Generación eléctrica en base a fenómenos meteorológicos

Jesús Pardos Sotodosos
https://github.com/sotodosos

Contenido

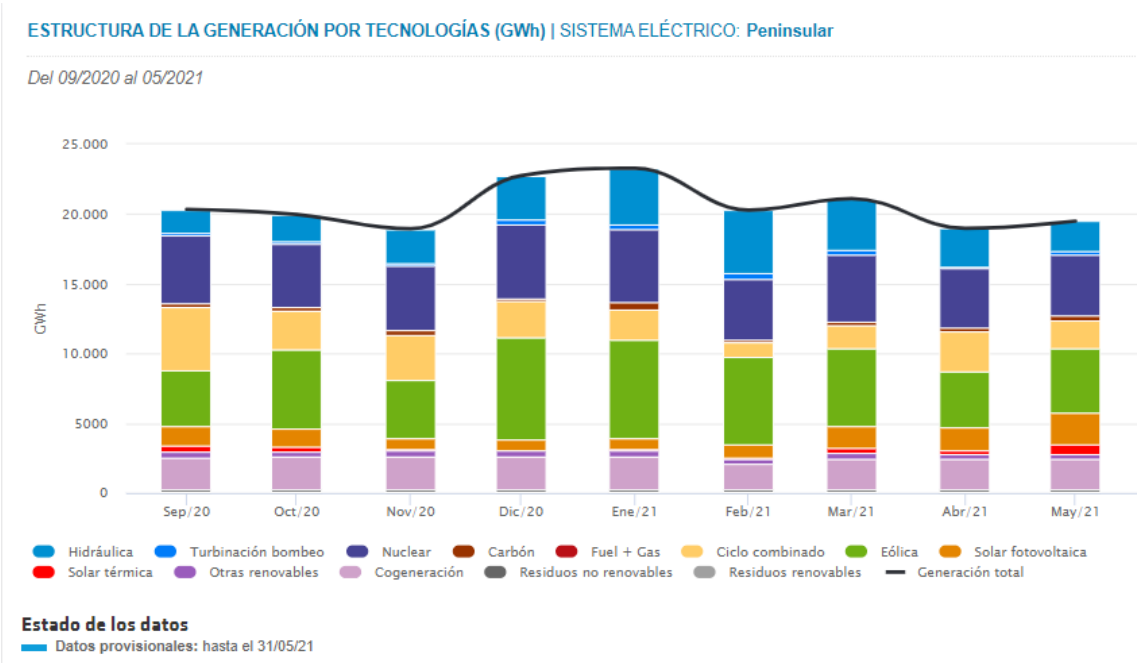
Objetivo

Con la situación actual del cambio climático que provoca graves inestabilidades meteorológicas y la necesitada cada vez mayor de energía eléctrica en detrimento de fuentes de energía fósiles, quise centrar mi proyecto en realizar una herramienta para la predicción de la energía eléctrica en función de los fenómenos meteorológicos, en la búsqueda de información relacionada con este desarrollo no encontré nada similar para datos de España.

Quiero destacar que este proyecto no tiene en cuenta las prioridades de las empresas de generación ni como gestionan la venta de la energía generada y la comercialización de esta, y no se va a tratar la demanda, ni el mercado eléctrico.

Por lo anterior la idea principal de este desarrollo es predecir la generación eléctrica y su distribución en función de la tecnología de generación, únicamente basándonos en los datos meteorológicos y fechas. Como intuitivamente pensamos que los fenómenos meteorológicos tienen una mayor relación con las energías renovables nos vamos a centrar en este tipo de tecnología de generación.

El primer análisis realizado fue comprobar que energías renovables son las más interesantes para su análisis, descarté las tecnologías de generación hidráulicas ya que realmente no tiene una relación directa con los fenómenos meteorológicos a corto plazo ya que se almacena el agua de las precipitaciones y posteriormente se utiliza su energía potencial almacenada en embalses, por lo que finalmente me decidí a centrarme en la energía solar fotovoltaica y eólica que además suponen gran parte de la generación, tal y como se puede ver en las propias graficas de Red Eléctrica de España (https://www.ree.es/es/datos/generacion/estructura-generacion).



Como fuentes de datos se van a utilizar:

Red Eléctrica de España (REE)

Se trata del transportista y operador del sistema eléctrico español, esta empresa pública controla toda la infraestructura y transporte de la energía eléctrica además es el encargado de integrar los distintos tipos de energía y controlar la producción y demanda, posee una API (https://www.ree.es/es/apidatos) con la que se puede obtener la información que se muestra en su página web, el inconveniente es que la información de la generación eléctrica a nivel diario solo está disponible por sistema eléctrico.

Agencia Estatal de Meteorología (AEMET)

Es el Servicio Meteorológico Nacional y Autoridad Meteorológica del Estado, una de sus principales objetivos es predecir y vigilar fenómenos meteorológicos adversos, este organismo posee un portal de datos abiertos (https://opendata.aemet.es/), al que se puede acceder vía API para obtener la información meteorológica en su red de estaciones de medición.

Con todo lo anterior el desarrollo de este proyecto se va a centrar en:

- Predicción de la generación total
- Predicción de la distribución entre energía renovable y energía no renovable.
- Predicción del porcentaje de energía Solar y Eólica.

Además, se generará un frontal donde el usuario podrá consultar la distribución y la generación eléctrica dados unos valores meteorológicos. Y se podrá consultar el histórico de esta información de manera interactiva para los tipos de energías a estudio.

Entorno y Requisitos

Para una mayor comprensión del resto del proyecto a continuación se detalla la organización del repositorio y sus requisitos de ejecución.

