

Assessment 3: Reinforcement learning

Machine Learning II

Máster en Big Data. Tecnología y Analítica Avanzada

Miguel Ángel Dávila Romero, Pablo Fernández Pita, Elena Cabrera Casquet

Índice

[1. Introducción 4](#_Toc166249326)

[1.1. Objetivos 4](#_Toc166249327)

[1.2. Metodología 4](#_Toc166249328)

[2. Análisis exploratorio de los datos 4](#_Toc166249329)

[2.1. Transformaciones y limpieza de datos 4](#_Toc166249330)

[2.2. Valores faltantes y detección de outliers 5](#_Toc166249331)

[2.3. Tipos de variables y codificación de variables categóricas 5](#_Toc166249332)

[2.4. Análisis exploratorio 6](#_Toc166249333)

[3. Bagging 9](#_Toc166249334)

[Métricas de rendimiento 9](#_Toc166249335)

[4. Conclusiones 11](#_Toc166249336)

Tablas

[Tabla 1. Métricas regression tree 9](#_Toc164098283)

[Tabla 2. Métricas bagged tree 10](#_Toc164098284)

[Tabla 3. Métricas random forest 12](#_Toc164098285)

[Tabla 4. Métricas de rendimiento de AdaBoost 14](#_Toc164098286)

[Tabla 5. Métricas de rendimiento de Gradient Boosting 16](#_Toc164098287)

[Tabla 6. Métricas de rendimiento para el XGBoost. 18](#_Toc164098288)

Figuras

[Figura 1. Boxplot de valores de utilidad e irradiación. 5](#_Toc164098092)

[Figura 2. Histograma para los valores de Utilidad e Irradiación 5](#_Toc164098093)

[Figura 3. Correlación entre los valores de Utilidad e Irradiación. 6](#_Toc164098094)

[Figura 4. Mapa de correlación entre variables. 6](#_Toc164098095)

[Figura 5. Series temporales por fecha, diferenciadas por hora para Utilidad e Irradiación. 7](#_Toc164098096)

[Figura 6. Serie temporal de utilidad por hora del día 8](#_Toc164098097)

[Figura 7. Promedio mensual de utilidad e irradiación. 8](#_Toc164098098)

[Figura 8. Utilidad e Irradiación mensual diferenciada por año. 9](#_Toc164098099)

[Figura 9. Importancia variables bagged trees 10](#_Toc164098100)

[Figura 10. Variabilidad bagged trees 11](#_Toc164098101)

[Figura 11. Residuos bagged tree 11](#_Toc164098102)

[Figura 12. Variabilidad random forest 12](#_Toc164098103)

[Figura 13. Residuos random forest 12](#_Toc164098104)

[Figura 14. Permutation importance random forest 13](#_Toc164098105)

[Figura 15. Dispersión de los residuos de entrenamiento en AdaBoost 14](#_Toc164098106)

[Figura 16. Dispersión de los residuos de test en AdaBoost 15](#_Toc164098107)

[Figura 17. Histograma y curva de densidad de los residuos en entrenamiento con AdaBoost 15](#_Toc164098108)

[Figura 18. Histograma y curva de densidad de los residuos en test con AdaBoost 15](#_Toc164098109)

[Figura 19. Dispersión de los residuos de entrenamiento en Gradient Boosting 17](#_Toc164098110)

[Figura 20. Dispersión de los residuos de test en Gradient Boosting 17](#_Toc164098111)

[Figura 21. Histograma y curva de densidad de los residuos en entrenamiento con Gradient Boosting 17](#_Toc164098112)

[Figura 22. Histograma y curva de densidad de los residuos en test con Gradient Boosting 17](#_Toc164098113)

[Figura 23. Residuos conjunto de entrenamiento para XGBoost. 19](#_Toc164098114)

[Figura 24. Residuos conjunto de test para XGBoost. 19](#_Toc164098115)

[Figura 25. Distribución de residuos conjunto de entrenamiento para XGBoost 19](#_Toc164098116)

[Figura 26. Distribución de residuos conjunto de test para XGBoost 20](#_Toc164098117)

[Figura 27. Importancia de características para XGBoost. 20](#_Toc164098118)

# Introducción

## Objetivos

El objetivo principal de esta práctica es utilizar un algoritmo de reinforcement learning para solucionar el problema planteado por el entorno de Gym elegido.

En este caso, el entorno elegido es Mountain Car. El objetivo que alcanzar en este entorno es manejar un pequeño coche para que alcance una bandera que se sitúa en lo alto de una montaña.

Si bien este entorno posee, dos versiones, una con un número de acciones continuas y otro con acciones discretas, se optó por la versión sencilla para reducir el tiempo de entrenamiento necesario.

