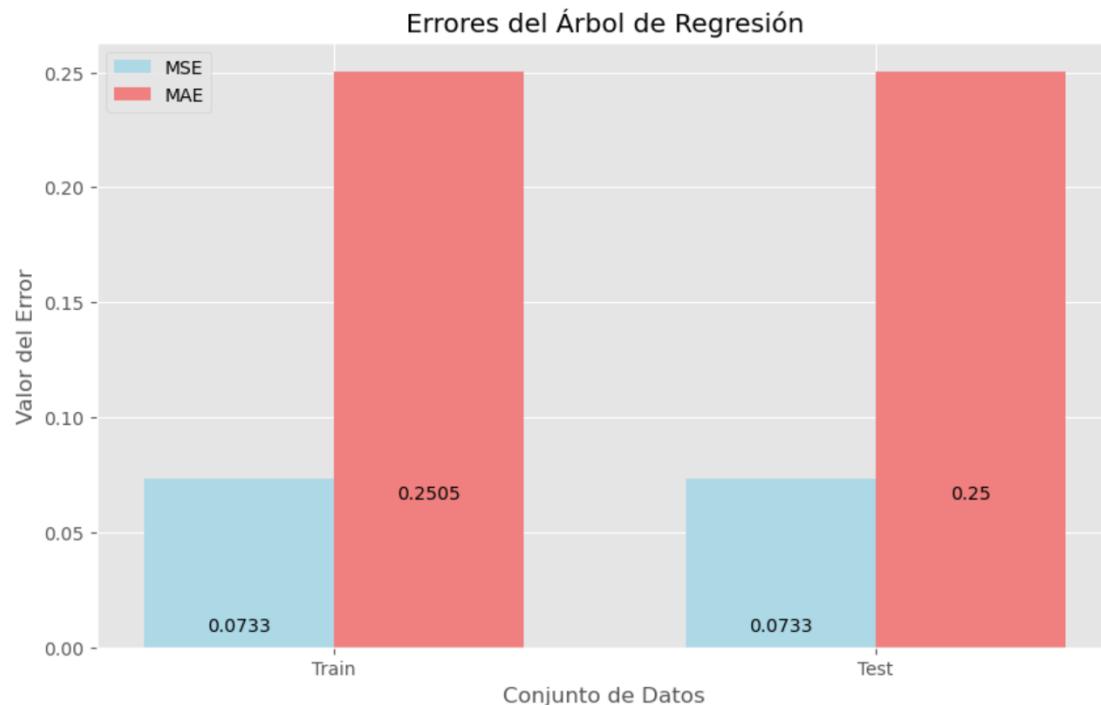


Ilustración 21. Métricas de error para el modelo de árbol de regresión sencillo.

Pese a lo que cabría esperar, al comparar los errores de este árbol de regresión con los errores obtenidos en la regresión lineal simple, se observa que el error con este modelo es muy superior.

Como el objetivo de estos modelos solamente es que sirvieran de baseline, decidimos no continuar el análisis con mayor profundidad, pues además hemos obtenido resultados mejorables.

Sin embargo, ya sabemos que los siguientes modelos, dado que ya son técnicas de ensamble, como máximo deben tener un error como el cometido con este árbol sencillo o con la regresión lineal.

4.3. Bagged Tree

El método Bagged Trees es un enfoque de ensamble que combina múltiples árboles de regresión entrenados en diferentes subconjuntos de datos de entrenamiento, con el objetivo de mejorar la precisión predictiva y reducir el sobreajuste.

Como el árbol con Bagging ya es un método de ensamblado, usamos varios árboles para obtener los resultados. En concreto, esta cantidad óptima de B árboles resulta en 25 árboles, tal y como muestra la Ilustración 22.

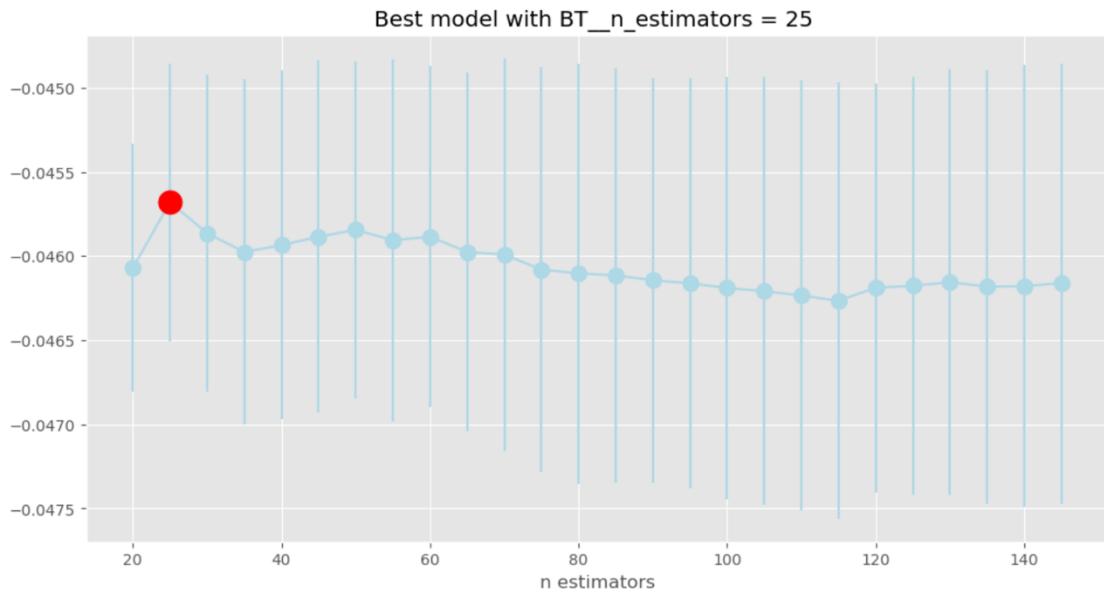


Ilustración 22. Gráfico del hiperparámetro sobre número de árboles ensamblados óptimo en el modelo Bagged Tree.

Para observar qué tal se comporta este modelo de Bagging, podemos representar de igual manera que en el apartado anterior, una comparación entre los valores predichos y los valores reales.

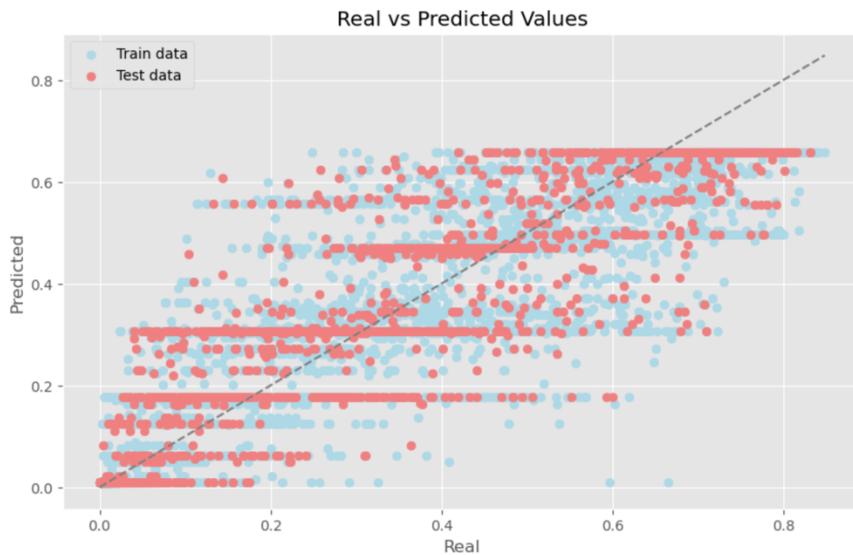


Ilustración 23. Representación de los valores reales vs. los valores predichos del modelo de ensamblado Bagged Tree.

La Ilustración 23 muestra como las predicciones obtenidas con este modelo de Bagging son muy erróneas. Ni con el conjunto de entrenamiento ni con el de prueba somos capaces de obtener predicciones centradas en la diagonal discontinua, donde estarían si las predicciones fueran correctas al representar dicha diagonal una relación perfecta entre los valores reales y los predichos.