Organización del Repositorio

El proyecto se encuentra ubicado en el repositorio público de Github:

https://github.com/sotodosos/TFM_Generacion_electrica_AEMET

Dentro de este repositorio encontramos la siguiente estructura de directorios:

- Data:
  - REGION_REE: Información de las regiones definidas por REE, su sistema eléctrico y su identificador único, esta información ha sido obtenida de https://www.ree.es/es/apidatos.
  - calendario.csv: Calendario laboral del ayuntamiento de Madrid, contiene datos históricos desde 2013 a 2021, esta información ha sido obtenida de https://datos.gob.es/en/catalogo/l01280796-calendario-laboral

- Doc:
  - environment.yml: listado de paquetes instalados para replicar el entorno con Conda.
  - requirements.txt: listado de paquetes instalados para replicar el entorno con Pip.

- Python:
  - 1-TFM Read data.ipynb: Lectura y tratamiento de datos de REE, AEMET y calendario laboral de Madrid
  - 2-TFM Analysis and preprocesing.ipynb: Limpieza de datos, análisis estadístico y preprocesamiento de datos
  - 3-TFM Model_Generation.ipynb: tratamiento de variables, entrenamiento y creación del modelo de generación eléctrica
  - 4-TFM Model_Renov.ipynb: tratamiento de variables, entrenamiento y creación del modelo de distribución de energía renovable
  - 5-TFM Model_Tech.ipynb: tratamiento de variables, entrenamiento y creación del modelo de distribución de energía solar y eólica.
  - 6-Execute_Streamlit.ipynb: Ejecución de app.py desde Google Colab generando una URL publica.
  - app.py: Aplicación streamlit para la creación del frontal de la aplicación.
  - Lectura_AEMET_REE.py: Librería desarrollada para almacenar las clases de Aemet y REE y sus métodos para la lectura y tratamiento de su información.
  - utils.py: Librería con las funciones más importantes del desarrollo.

- Models:
  - best_model_Generation.sav: Mejor modelo predicción de la generación eléctrica total.

- best_model_renovable.sav: Mejor modelo predicción del porcentaje de energía renovable producida sobre el total del día y sistema.
- best_model_tech.sav: Mejor modelo predicción del porcentaje de energía solar fotovoltaica y eólica sobre el total del día y sistema.

Además de las carpetas descritas anteriormente existe una carpeta no subida al repositorio '*./API*', donde se almacena el fichero con la clave del API de Aemet, esta es la carpeta por defecto desde donde se lee este fichero, pero se puede modificar en la creación del objeto de la clase Ingestion_AEMET. Esta clave se puede obtener dándose de alta en Opendata Aemet, se recibe en aproximadamente 3 días en la dirección de email usado para darse de alta la clave para poder utilizar la API, la URL para darse de alta es la siguiente: *https://opendata.aemet.es/centrodedescargas/altaUsuario*

Requisitos de ejecución

En una primera versión como herramientas de desarrollo se utilizó Google Colab, para un mejor desarrollo y una mejor replicabilidad se continuo el desarrollo usando PyCharm Community y jupyter-notebook sobre un entorno Conda.

Se puede replicar el entorno de desarrollo del proyecto utilizando los siguientes ficheros de paquetes de Python:

- Conda: *'./Doc/environment.yml'*
- Pip: *'./Doc/requirements.txt'*

Modelo de Datos

REGION_REE

NOMBRE	REGION_REE
DESCRIPCION	Información de las regiones definidas por REE, su ámbito y su identificador único
ORIGEN	https://www.ree.es/es/apidatos
VARIABLES	
Region	Descriptivo de la región
geo_limit	Sistema eléctrico al que pertenece la región, puede tomar los valores: ▪ peninsular ▪ canarias ▪ baleares ▪ ceuta ▪ melilla ▪ ccaa
geo_id	Identificador numérico único de la región

calendario.csv

NOMBRE	calendario.csv
DESCRIPCION	Calendario laboral del ayuntamiento de Madrid, contiene datos históricos desde 2013 a 2021
ORIGEN	https://datos.gob.es/en/catalogo/l01280796-calendario-laboral
VARIABLES	
Dia	Fecha en formato DD/MM/AAAA
Dia_semana	Descriptivo del día de la semana (Ej. :'lunes')
laborable / festivo / domingo festivo	Descriptivo del tipo de día, puede tomar los siguientes valores: ▪ laborable ▪ festivo ▪ domingo ▪ sábado
Tipo de Festivo	Ámbito de la festividad, puede tomar los siguientes valores: ▪ Festivo nacional ▪ Festivo de la Comunidad de Madrid ▪ Festivo local de la ciudad de Madrid
Festividad	Descriptivo de la festividad (Ej.: 'Año Nuevo')

1_weather.csv

NOMBRE	1_weather.csv
DESCRIPCION	Información meteorológica leída a través de la API de AEMET además se enriquece con los campos Holiday y weekday obtenidos del calendario laboral del ayuntamiento de Madrid, esta información es generada en el notebook 1-TFM Read data.ipynb
ORIGEN	https://opendata.aemet.es/
VARIABLES	
fecha	Fecha de la lectura en formato DD/MM/YYYY.
indicativo	Identificador único de la estación de AEMET donde se ha realizado la lectura.
nombre	Nombre de la estación de AEMET.
provincia	Provincia donde se encuentra situada la estación.
altitud	Altitud a la que se encuentra la estación en metros.
tmed	Temperatura media del día en grados Celsius.
prec	Precipitaciones totales registradas en el día en mm agua acumulada.
tmin	Temperatura mínima del día en grados Celsius.
horatmin	Hora a la que se produce la temperatura mínima en formato HH:MM en 24h.
tmax	Temperatura máxima del día en grados Celsius.
horatmax	Hora a la que se produce la temperatura máxima en formato HH:MM en 24h.
dir	Punto cardinal del que proviene el viento
velmedia	Velocidad media del viento en km/h
racha	Velocidad máxima del viento en km/h
horaracha	Hora a la que se produce la velocidad máxima del viento en formato HH:MM en 24h.
sol	Número de horas de sol.
presMax	Presión atmosférica máxima en bar
horaPresMax	Hora a la que se produce la máxima presión en formato HH:MM en 24h.
presMin	Presión atmosférica mínima en bar
horaPresMin	Hora a la que se produce la máxima presión en formato HH:MM en 24h.
Holiday	Indicador de si el día es un festivo nacional (1) o no lo es (0)
Weekday	Día de la semana en formato numérico (0-6)