## Descripción del entorno

El entorno escogido tiene las siguientes características:

* Espacio de observaciones:
  + Tenemos un espacio continuo con dos elementos: la posición del coche en el eje x y su velocidad. Estas dos variables son continuas y varían entre -1.2 y 0.6 en el caso de la posición, y entre -0.07 y 0.07 en el caso de la velocidad
* Espacio de acciones: como hemos comentado antes, el espacio es discreto y solo tiene 3 acciones deterministas:
  + Acelerar hacia la izquierda
  + Acelerar hacia la derecha
  + No acelerar

Dado una acción, el coche sigue las siguientes dinámicas de transición:

velocidadt+1 = velocidadt + (acción - 1) \* fuerza - cos(3 \* posiciónt) \* gravedad

posiciónt+1 = posiciónt + velocidadt+1

donde la fuerza = 0.001 y la gravedad = 0.0025. Las colisiones en ambos extremos son inelásticas con la velocidad establecida en 0 al colisionar con la pared.

## Final del episodio

El episodio termina cuando el coche llega a la bandera objetivo o se llega al episodio 200 sin alcanzar dicha bandera.

## Métodos de RL a aplicar

En este estudio, se ha optado por emplear el método Q-learning para abordar el problema. Esta decisión se fundamenta en la reputación de Q-learning como uno de los algoritmos más sencillos de implementar en el ámbito del aprendizaje por refuerzo. A diferencia de otros enfoques, como Reinforce, DQN o A2C, que, si bien son potentes y pueden resolver una amplia gama de problemas, también requieren mayores recursos computacionales. Por el contrario, Q-learning es más liviano en términos de demanda de recursos, lo que lo hace ideal para este proyecto. Con Q-learning, se confía en que se pueden lograr resultados significativos sin la carga adicional de complejidad computacional, lo que permite centrarse en la esencia del desafío y obtener resultados prometedores de manera más eficiente.

# Análisis exploratorio de los datos

En este apartado se explorará y analizará el conjunto de datos relacionados con la irradiación solar y la utilidad de sistemas fotovoltaicos que se han mencionado anteriormente. Mediante este análisis, se busca comprender, principalmente, las correlaciones existentes entre la irradiación solar y la eficiencia de la conversión de esta energía en electricidad, así como patrones que se puedan extraer de las demás variables implicadas.

## Transformaciones y limpieza de datos

Inicialmente, importamos los conjuntos de datos “G04A\_DATOS\_IRRAD.csv” y “G04A\_DATOS\_UTIL.csv”, los cuales contienen mediciones de irradiación y utilidad, respectivamente. Posteriormente, realizamos una fusión de ambos conjuntos en un único dataframe para facilitar el análisis. Con el objetivo de tener una variable a predecir y realizar un solo modelo para una única columna, simplificando así el proceso, se ha realizado un melt, La función pd.melt() convierte las columnas especificadas (['IRRADH00', 'IRRADH03', 'IRRADH06', 'IRRADH09', 'IRRADH12', 'IRRADH15', 'IRRADH18', 'IRRADH21']) en filas bajo dos nuevas columnas: HOUR y IRRAD\_VALUE. La columna HOUR indica la hora del día (extraída del nombre de la columna original), e IRRAD\_VALUE contiene los valores de irradiación correspondientes a esa hora. De forma análoga se realiza el proceso con la columna UTIL\_VALUE.

Por otro lado, cabe recalcar que se ha eliminado la columna del día de la semana, ya que esta no es relevante para el modelo, así como la fecha y se ha creado otra denominada “dia\_mes”, la cual llevará un conteo de los días del año y variará entre 1 y 365 o 366, si es bisiesto y facilitará el orden temporal al modelo.

## Valores faltantes y detección de outliers

En cuanto a los valores faltantes, no hay ninguno que sea nulo.

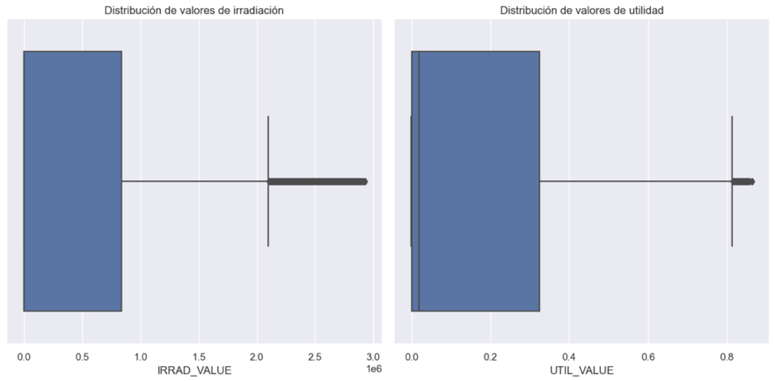
Por otro lado, se ha realizado un boxplot para visualizar potenciales outliers. En la siguiente imagen se puede ver como en el caso de la irradiación, a partir de se hayan bastantes. De forma análoga, si observamos la distribución para la utilidad, existen también muchos a partir del 80%.