Siguiendo la misma metodología que con los modelos anteriores, se procede a representar con gráficos de barras las métricas de error para este primer modelo de ensamblado entrenado. Aunque, a priori, podemos esperar que este modelo no sea el que escojamos en un futuro por nuestro conocimiento sobre otros modelos de ensamblado que surgen como mejora a este.

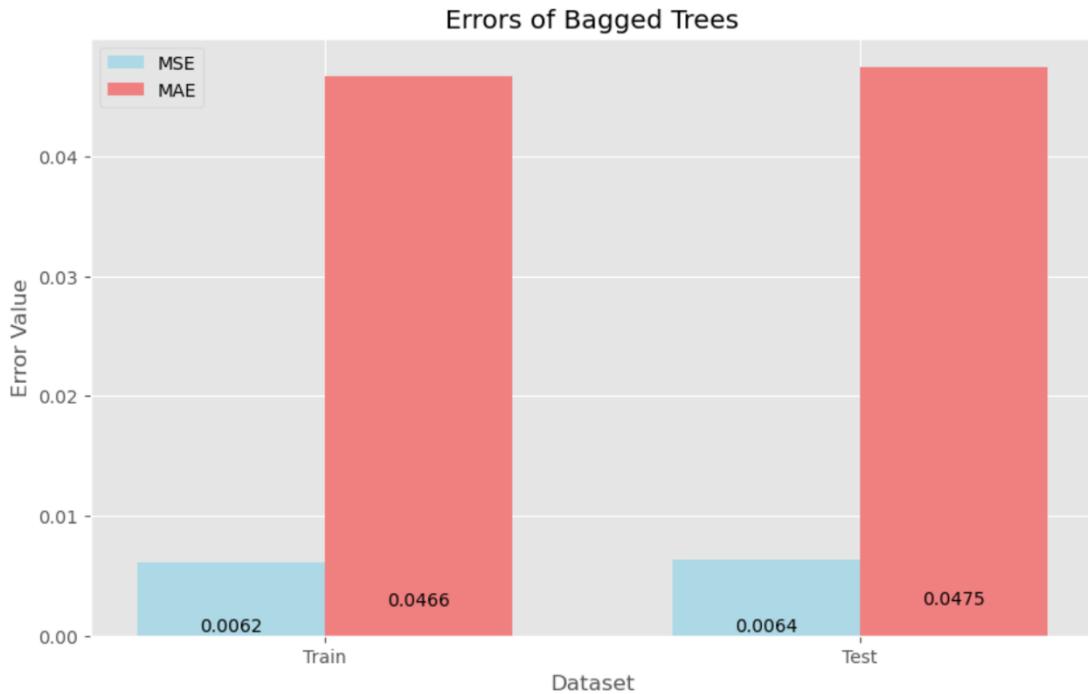


Ilustración 24. Métricas de error para el modelo de ensamblado Bagged Tree.

Si bien es cierto que los errores en este caso son elevados (Ilustración 24) e incluso más elevados que con el árbol sencillo (Ilustración 21), no estamos cometiendo sobreaprendizaje, dado que los errores del conjunto de prueba son prácticamente iguales a los errores en el de entrenamiento.

4.4. Random Forest

A continuación, continuamos probando en nuestro análisis con un modelo de Random Forest.

Random Forest es preferible a un Bagged Tree porque además de entrenar múltiples árboles en subconjuntos aleatorios de datos, también utiliza un subconjunto aleatorio de características en cada árbol, lo que aumenta la diversidad y la robustez del modelo, reduciendo aún más el riesgo de sobreajuste y mejorando la precisión predictiva.

Para entrenar este modelo de ensamblado, se ha hecho un tuneado de hiperparámetros con *GridSearch*.

Veamos los 3 primeros árboles que se obtienen en la siguiente Ilustración 25.

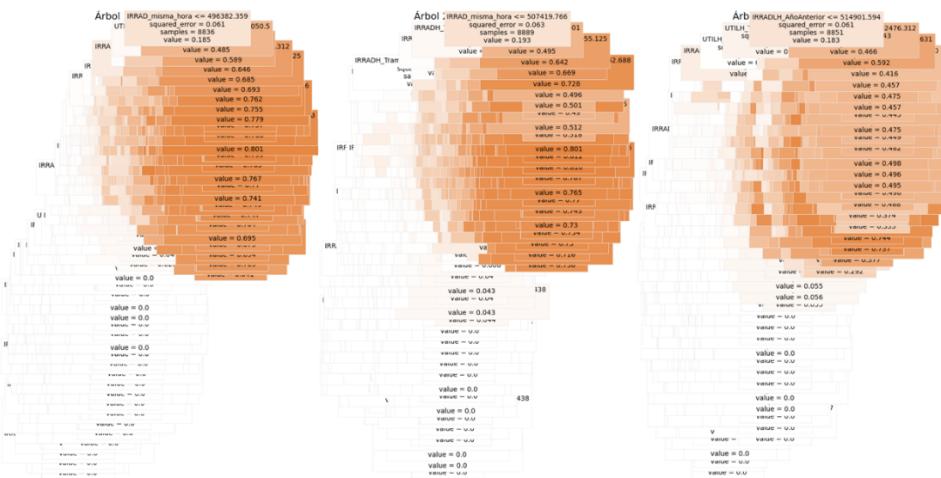


Ilustración 25. Representación de los tres primeros árboles que forman el modelo entrenado de ensamblado tipo Random Forest.

Aunque no se aprecia prácticamente nada, sí que se ve que los primeros cortes están relacionados con la Irradiación registrada a esa misma hora.

Siguiendo con este análisis, podemos obtener de nuevo las importancias de cada una de las variables para el caso de este modelo. Esto queda reflejado en la Ilustración 26.

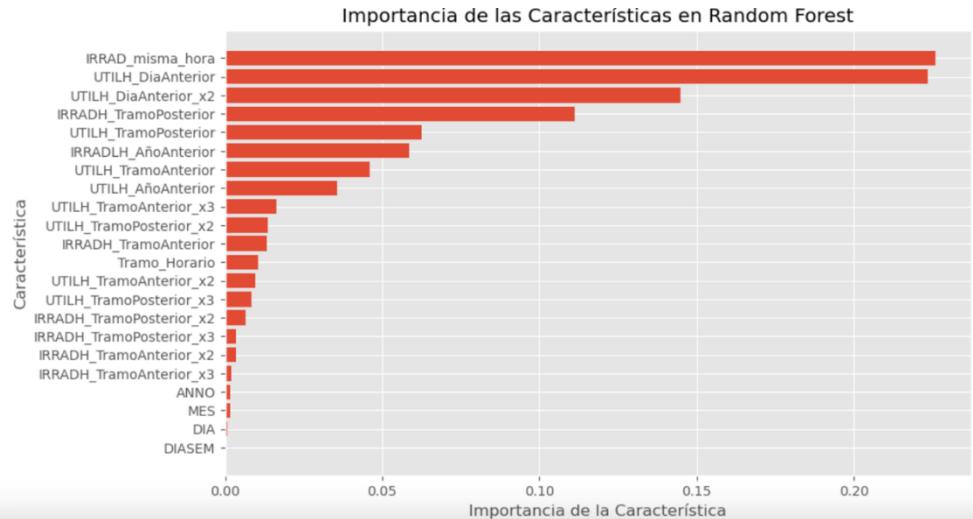


Ilustración 26. Representación de la importancia de las características en el modelo ensamblado Random Forest.

Aquellas variables con mayor relevancia en este modelo son la Irradiación registrada esa misma hora, muy seguida de la utilización del mismo tramo horario el día anterior.

En otro nivel se encuentra ya la utilización de 2 tramos horarios más tarde el día anterior o la irradiación del tramo horario siguiente.

No obstante, todas las variables, exceptuando el Día de la Semana, tienen cierta relevancia en el modelo.