1_ree_system.csv

NOMBRE	1_ree_system.csv
DESCRIPCION	Información de la generación eléctrica leída a través de la API de REE, generada en el notebook 1-TFM Read data.ipynb
ORIGEN	https://www.ree.es/es/apidatos
VARIABLES	
value	Valor de la generación eléctrica en Mwh
percentage	Porcentaje sobre el total de la generación eléctrica
datetime	Fecha en formato UTC de la lectura de la generación eléctrica
title	Tecnología de generación
type	Tipo de generación, puede tomar los valores: ▪ Renovable ▪ No Renovable
system	Sistema eléctrico de la lectura puede tomar los valores: ▪ peninsular ▪ canarias ▪ baleares ▪ Ceuta ▪ melilla

Metodología

EL primer modelo viable que se genero fue una red neuronal aplicando Deep Learning, pero los resultados obtenidos no fueron los esperados, debido a esto gran parte del proyecto se ha realizado utilizando Google Colab por la posibilidad de tener disponibilidad de un entorno con TPU.

Finalmente, el desarrollo se ha realizado en jupyter-notebook en un entorno Windows y las librerías utils.py, Lectura_AEMET_REE.py, así como la aplicación app.py se han desarrollado utilizando PyCharm Community.

Lectura del dato

El primer paso que se realiza es la lectura de datos de AEMET, REE y del calendario laboral, para ello se utilizan la librería Lectura_AEMET_REE, de la cual se utilizan las clases creadas para la lectura de datos desde las API´s AEMET y REE, Ingestion_AEMET y Ingestion_REE.

AEMET

Para la lectura de los datos de AEMET, se toman como datos de entrada una fecha de inicio y una fecha fin de la lectura, y se obtiene un DataFrame con las lecturas de las observaciones meteorológicas de todas las estaciones entre las 2 fechas dadas. Para poder ejecutar el método de lectura de datos (read_weather_dates) de Ingestion_AEMET es necesario tener la clave del API de AEMET, la cual se puede obtener desde su portal de Open Data, el proceso de lectura controla que no se produzcan errores por alcanzar el máximo número de peticiones al API por minuto, realizando un sleep de 56 segundos cuando se alcanza este límite, esto provoca que los tiempos de lectura sean más largos.

En la creación del objeto de la clase AEMET se puede definir las rutas que usarán los métodos de esta clase para la lectura de datos y su almacenamiento, por defecto usa las rutas '../API/' y '../Data'.

Con el fin de mejorar el modelo a la información leída por AEMET, que serán las variables de entrada de nuestro modelo, se le añade la información de los festivos nacionales a partir del fichero calendario.csv. Además, se crea un campo con el día de la semana en formato numérico ya que como parece intuitivo pensar la generación de la electricidad depende mucho de sí estamos tratando un Domingo o un lunes. Esta información se almacena en 1_weather.csv.

REE

La lectura de los datos de la generación eléctrica se realiza a través del método read_ree_dates de la clase Ingestion_REE dada una fecha de inicio y una fecha de fin de lectura, este método utiliza los id´s de regiones cuyo geo_limit es 'ccaa' de REGION_REE para realizar request a la API de REE y obtener el json con los valores de la generación, posteriormente da formato de pandas Dataframe a los datos, este paso se realiza para cada sistema eléctrico de REE. Esta información se almacena en 1_ree_system.csv.

Limpieza y adecuación

En la adecuación de los datos para ser usados por los modelos el primer paso que se realiza es convertir a numéricos los datos leídos de AEMET (1_weather.csv) para ellos se sustituye el carácter ',' por '.' De las variables que nos van a ser de utilidad, para los datos de REE (1_ree_system.csv) se renombran los campos los campos para que sean más entendibles realizando la siguiente transformación:

- Value → Generacion_Mwh
- title → Tecnologia
- type → Renov_norenov

Posteriormente se eliminan los registros cuya fecha no esté informada y se rellenan a 0 los valores sin informar de generación eléctrica.

Eliminación de variables no usadas

Para evitar usar variables que no tiene importancia para el modelo se eliminan las siguientes variables:

- altitud: Representa la altitud de cada estación, como posteriormente se agregará la información a nivel sistema eléctrico no nos aporta nada esta variables, además en la mayoría de casos se encuentra sin informar.
- dir: Representa la dirección del viento de cada estación, ocurre lo mismo que para la altitud, no podemos utilizarlo ya que en muchas de las estaciones no se mide y no se puede unificar de manera correcta a nivel de sistema eléctrico.
- horaPresMax, horaPresMin, horaracha, horatmin, horatmax: estas variables hacen relación a la hora del día en el que se producen ciertos fenómenos, como se va a agrupar el dato a nivel sistema eléctrico, esta información no va a ser tratada, además nuestros datos están a nivel diario por lo que en principio no parece necesaria.
- percentage: Representa el % que supone cada registro sobre el total de la energía eléctrica, este campo se va a eliminar por que posteriormente se va a trasponer la información por tecnología de generación.

Información al mismo nivel

Con el fin de unir la información de REE y AEMET, debemos dejar la información al mismo nivel de agrupación, que en este caso será al nivel en el que se encuentra el dato de REE, es decir, a nivel de sistema eléctrico y fecha.

Por lo anterior se agrupar pasar los datos de AEMET, que se encuentran a nivel provincia, estación meteorológica y fecha.

Para intentar que el dato siga siendo real y siguiendo la lógica de cada valor, se realiza una primera agrupación por provincia y fecha recalculando las variables como se detalla a continuación:

- tmax, presMax, racha: Estas variables representan valores máximos de cada estación, por lo que para obtenerlas a nivel de provincia nos quedamos con el valor máximo de las estaciones de cada provincia por fecha.