Figura 1. Boxplot de valores de utilidad e irradiación.

Estos outliers son producto de la variabilidad intrínseca de los datos, por lo que hemos decidido no eliminarlos. Cuando llega la noche, la irradiación es nula, ergo la utilidad lo es también. Esto provoca que aproximadamente la mitad de los datos sean 0 y genera un desplazamiento de la mediana hacia este número. Las implicaciones que esto tiene, entre otras, es la aparición de outliers. Pero estos no han de tratarse diferente, sino como una característica intrínseca de los datos.

En la siguiente imagen se pueden ver dos histogramas para las variables mencionadas que corroboran la distribución de los datos según lo dicho.

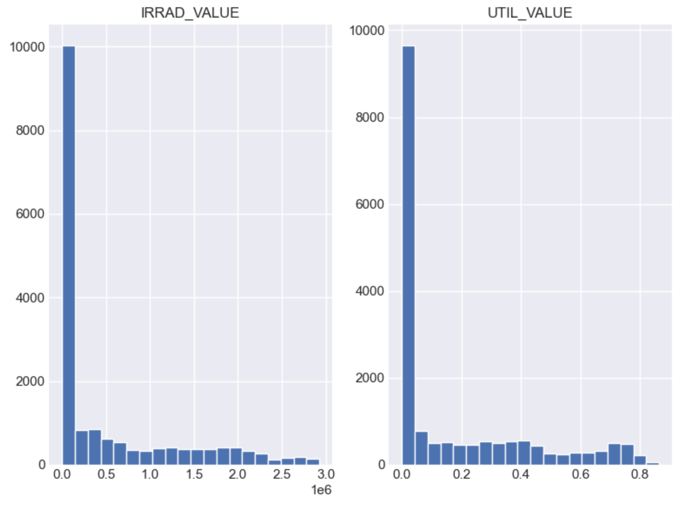


Figura 2. Histograma para los valores de Utilidad e Irradiación

## Tipos de variables y codificación de variables categóricas

En este caso, hemos tratado como categóricas las siguientes variables: ANNO\_x, MES\_x y DIA\_ x. Esto ha hecho que, a la hora de entrenar los modelos, hayamos incluido en el one hot encoding dichas variables.

## Análisis exploratorio

En primer lugar, se ha realizado un describe de los datos y un dato interesante que se puede extraer es que, durante todos los años, de media anual, se ha registrado que únicamente el 17% del total que le llega, se convierte en energía eléctrica, es decir, es utilizada. Un valor medio de utilidad alto indicaría una alta eficiencia y producción de energía de las instalaciones solares, lo que se espera durante los meses más soleados. España disfruta de una excelente irradiación solar, lo que la convierte en uno de los mejores lugares de Europa para la energía solar. Gracias a su posición geográfica disfruta de una elevada insolación durante todo el año y por tanto tiene un importante potencial para el desarrollo de la energía solar, tanto fotovoltaica como térmica. Sin embargo, las placas son capaces de convertir en electricidad el 17%. En los países más alejados del ecuador, este valor debería reducirse significativamente.

Por otro lado, mediante el coeficiente de correlación de Pearson, establecimos una fuerte correlación lineal entre la irradiación y la utilidad de los sistemas fotovoltaicos. Esto indica que a medida que aumenta la irradiación solar, también lo hace la cantidad de energía generada y aprovechada por los paneles solares.

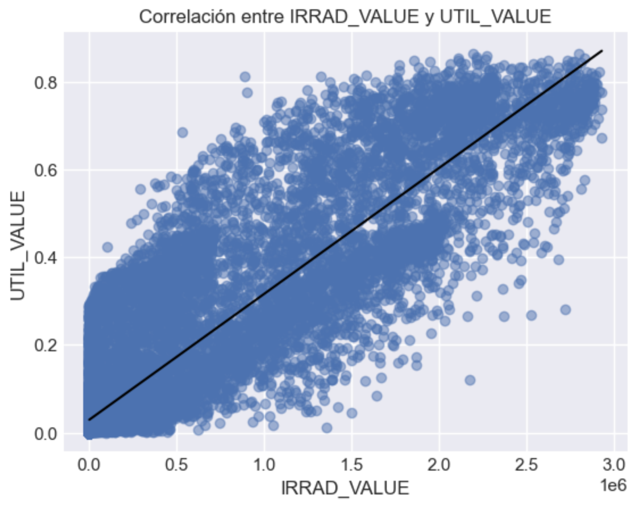


Figura 3. Correlación entre los valores de Utilidad e Irradiación.

Esta correlación es de 0,902.