Veamos ahora cómo nos estamos equivocando en la siguiente

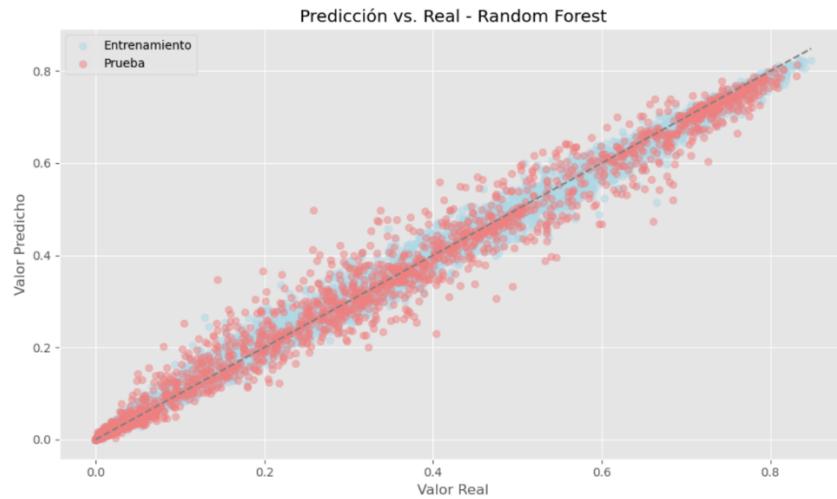


Ilustración 27. Representación de los valores reales vs. los valores predichos del modelo de ensamblado Random Forest.

En relación a la Ilustración 27. Representación de los valores reales vs. los valores predichos del modelo de ensamblado Random Forest., para tener predicciones perfectas, deberíamos tener todos los puntos (azules y rojos) sobre la línea discontinua diagonal. Pese a estar cometiendo errores, no son errores de demasiada magnitud; casi todos los puntos están sobre la diagonal.

Ahora bien, sí parece que hay cierto sobreentrenamiento, pues los errores son superiores en el conjunto de prueba (puntos rojos), principalmente cuando las utilidades tienen valores intermedios. Para obtener una mejor idea sobre dicho sobreentrenamiento, se procede a obtener las métricas de error para este modelo de Random Forest.

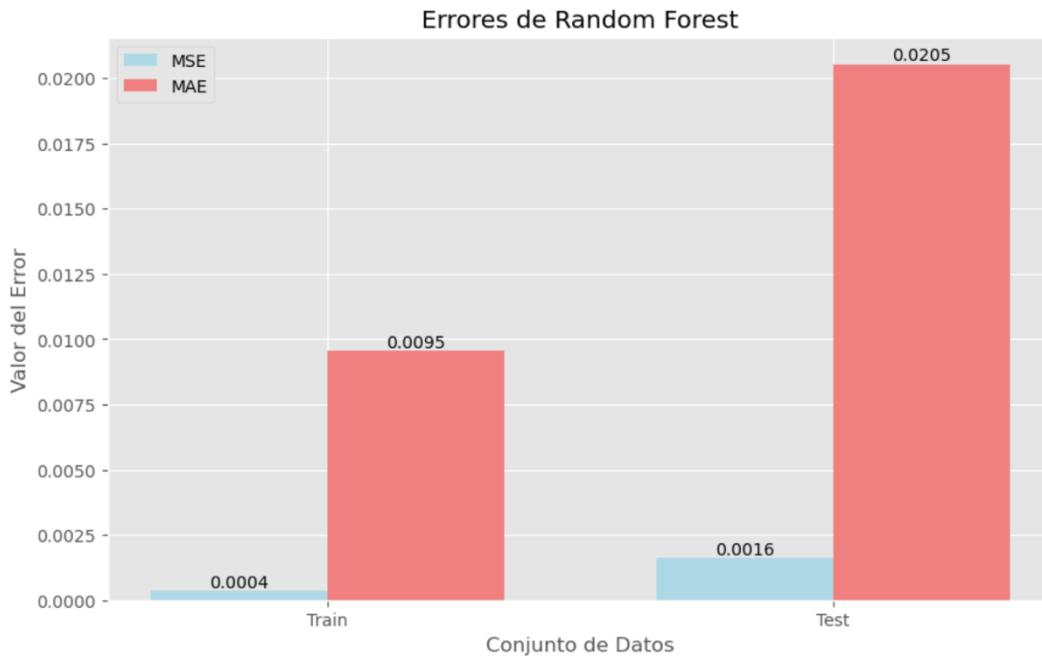


Ilustración 28. Métricas de error para el modelo de ensamblado Random Forest.

Efectivamente, la Ilustración 28 muestra que estábamos en lo cierto. Pese a haber realizado un tuneado de los hiperparámetros estamos cayendo en sobreentrenamiento (pues el error en el conjunto de prueba es bastante superior al error en el conjunto de entrenamiento).

4.5. GradientBoosting

Para poner a prueba modelos de Boosting también en este análisis, nos decantamos por Gradient Boosting.

En un inicio, también nos planteamos la posibilidad de usar AdaBoost. Sin embargo, optamos por Gradient Boosting debido a su capacidad para optimizar funciones de pérdida arbitrarias y su mejor capacidad para manejar datos de alta dimensionalidad y ruido. Además, Gradient Boosting tiende a ofrecer mejores resultados en términos de precisión predictiva en comparación con AdaBoost en una variedad de conjuntos de datos.

4.5.1. Prueba 1: con todas las variables explicativas

Como se acaba de explicar, Gradient Boosting es preferido sobre AdaBoost debido a su capacidad para optimizar directamente cualquier función de pérdida diferenciable, lo que permite una mayor flexibilidad en la optimización del modelo. Además, Gradient Boosting puede manejar eficazmente datos heterogéneos y ruidosos, y tiende a ser más robusto frente al sobreajuste.

Dado que en las técnicas de ensamblado basadas en boosting se emplean árboles muy simples, la Ilustración 29 muestra como en este caso sí podemos llegar a visualizar algunos de los árboles que componen nuestro modelo.

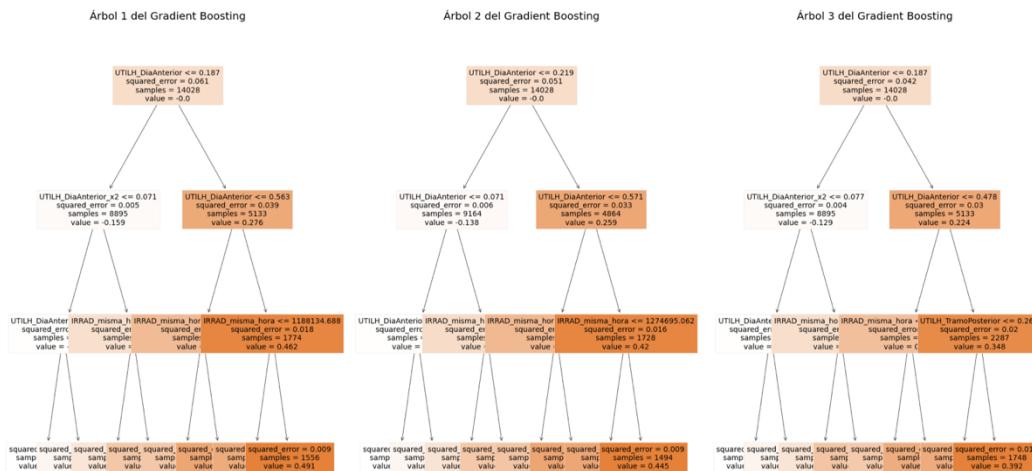


Ilustración 29. Representación de los tres primeros árboles que forman el modelo entrenado de Gradient Boosting.

Veamos a continuación cuáles han sido las importancias de cada una de las variables en este modelo, representando esto en la siguiente Ilustración 30.