- tmin, presMin: Estas variables representan valores mínimos de cada estación, por lo que para obtenerlas a nivel de provincia nos quedamos con el valor mínimo de las estaciones de cada provincia por fecha.
- prec, tmed, sol, velmedia: Para estas variables que representan valores medios o acumulados nos vamos a quedar con el valor medio de todas las estaciones de la provincia.

Una vez tenemos el dato a nivel de provincia y fecha, se asigna a cada provincia el sistema eléctrico al que pertenece y ahora sí, se agrupa el dato por sistema eléctrico realizando la media de cada variable, por si hubiese variables para las que no se obtengan datos, rellenamos con 0.

Obtenemos el valor medio de cada variable ya que queremos que todas las provincias tengan el mismo peso en nuestro modelo, si realizásemos el mismo tratamiento que el seguido para la agrupación por provincia, daríamos más importancia a las provincias con clima más extremo, por ejemplo, nunca se tendría en cuenta el valor de la temperatura máxima de Teruel o la mínima de Sevilla.

Uno de los inconvenientes de realizarlo de esta manera es que damos por hecho que la generación eléctrica es similar en todas las provincias, pero como no disponemos del dato de generación de cada provincia ni de cada comunidad lo realizamos de este modo.

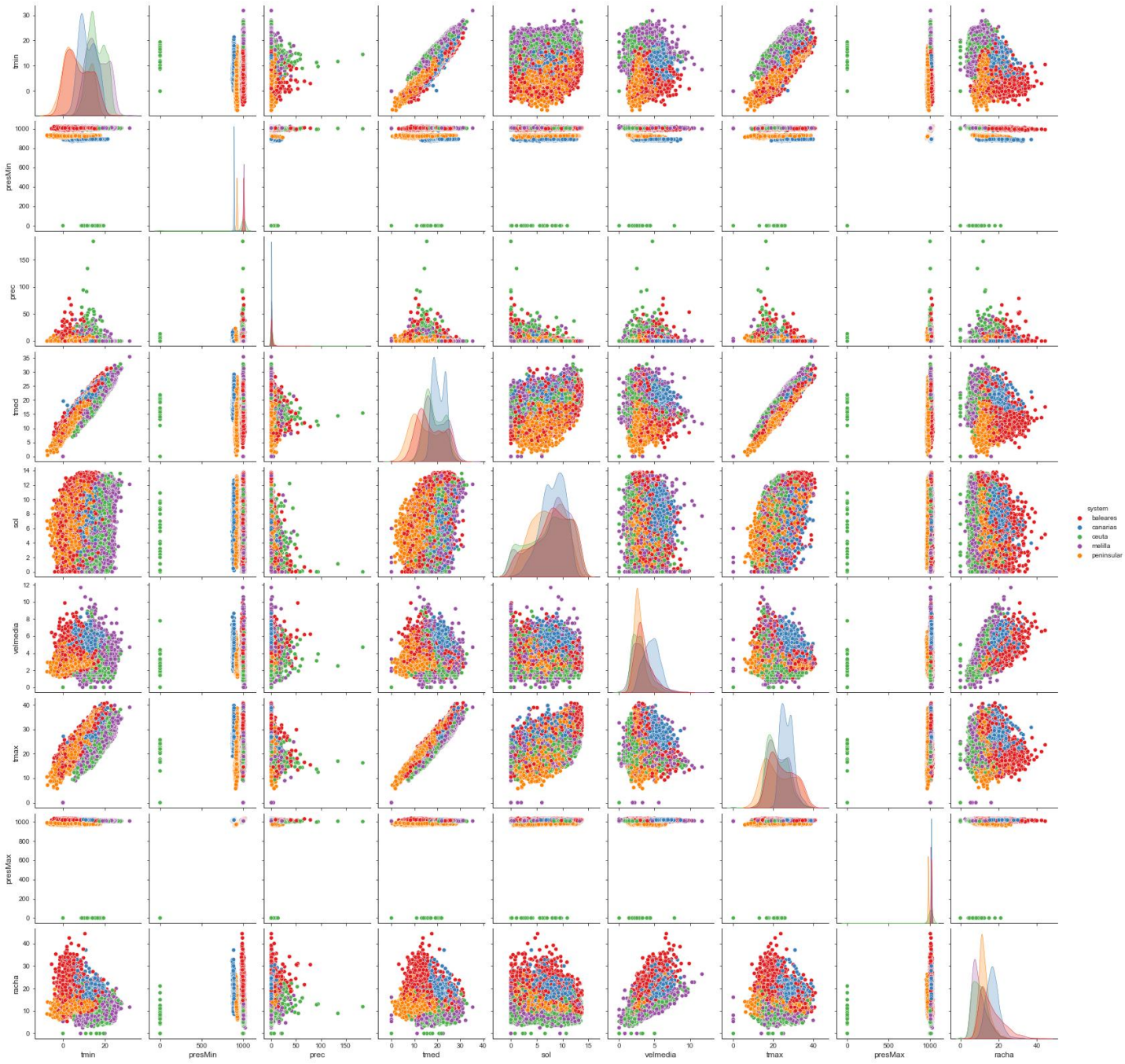
Análisis descriptivo

Antes de comentar con el preprocesado de los datos y la creación de los modelos de regresión, se ha realizado un análisis de las variables para comprobar su calidad, distribución, correlaciones y cualquier otra anomalía.

De este análisis se han sacado algunas conclusiones que han provocado que algunos datos se descarten e incluso el ámbito de desarrollo de los modelos se haya modificado.