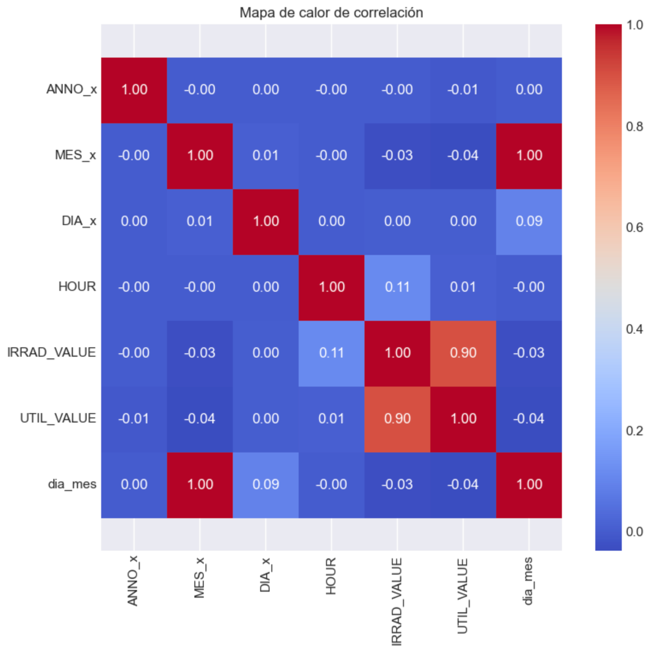
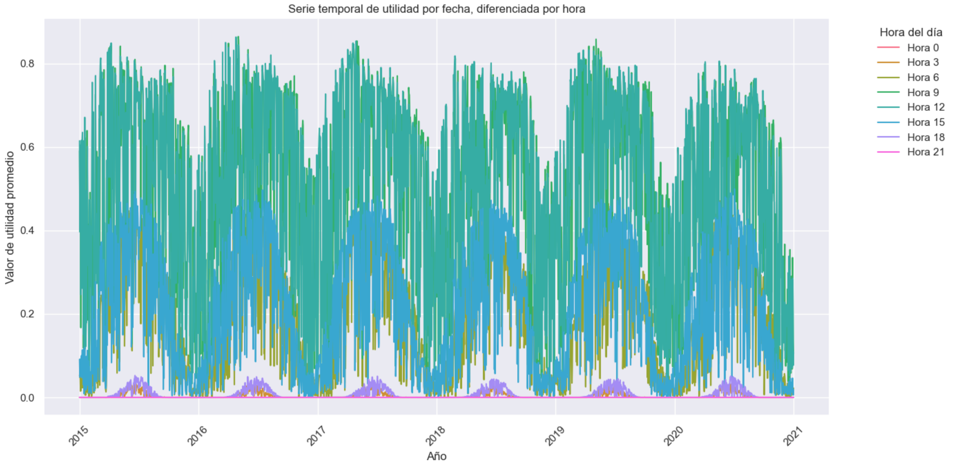
También se ha realizado un mapa de correlación para todas las variables, con el objetivo de visualizar potenciales relaciones, además de la obvia.

Figura 4. Mapa de correlación entre variables.

Tal y como se puede ver, aquí se muestran las correlaciones lineales directas entre las variables y, aparte de la obvia, no existen otras relaciones. Es por ello que en el siguiente apartado se analizarán en profundidad dichas variables.

En primer lugar, se comprueba si hay una periodicidad anual. Para ello haremos una distinción entre horas, ya que existe una gran variabilidad tanto en la utilidad como en la irradiación a lo largo del tiempo según se trate de una hora u otra, como consecuencia de las horas de sol.



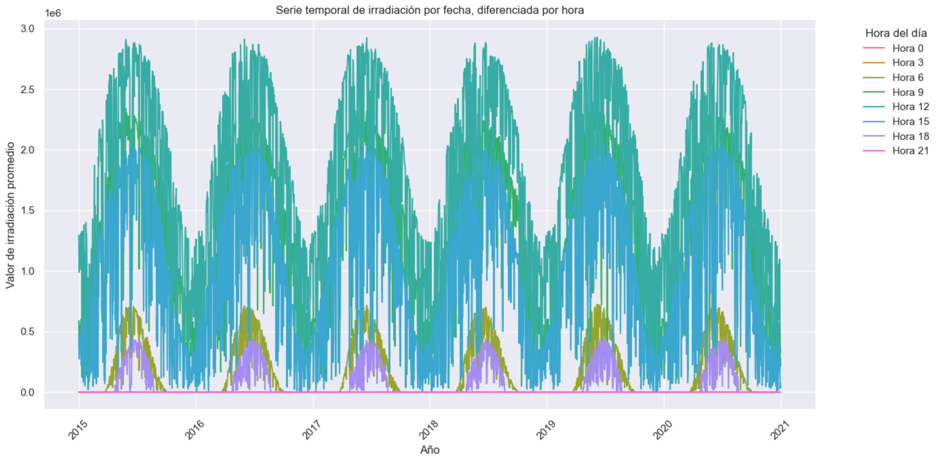


Figura 5. Series temporales por fecha, diferenciadas por hora para Utilidad e Irradiación.