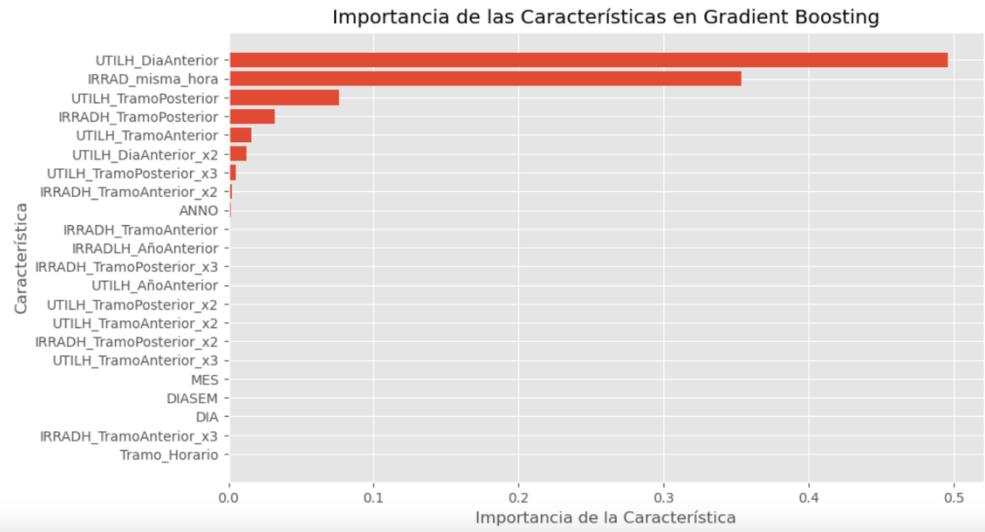


Ilustración 30. Representación de la importancia de las características en el modelo Gradient Boosting.

A diferencia de en el caso de Random Forest (Ilustración 26), en este caso (Ilustración 30), no todas las variables aparecen como relevantes. De hecho, solo las 9 primeras tienen algo de importancia, aunque principalmente destacan 2: la utilización registrada a la misma hora el día anterior y la irradiación registrada a esa misma hora ese mismo día.

A estas 2 le siguen la radiación y la utilización registradas en el tramo horario posterior.

Seguimos el análisis estudiando, como en los casos anteriores, los errores cometidos y la diferencia entre los valores reales y predichos por el modelo.

En cuanto a la diferencia entre los errores de predicción del conjunto de entrenamiento y de prueba, en este caso, incluso parecen superiores en el entrenamiento que en el conjunto de prueba según lo observable en la Ilustración 31. Es decir, parece que Gradient Boosting sí que es un modelo robusto que sabe generalizar.

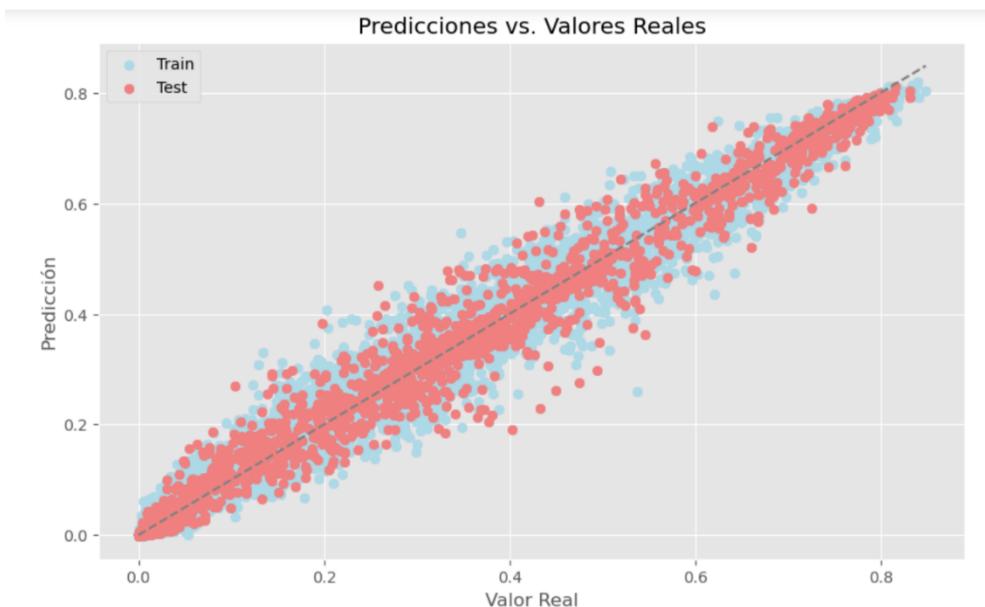


Ilustración 31. Representación de los valores reales vs. los valores predichos del modelo de Gradient Boosting.

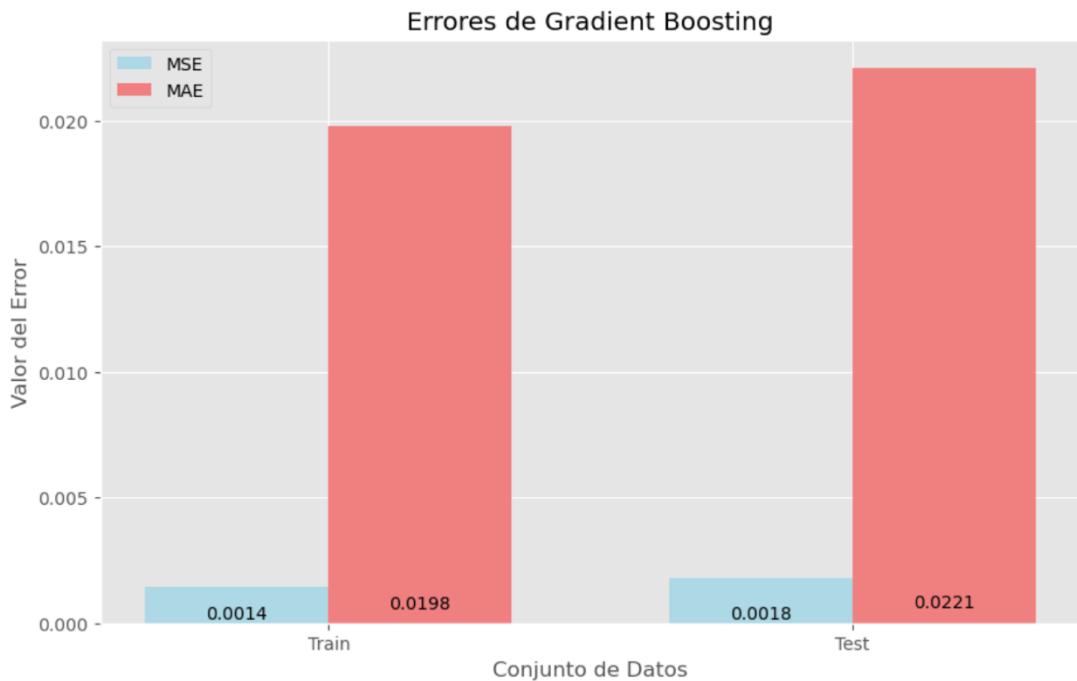


Ilustración 32. Métricas de error para el modelo de Gradient Boosting.

Efectivamente, se observa en la Ilustración 32 como obtenemos métricas de error superiores a las cometidas anteriormente con el modelo Random Forest. Obtenemos un MAPE de hasta dos veces el del anterior caso.

4.5.2. Prueba 2: Solo variables resultantes como significativas del modelo anterior
Hacemos esta última prueba con la intención de extraer del modelo aquellas variables que no son de importancia, para ver si sin ellas, logramos mejorar las predicciones debido a no meter 'ruido' en el modelo.

En primer lugar, en cuanto a la importancia de las variables explicativas, seguimos manteniendo el mismo orden y grado de relevancia que en el caso anterior (Ilustración 30 de todas las variables vs. Ilustración 33 excluyendo las variables no significativas).