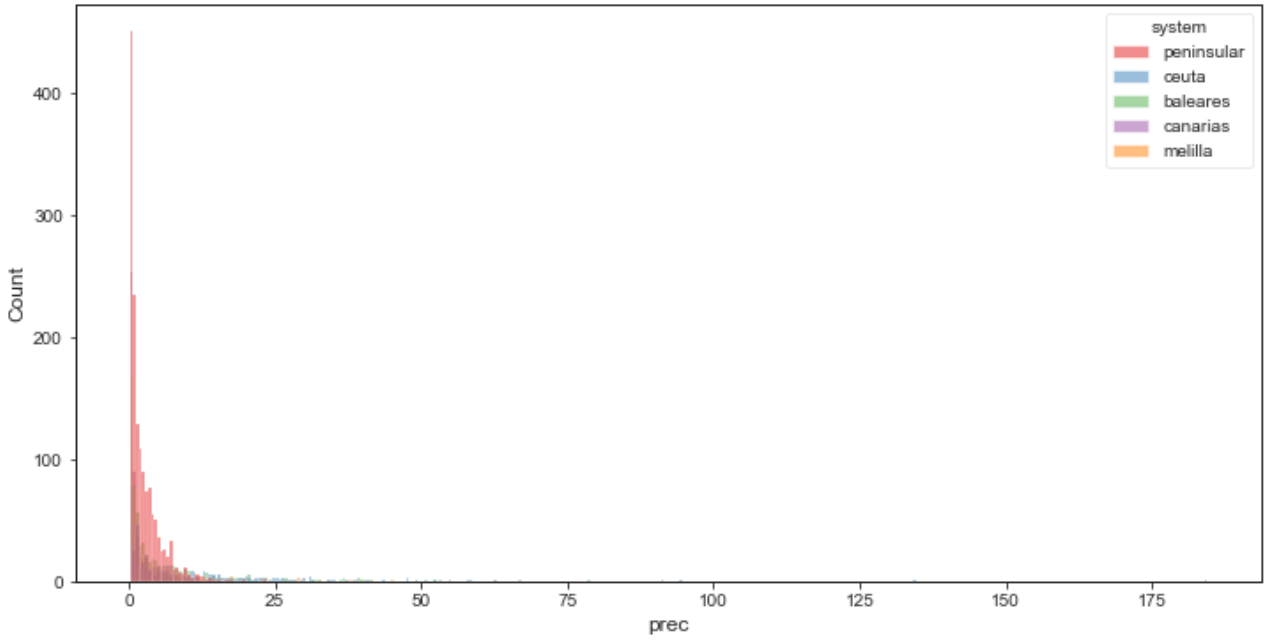
Como primer paso se realizó un gráfico pairplot para ver la relación entre todas nuestras variables, de este primer análisis podemos sacar ya algunas conclusiones, los datos para el sistema eléctrico de Ceuta tiene poca calidad muchos de sus registros no tienen información para las variables presmin y presmax, esto es debido a este sistema está compuesto por una sola estación meteorológica, por lo que, si falla la lectura de esos datos, provoca que no tengamos información.

En este primer análisis también se puede ver como existe cierta correlación entre algunas variables como tmax, tmin y tmed y racha y velmedia, estas correlaciones parecen lógicas al ser distintas mediciones sobre un mismo fenómeno. Por otro lado, vemos como hay algunos outliers para las variables prec y los campos de temperaturas, todos estos datos vistos en este primer análisis serán analizados con más profundidad.

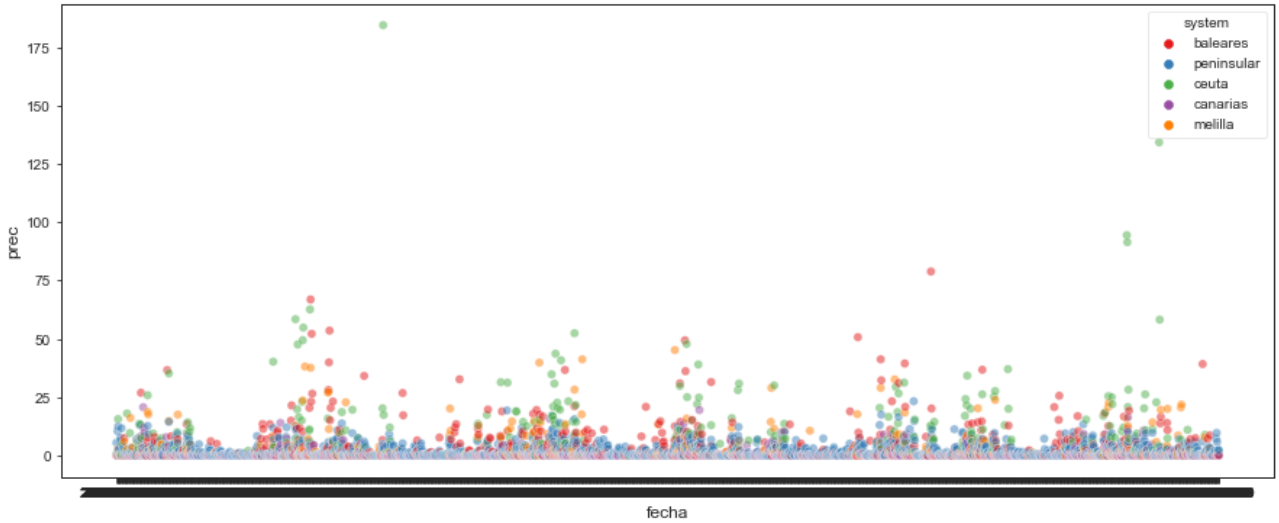


Precipitaciones

En el análisis de la variable prec, lo primero que se observa es que los datos se agrupan entorno al valor 0, como una distribución gamma con algunos valores extremos, teniendo una media de 1,33 y siendo su percentil 75% 0,466.