Se puede observar que la tendencia entre estas dos variables es muy similar. Esto es lógico, ya que, si durante esa hora del día la zona está más expuesta a la irradiación solar, los paneles fotovoltaicos podrán aprovechar más su capacidad, ya que les llegan más rayos y por tanto podrán realizar una mayor conversión en electricidad.

Ahora veamos con más detalle en qué momentos del día se alcanzan estos picos de utilidad en promedio durante todo el año. Estas horas pueden variar ligeramente a medida que avanza el año, pero en esencia y en promedio se mantendrán dentro de un rango estrecho. Por eso podemos hacer la siguiente gráfica.

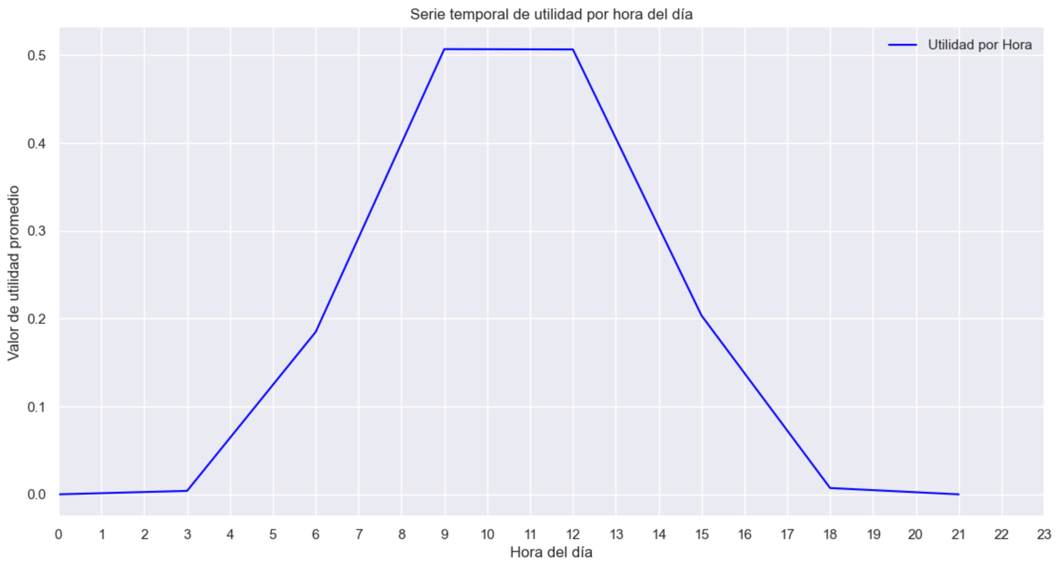


Figura 6. Serie temporal de utilidad por hora del día

Como es lógico, desde las 9 hasta las 12 es donde se haya el pico de utilidad, al existir mayor irradiación del sol.

Unimos los datos con un grupo en función de los meses promediando, así podemos ver el progreso a lo largo de los meses del año. Esto nos permite tener una visión promedio global de los datos.