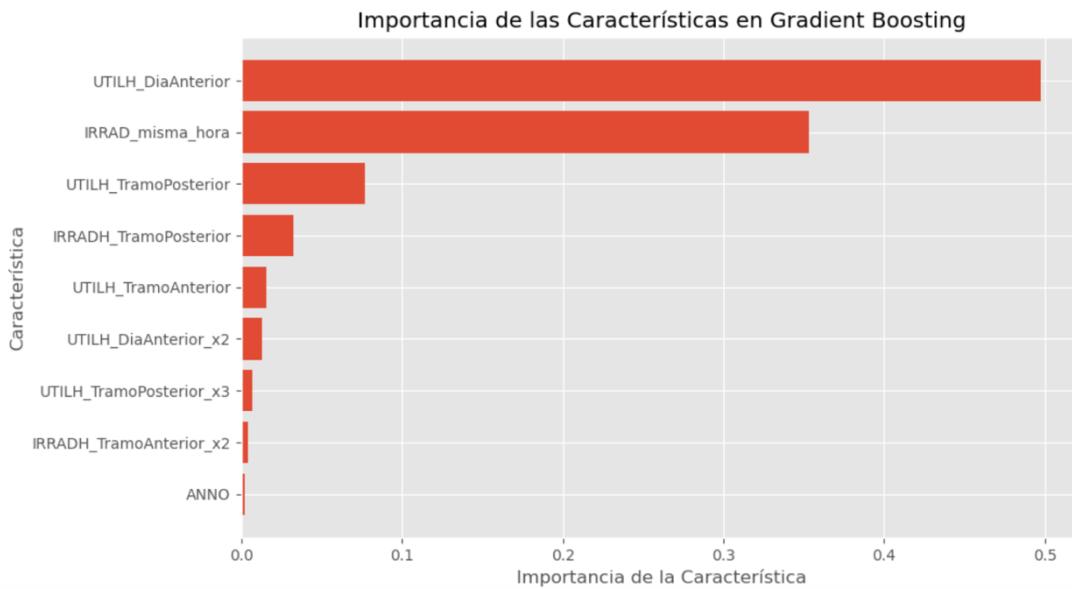


Ilustración 33. Representación de la importancia de las características en el modelo Gradient Boosting excluyendo las variables no significativas.

Veamos en último lugar, una comparación de las métricas de error registradas en la ‘prueba1’ y ‘prueba2’ (actual) de Gradient Boosting, lo cual queda representado en la Ilustración 34.

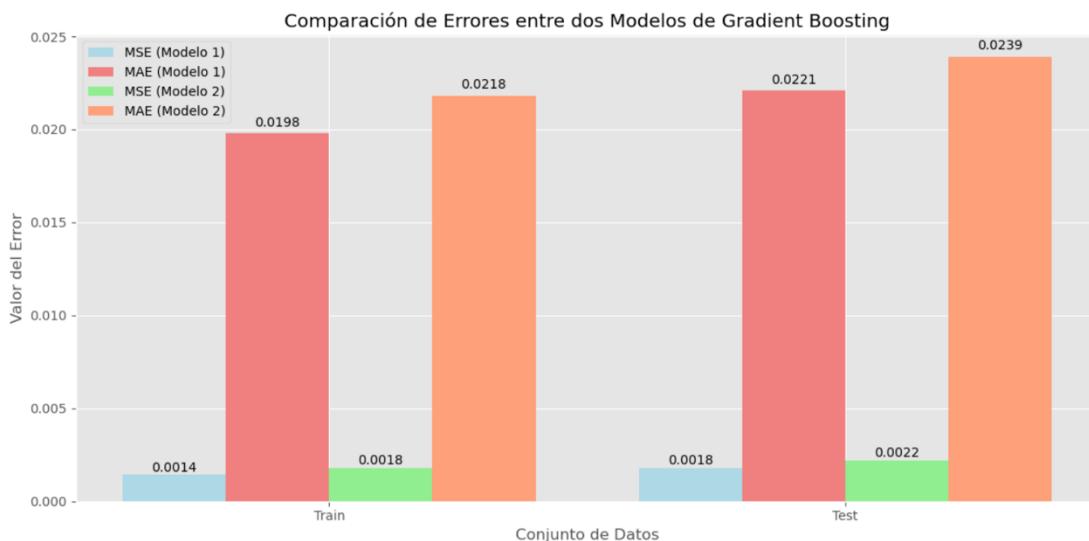


Ilustración 34. Comparación de las métricas de error en los modelos de Gradient Boosting.

En el anterior gráfico (Ilustración 34) se representa en azul y rojo los errores que cometíamos en el anterior modelo, mientras que en verde y naranja se representan los errores cometidos en este último con menos variables explicativas.

Se aprecia cómo en este segundo modelo el error se ha visto incrementado, tanto en el MSE como en el MAPE. Esto se deberá a que, pese a que la importancia del resto de variables explicativas no es muy alta, sí que aportan algo en el modelo.

En consecuencia, tras esta comparación de error, descartamos este segundo modelo y nos decantamos por el Gradient Boosting que empleaba todas las variables explicativas como entradas.

5. CONCLUSIONES

Una vez todos los modelos han sido entrenados, y tras presentar sus resultados individualmente, procedemos a compararlos y decidir cuál es el modelo óptimo para nuestro caso.

Estudiemos en primera instancia, los errores MSE obtenidos tanto en el conjunto de prueba como en el conjunto de entrenamiento para todos los modelos.

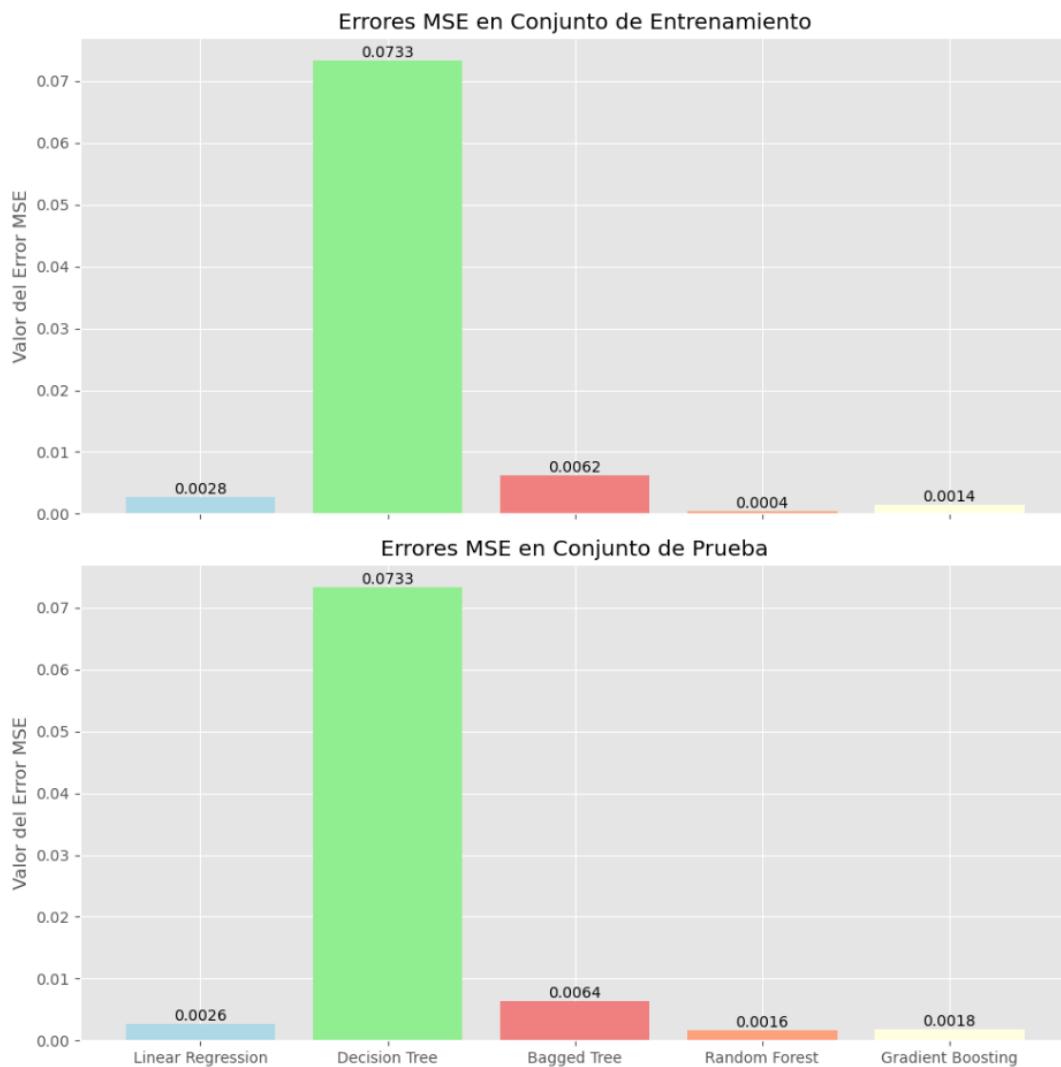


Ilustración 35. Comparación de métricas de error (MSE) en conjunto de entrenamiento y prueba para todos los modelos entrenados.