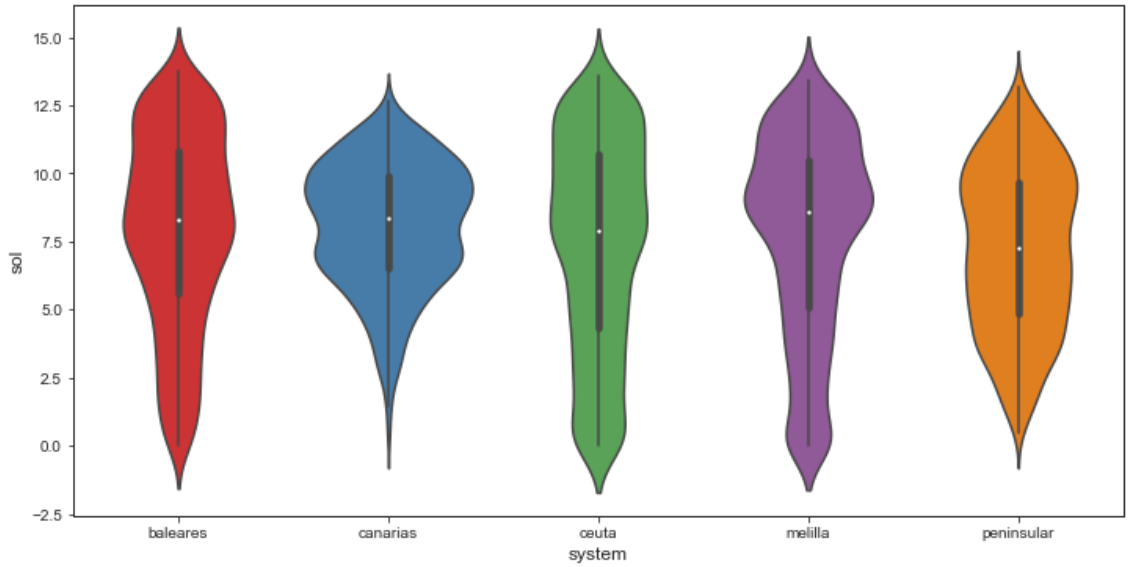


Para poder ver el detalle, se realiza un gráfico de puntos con los valores por fecha y sistema eléctrico podemos ver con más detalle como los valores extremos pertenecen a Ceuta, que como solo tiene una estación, no se realiza la media de las precipitaciones entre los distintos puntos de su territorio.



Horas de sol

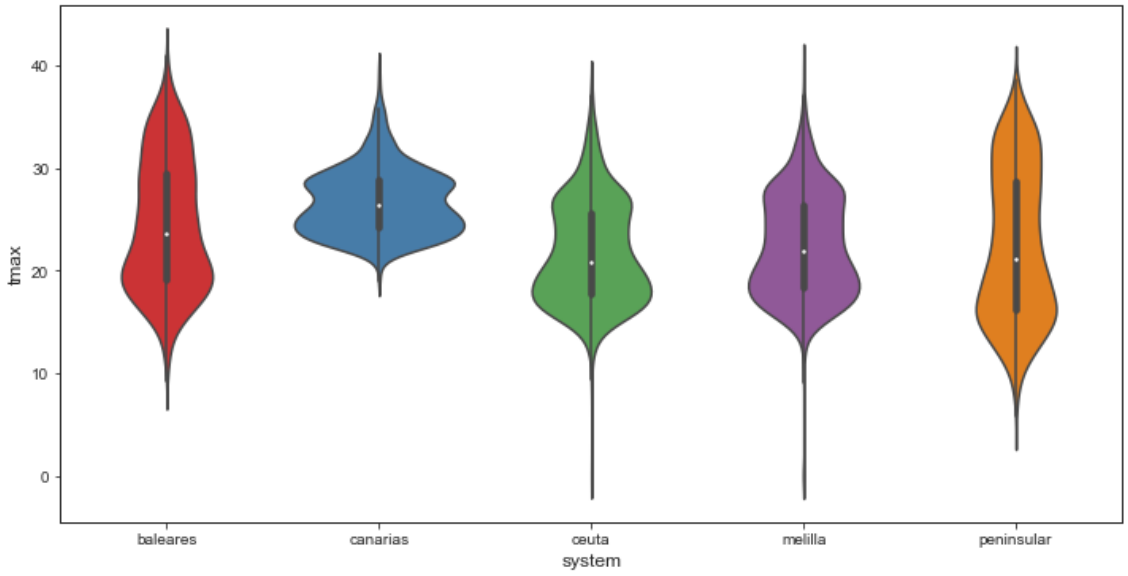
Como se puede ver en el siguiente gráfico las horas de sol tiene una distribución similar en todos los sistemas eléctricos tiendo algo distinta en canarias donde los datos se concentran más, además vemos que tiene una distribución muy uniforme, su media es de 7,6 con una desviación de 3,36 y un máximo de 13.78, además vemos que los datos tienen 2 modas que se corresponden con el verano e invierno.