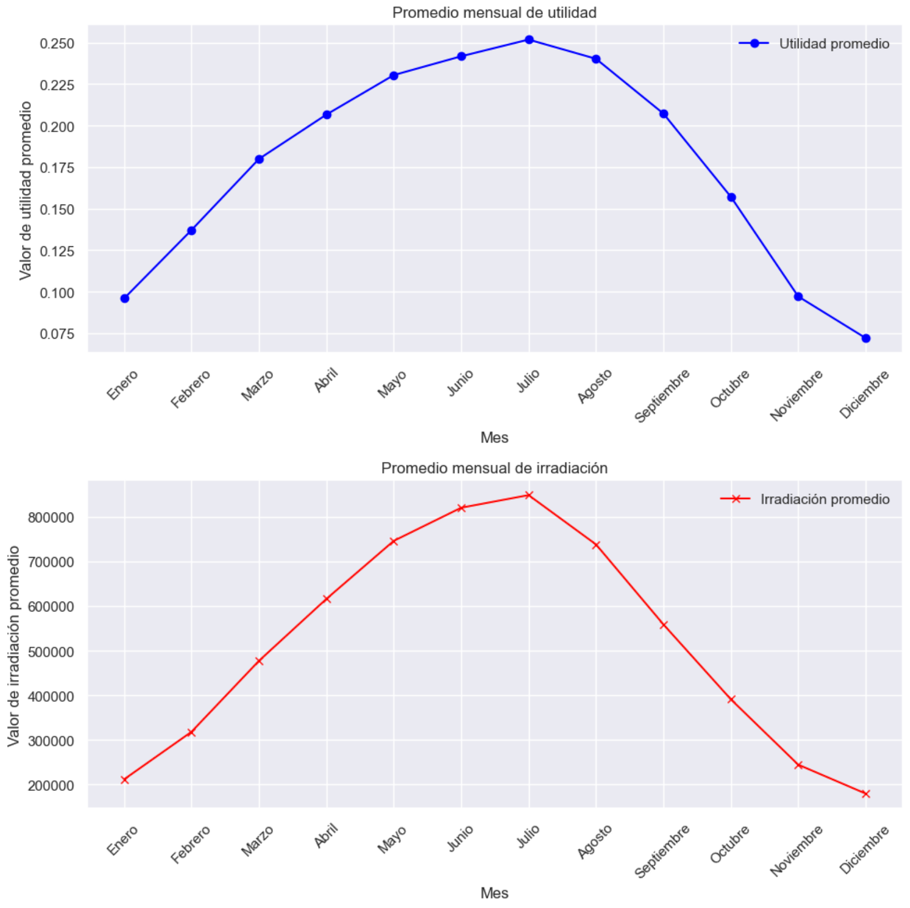


Figura 7. Promedio mensual de utilidad e irradiación.

Se puede ver como el pico se alcanza entre junio y julio, lo cual, en España, tiene mucho sentido.

Desglosemos ahora los datos por años, para ver si hay un año más relevante que otro o si hay mucha discrepancia entre estos.

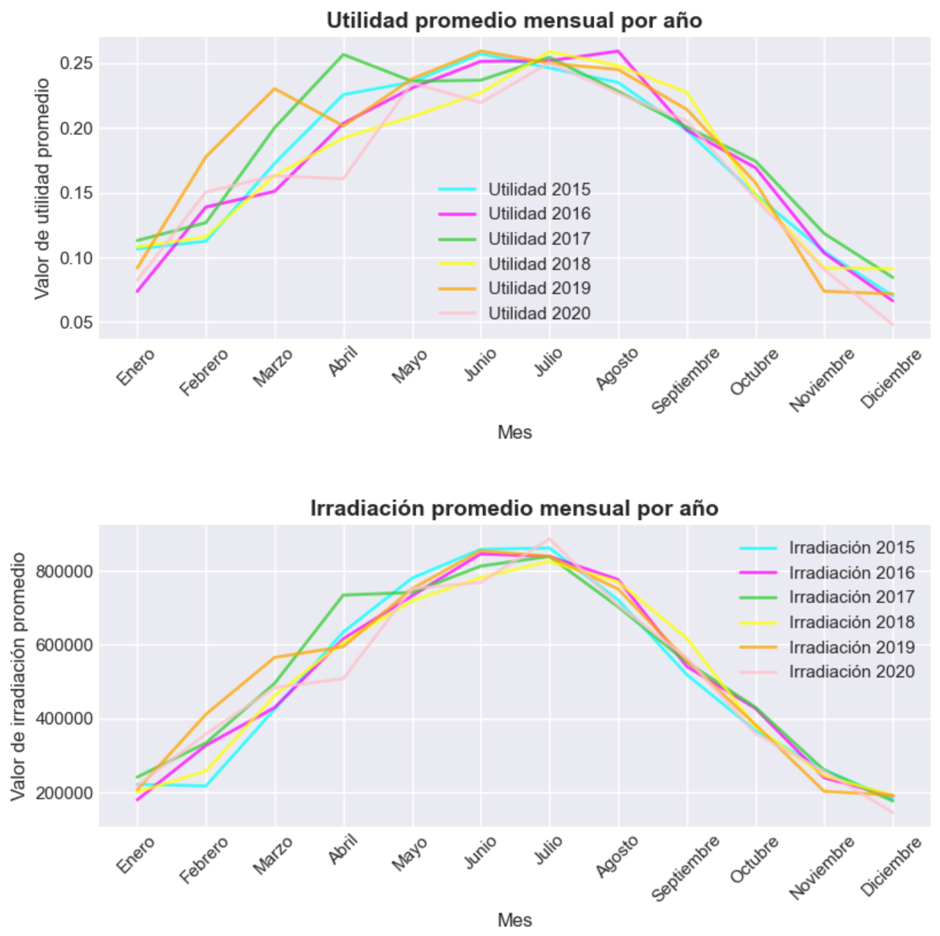


Figura 8. Utilidad e Irradiación mensual diferenciada por año.

Se puede observar como el valor de la utilidad promedio varía significativamente en el mes de abril para los años 2017 y 2020. También existe una discrepancia entre los años 2016 y 2019 para el mes de marzo.

# Bagging

Bagging es un método de ensamblado para reducir la varianza de un modelo de machine learning a partir de la agregación de muestras Bootstrap. Para resolver el problema propuesto, primero vamos a ajustar un modelo de regresión con un solo árbol. Este modelo sencillo nos dará un *baseline* desde el que comparar si los modelos siguientes mejoran o no la precisión de la predicción de nuestra variable de salida.