Sin lugar a dudas, la Ilustración 35 muestra como **el modelo con peor desempeño es el árbol de regresión sencillo**, con un MSE muy por encima al resto de casos.

En cuanto a los mejores modelos, encontramos Gradient Boosting y Random Forest. En este sentido, tanto en el conjunto de entrenamiento como en el de prueba **Random Forest se equivoca menos**, con tan solo un error MSE igual a 0.0016.

Igualmente, es necesario analizar otras métricas de error empleadas en regresión para obtener una conclusión más robusta. Continuemos por ello con el análisis de los valores MAE obtenidos, los cuales quedan representados gráficamente a continuación en la Ilustración 36.

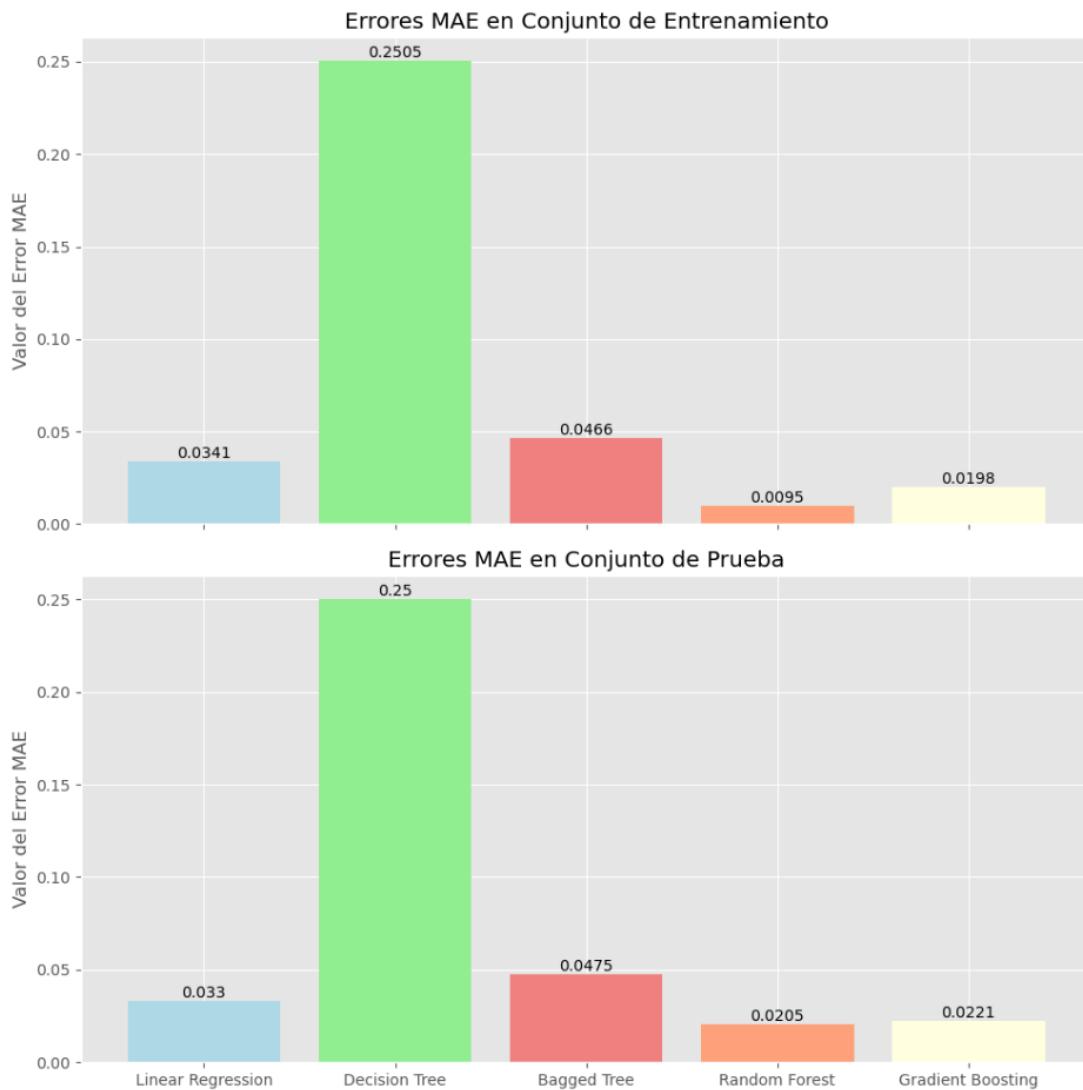


Ilustración 36. Comparación de métricas de error (MAE) en conjunto de entrenamiento y prueba para todos los modelos entrenados.

De nuevo, la Ilustración 36 nos muestra como **la peor opción es el árbol de regresión simple, y las mejores opciones Random Forest y Gradient Boosting**.

En cuanto al MAE vuelve a tener un error menor Random Forest frente a Gradient Boosting, no obstante, pese a que este segundo no tiene sobreaprendizaje, el primero presenta un ligero sobreajuste.

Independientemente, dado que en ambas métricas el menor error es el cometido con Random Forest, elegimos decantarnos por este modelo.

Es decir, la opción escogida es un Random Forest utilizando todas las variables explicativas creadas y ya existentes (21 variables descritas en el apartado 3.2 del presente trabajo) y con los hiperparámetros representados en la Ilustración 37, los cuales fueron escogidos mediante GridSearch y 10 folds.

```
'estimator__steps': [('RF', RandomForestRegressor(random_state=99
9))],
'estimator__verbose': False,
'estimator__RF': RandomForestRegressor(random_state=999),
'estimator__RF__bootstrap': True,
'estimator__RF__ccp_alpha': 0.0,
'estimator__RF__criterion': 'squared_error',
'estimator__RF__max_depth': None,
'estimator__RF__max_features': 1.0,
'estimator__RF__max_leaf_nodes': None,
'estimator__RF__max_samples': None,
'estimator__RF__min_impurity_decrease': 0.0,
'estimator__RF__min_samples_leaf': 1,
'estimator__RF__min_samples_split': 2,
'estimator__RF__min_weight_fraction_leaf': 0.0,
'estimator__RF__n_estimators': 100,
'estimator__RF__n_jobs': None,
'estimator__RF__oob_score': False,
'estimator__RF__random_state': 999,
'estimator__RF__verbose': 0,
'estimator__RF__warm_start': False,
'estimator': Pipeline(steps=[('RF', RandomForestRegressor(random_stat
e=999))]),
'n_jobs': -1,
'param_grid': {'RF__max_features': range(1, 6),
'RF__min_impurity_decrease': array([0.]),
'RF__min_samples_leaf': array([1, 6]),
'RF__min_samples_split': array([1, 6]),
'RF__n_estimators': range(20, 150, 5)},
'pre_dispatch': '2*n_jobs',
'refit': True,
'return_train_score': False,
'scoring': None,
'verbose': 0}
```

Ilustración 37. Hiperparámetros óptimos del modelo final seleccionado.

Es decir, el modelo Random Forest seleccionado está definido con los siguientes hiperparámetros óptimos para su entrenamiento: *max_features=1*, *min_impurity_decrease=0*, *min_samples_leaf=1*, *min_samples_split=2* y *n_estimators=100*. Estos valores indican que el modelo utiliza la totalidad de características en cada árbol, no aplica un umbral mínimo para la disminución de la impureza, no establece un número mínimo de muestras en las hojas y utiliza 100 árboles en el bosque.