Temperatura

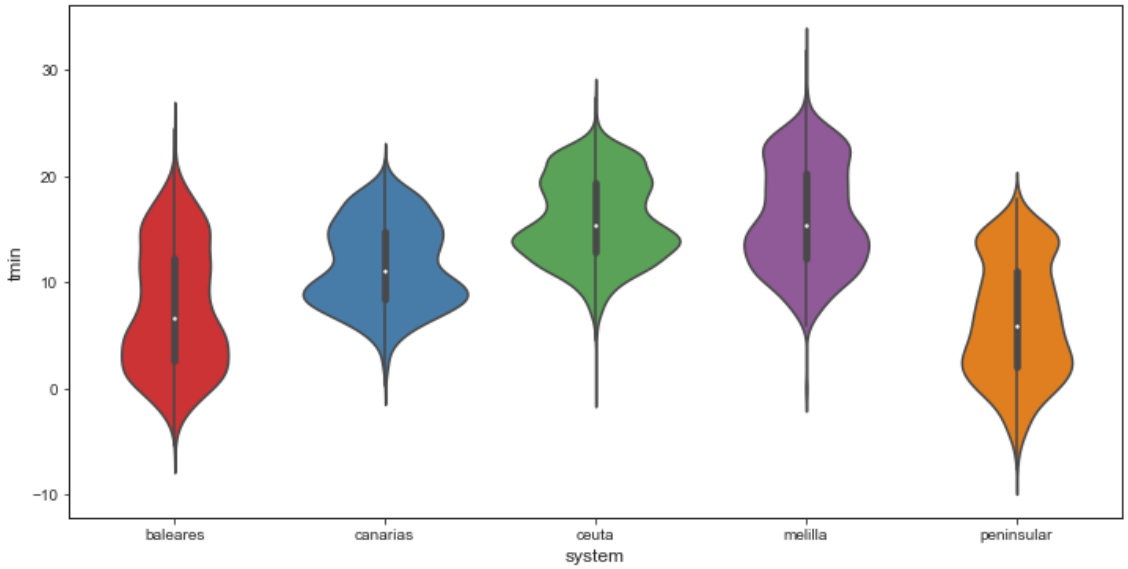
Temperatura máxima

Como en el caso de las horas de sol se ven 2 modas, teniendo distribuciones similares entre los distintos sistemas y concentrándose más el dato en canarias, donde hay menos variabilidad en el clima. Además, tampoco se observan grandes valores anómalos con un máximo de 40.9 grados, pero se ven valores a 0 para Ceuta y melilla que se corresponden estaciones sin lecturas.



Temperatura mínima

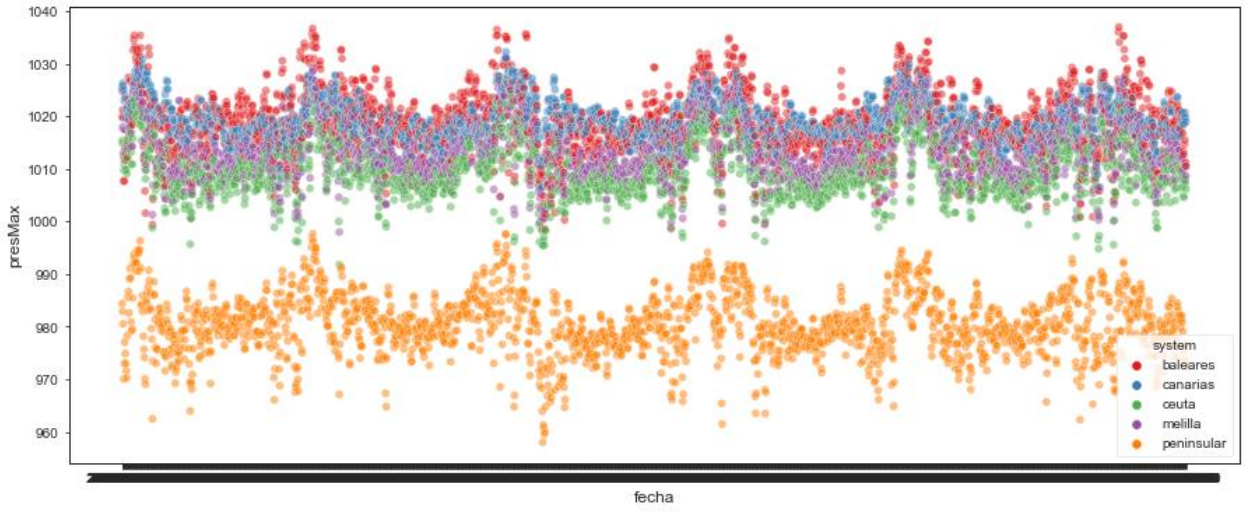
En cuanto a los valores de temperaturas mínimas tenemos una situación muy similar excepto que se ven outliers con valores muy elevados para melilla, siendo el máximo 31.8 grados y la media de 11.49. En el análisis del dato encontramos que existe un valor mínimo de -7.5 grados para el sistema eléctrico peninsular para el día 12/01/2021 coincidiendo con la borrasca filomena, además como en las variables anteriores los valores de canarias se encuentran más concentrados y las distribuciones son bimodales.



Presión

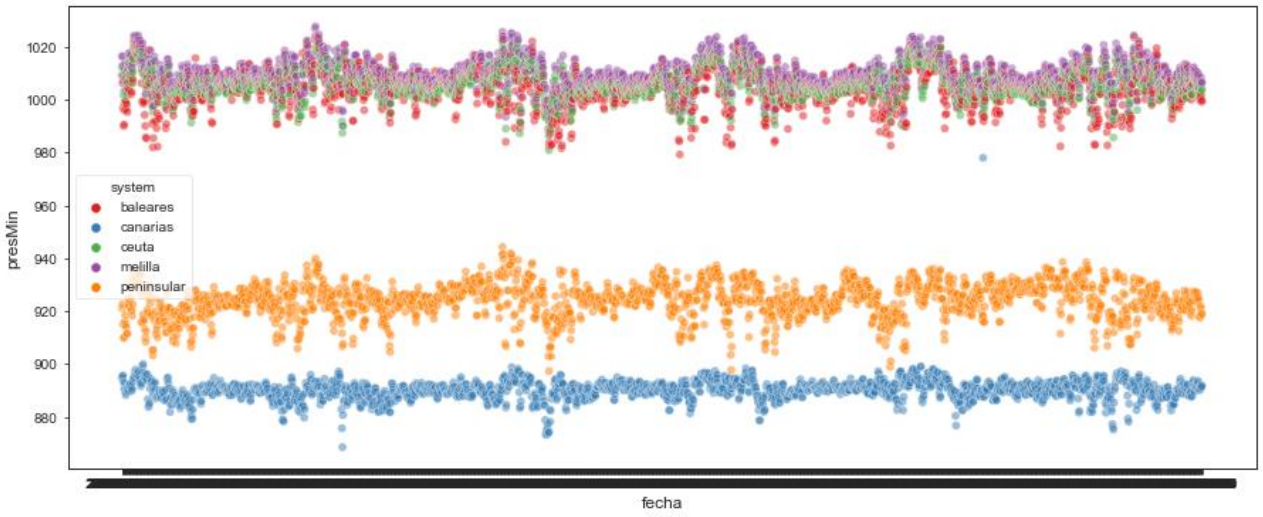
Presión máxima

El análisis de la presión máxima vemos como canarias y península tiene valores mucho más bajos que el resto de sistemas eléctricos además salvo los valores 0 que se han eliminado del gráfico la distribución entre los sistemas eléctricos parece similar. Los datos de la variable tienen una media de 1005.88 hPa con una desviación de 52,6 hPa y un valor máximo de 1037 hPa.