### Métricas de rendimiento

Con este modelo simple conseguimos las siguientes métricas:

|  |  |  |
| --- | --- | --- |
| **Métrica** | **Entrenamiento** | **Test** |
| MSE | 0.002012 | 0.005136 |
| RMSE | 0.04486 | 0.071665 |
| R2 | 0.966587 | 0.913117 |
| MAE | 0.021849 | 0.035214 |

Tabla 1. Métricas regression tree

Viendo estas métricas podemos decir que el primer intento de modelo es razonablemente bueno, y aunque quizás tengo un ligero sobre entrenamiento. Observando el diagrama de barras de la importancia de cada variable, nos hacemos una idea de las variables más relevantes. Como vimos en el EDA, la irradiancia y la utilización tienen una alta correlación, y este gráfico hace ver esa importancia. También aparecen como relevantes la hora y el día del año.

Partiendo de esta base, procedemos a ajustar un modelo de bagging usando ese mismo decision tree regressor óptimo con:

* Min\_impurity\_decrease: 0.0
* Min\_samples\_leaf: 5
* Min\_samples\_split: 2

Este modelo de bagged trees optimizados mejora nuestra precisión, y observando los residuos, el modelo parece poco sesgado y con residuos normales. Sin embargo, todavía tenemos otra opción, usar validación cruzada con grid search.

La cantidad de árboles a agregar se seleccionará con un barrido de 5 en 5 entre 20 y 160.

|  |  |  |
| --- | --- | --- |
| **Métrica** | **Entrenamiento** | **Test** |
| MSE | 0.00058 | 0.003941 |
| RMSE | 0.02409 | 0.062777 |
| R2 | 0.990365 | 0.933332 |
| MAE | 0.011934 | 0.030988 |

Tabla 2. Métricas bagged tree

Si bien las métricas con un árbol de decisión con gridseach ya era muy buenas, utilizando bagged tres hemos logrado unas incluso mejores. Además, obtenemos las ventajas ya comentadas utilizando bagging al combinar árboles reduciendo la varianza y sin tener miedo al sobre entrenamiento. Como se puede observar en los gráficos de residuos, si existe un ligero sesgo en los residuos que no sabemos aliviar con estos datos y este modelo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Figura 9. Importancia variables bagged trees

Una de las mayores ventajas de utilizar bagging trees es reducir la variabilidad de la variable de salida. Esta se debe a los cortes que haga cada árbol para predecir la variable de salida. Si observamos los árboles usados por este método de ensamblado mirando sus nodos y la profundidad del árbol generado.

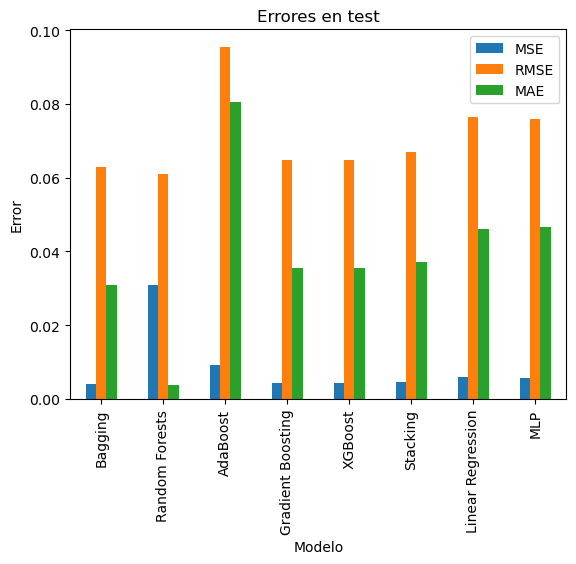
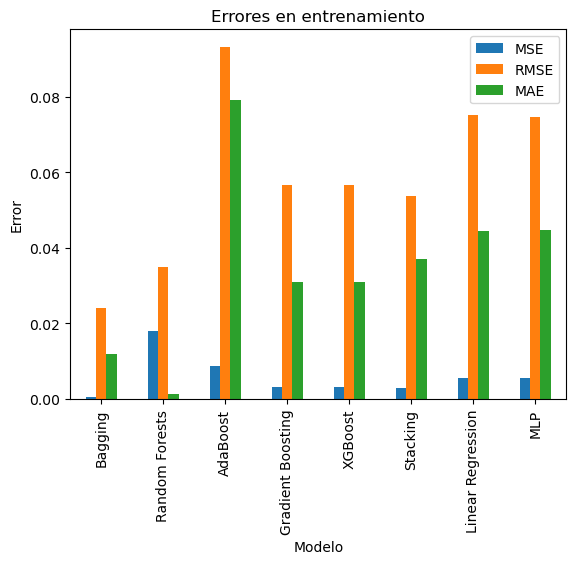
Como se mencionó antes, en los gráficos de los residuos se puede observar un sesgo en los residuos. Si bien la distribución de los residuos es normal, cuando observamos los puntos de las observaciones de arriba, podemos apreciar un sesgo. Con observaciones bajas los residuos tienden a ser negativos, con varios valores atípicos de infraestimación y con observaciones altas, pasa lo contrario. Esto parece indicar que tenemos una relación que no logramos capturar con el modelo.