Esta configuración, ante otros casos podría suponer un resultado con sobreaprendizaje, pues sugiere **un enfoque más flexible y complejo de lo normal**. Ahora bien, aquí es la configuración seleccionada debido al conjunto de datos tan diverso y grande del que partimos.

Para terminar de comprobar que el modelo final seleccionado “funciona” y predice correctamente, vamos a representar las predicciones con dicho modelo Random Forest y los valores reales de las utilizaciones a diferentes horas del día (Ilustración 38, Ilustración 39, Ilustración 40) para complementar las medidas de error presentadas en este análisis.

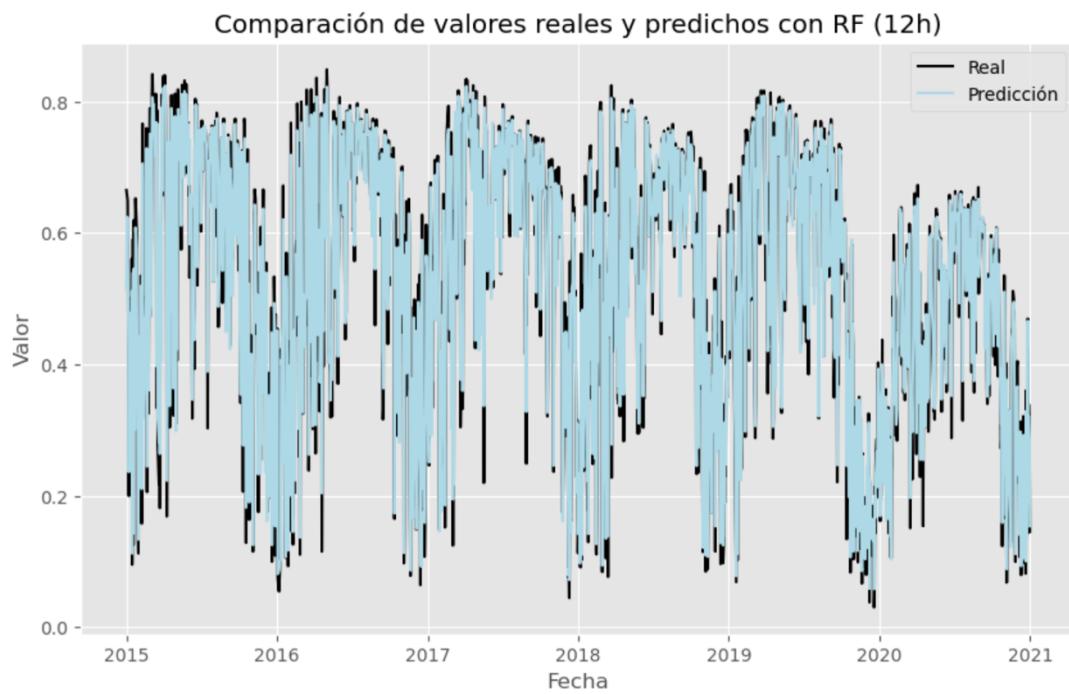


Ilustración 38. Comparación de los valores reales y predichos con modelo final Random Forest para las 12h.

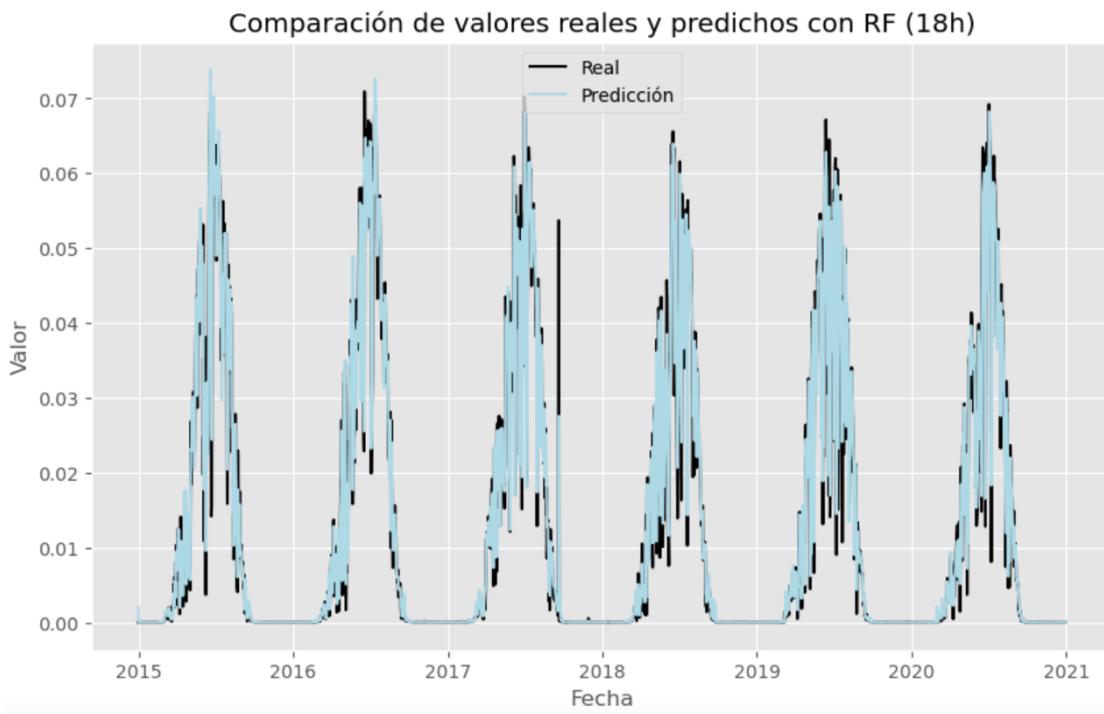


Ilustración 39. Comparación de los valores reales y predichos con modelo final Random Forest para las 18h.

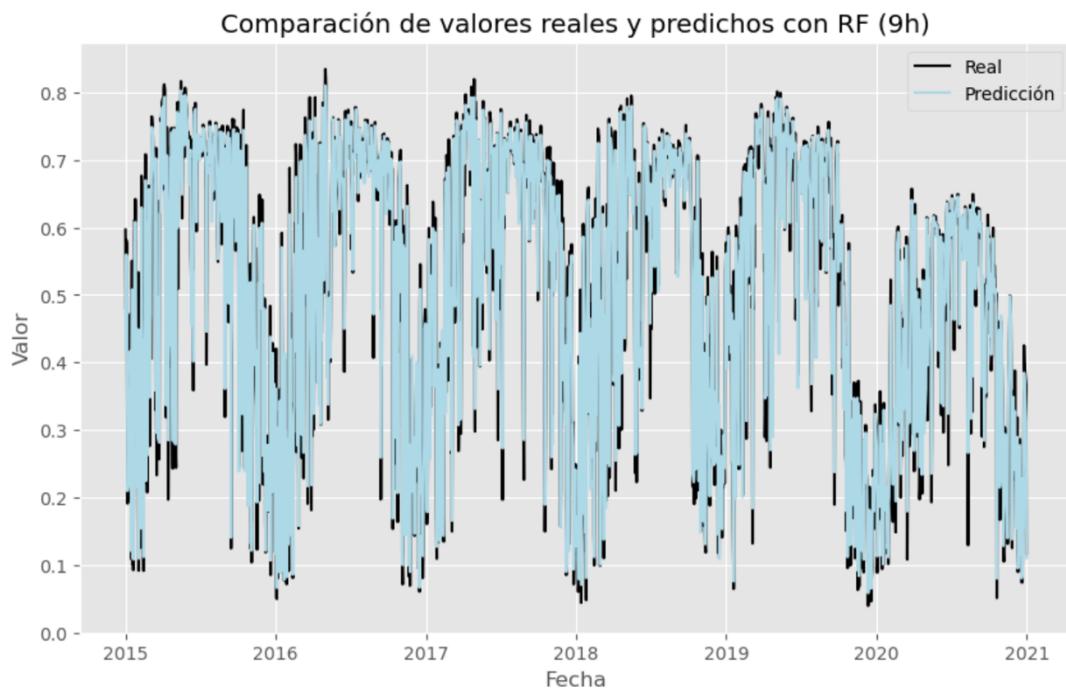


Ilustración 40. Comparación de los valores reales y predichos con modelo final Random Forest para las 9h.

En todos los casos (Ilustración 38, Ilustración 39, Ilustración 40) se observa como el error es mínimo, pues los valores reales y predichos por el modelo son prácticamente iguales e indistinguibles en dichas ilustraciones.

5.1. Comprobación final

Como hemos observado al comparar los distintos modelos entrenados, tanto en las métricas de error sobre el conjunto de entrenamiento como en el conjunto de prueba, el modelo óptimo resultó ser el Random Forest.

Sin embargo, es crucial verificar de manera rigurosa que este modelo funciona adecuadamente ante datos no observados. Por esta razón, se llevará a cabo un nuevo entrenamiento del modelo, manteniendo los hiperparámetros óptimos obtenidos previamente para el Random Forest, pero esta vez utilizando los conjuntos de entrenamiento y prueba que representan el 90% del conjunto de datos original.

Los restantes 10% del conjunto de datos se reservan para realizar predicciones y calcular el error cometido. Al realizar este procedimiento, se obtiene un error cuadrático medio (MSE) sobre el conjunto de prueba igual a 0.0016, y un error absoluto medio (MAE) igual a 0.021.

Estos valores son prácticamente idénticos a los obtenidos anteriormente (0.0016 y 0.0205, respectivamente). Por lo tanto, podemos concluir que el modelo Random Forest es robusto y generaliza bien a datos no observados, lo que nos permite validar su eficacia y confiar en su desempeño sin necesidad de realizar más pruebas.

6. RESULTADOS INTERMEDIOS NO ACEPTABLES

Finalmente, y dado que también se pide en el enunciado incluir casos en los que los resultados no sean buenos, vamos a obtener esta misma representación con el árbol de regresión simple, que era el modelo que nos proporcionaba un error significativamente más alto, tal y como hemos observado en las métricas anteriormente.

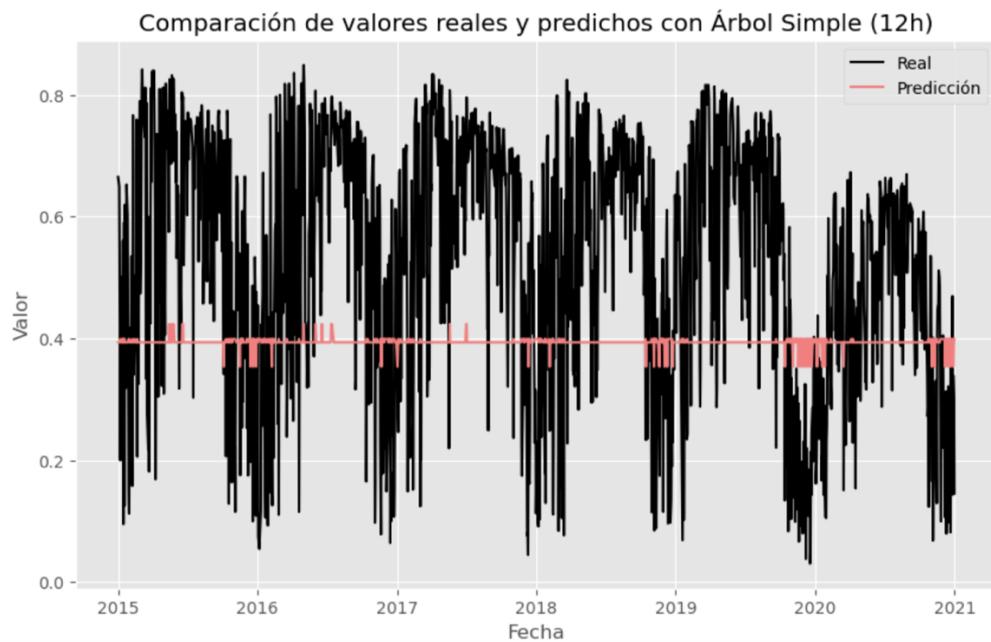


Ilustración 41. Comparación de los valores reales y predichos con modelo de peores resultados (árbol de regresión simple) para las 12h.

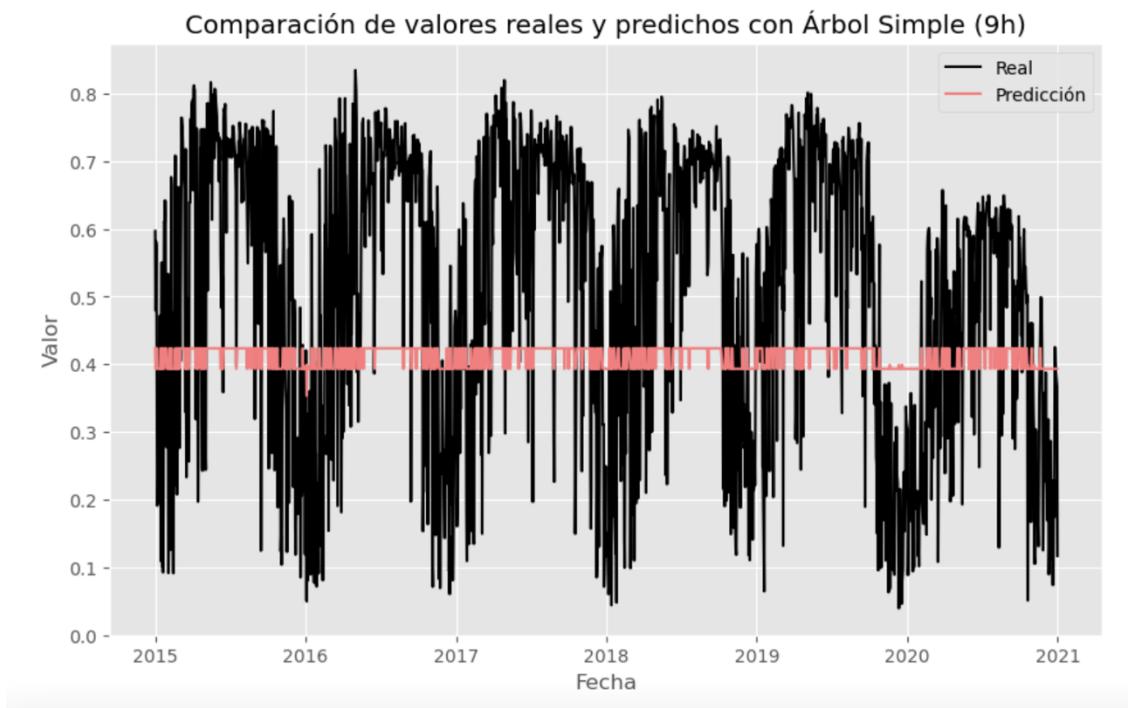


Ilustración 42. Comparación de los valores reales y predichos con modelo de peores resultados (árbol de regresión simple) para las 9h.

Este árbol de regresión simple que entrenamos al inicio de la práctica funciona muy mal tal y como queda representado en las figuras superiores (Ilustración 41 y Ilustración 42). En este sentido, el modelo asigna independientemente del tramo horario, una utilización media en torno al 0.4, pero no es capaz de aprender los patrones de los datos que, como se demostró en el EDA, cuentan tanto con estacionalidad diaria (en función de la hora el comportamiento es muy diferente), como estacionalidad anual (cambian los valores en función de la época del año).

Por tanto, una vez realizado el análisis completo de este problema y de los modelos entrenados para él, podemos concluir con lo mencionado anteriormente que el modelo que mejor funciona no siempre tiene porque resultar en el modelo más complejo teóricamente, si no en aquel que se ajuste mejor a la complejidad de los datos de tu problema. Es decir, en nuestro caso, aunque en iteraciones posteriores con modelos más complejos como Gradient Boosting esperábamos una mejora significativa de los resultados, hemos visto como esto no ocurría, si no que solo añadía complejidad al modelo no implicando en una mejora de ajuste al mismo.