Con estas métricas y gráficos podemos decir que el modelo obtenido es razonablemente bueno, pero siendo conscientes de que existe un sesgo y quizás necesitásemos un modelo capaz de encontrar relaciones no lineares en los datos o un dataset más rico.

# Conclusiones

Una vez hemos recogido los resultados de todos los modelos entrenados, procedemos a exponer una conclusión sobre los mismos.

En los siguientes gráficos de barras, podemos observar las métricas obtenidas por cada uno de los modelos en entrenamiento y test.



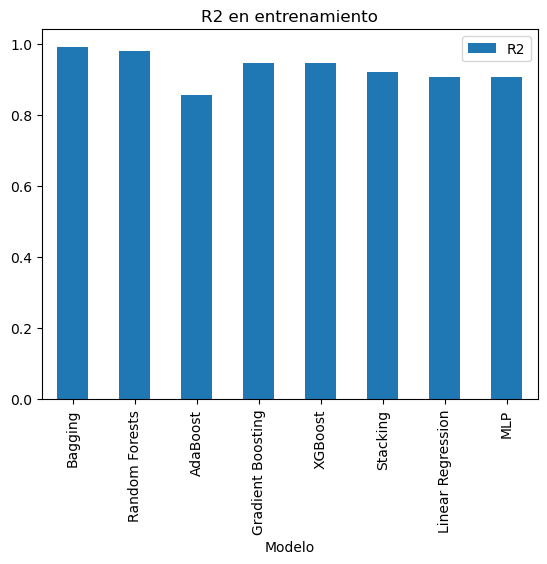
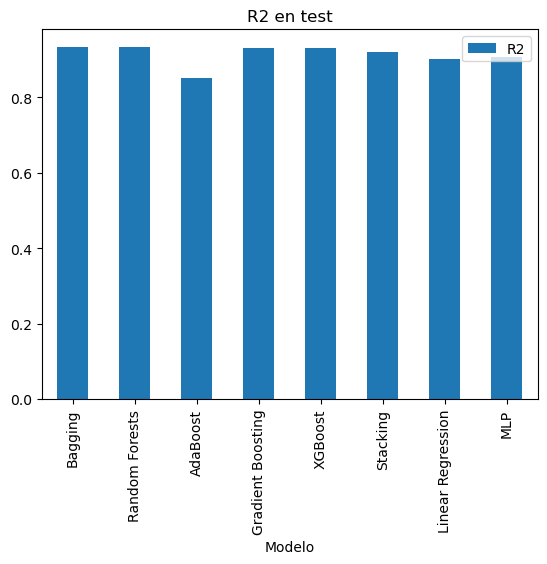


Figura 28. Métricas de cada modelo para entrenamiento y test

Como se puede apreciar, el modelo que mejor resultados ofrece es el de Bagging. Sin embargo, destaca un overfitting bastante notorio si nos fijamos en los valores de error, no siendo prácticamente perceptible en el valor del coeficiente R2.

Random Forests tiene un comportamiento similar, con un muy buen rendimiento en entrenamiento que disminuye bastante para datos de test.

Por tanto, si tuviéramos que quedarnos con un algoritmo, quizás sería mejor opción Gradient Boosting o XGBoost que, aunque presentan rendimientos algo inferiores, no sufren de sobreaprendizaje y, por tanto, podemos estar seguros de que las estimaciones de test son más realistas con respecto al comportamiento futuro del modelo con nuevos datos: es decir, son modelos más robustos.

No obstante, en el caso de que nuestra prioridad fuera la interpretabilidad, es probable que un modelo más sencillo, dentro de que sea un ensemble, podría ser bagging, a pesar del sobreaprendizaje al que se ha visto sometido. Usar un modelo base interpretable como la regresión lineal sería una aproximación extrema a esta idea, pero podría ser de utilidad en algunos casos, teniendo en cuenta que los resultados no son malos.

Una conclusión que sí queda clara a la vista de los datos es que los modelos ensemble ofrecen rendimientos superiores a los modelos base entrenados de manera individual. Por tanto, debemos considerarlos como una opción a tener en cuenta siempre que nos enfrentemos a un problema, aunque a simple vista parezca que, como en este caso, la relación entre las variables permita un buen rendimiento de modelos sencillos.