Presión mínima

El análisis de la presión mínima vemos datos similares a los de la presión máxima, con una media de 964.45 hPa y una desviación de 69.4 hPa, pudiendo distinguir como canarias y península tiene valores mucho más bajos que el resto de sistemas eléctricos además salvo los valores 0 que se han eliminado del gráfico la distribución entre los sistemas eléctricos parece similar.

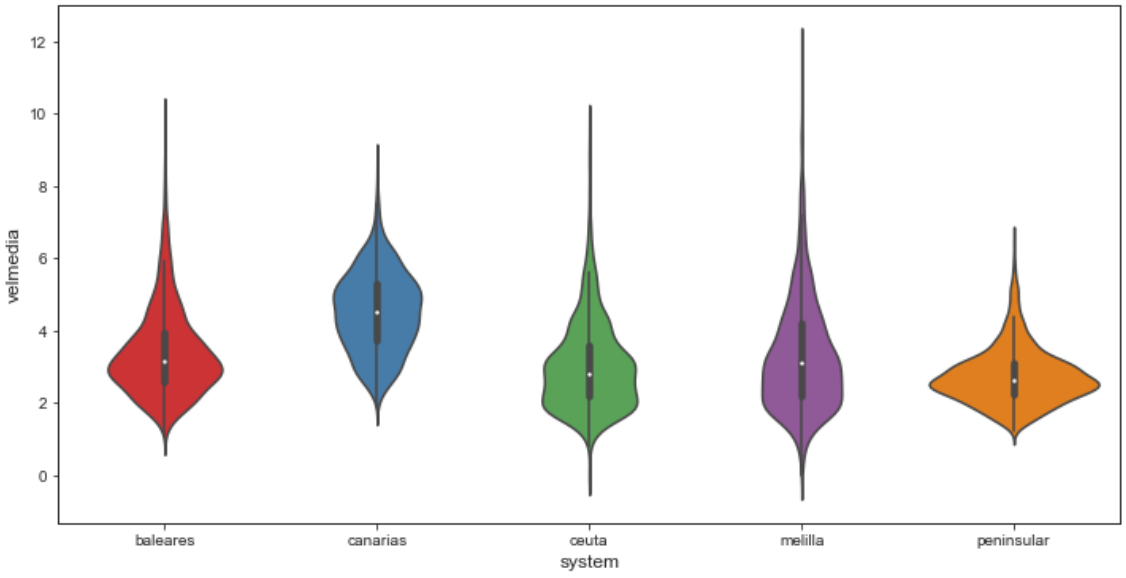


Viento

De igual modo que ocurre con la temperatura las distribuciones de velocidad media del viento y racha máxima son similares, ya que existe cierta correlación entre estos valores.

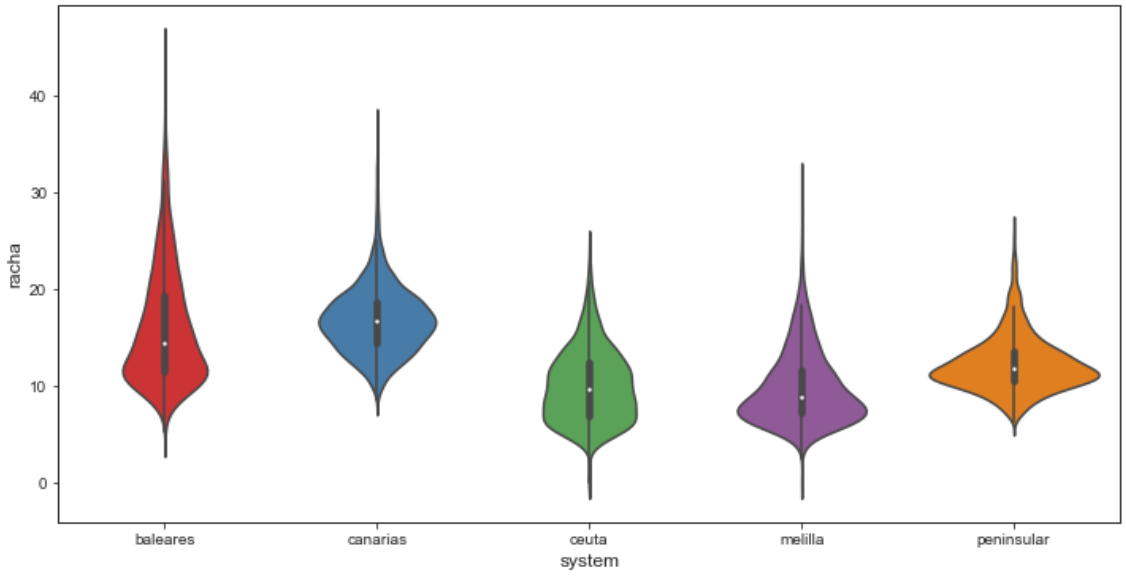
Velocidad media

En cuanto a la velocidad media podemos observar como los datos están bastante concentrados existiendo valores muy altos, de este modo la media de esta variable es de 3.39, el percentil 75 es de 4.2, pero el valor máximo se sitúa en 11.7.



Racha máxima

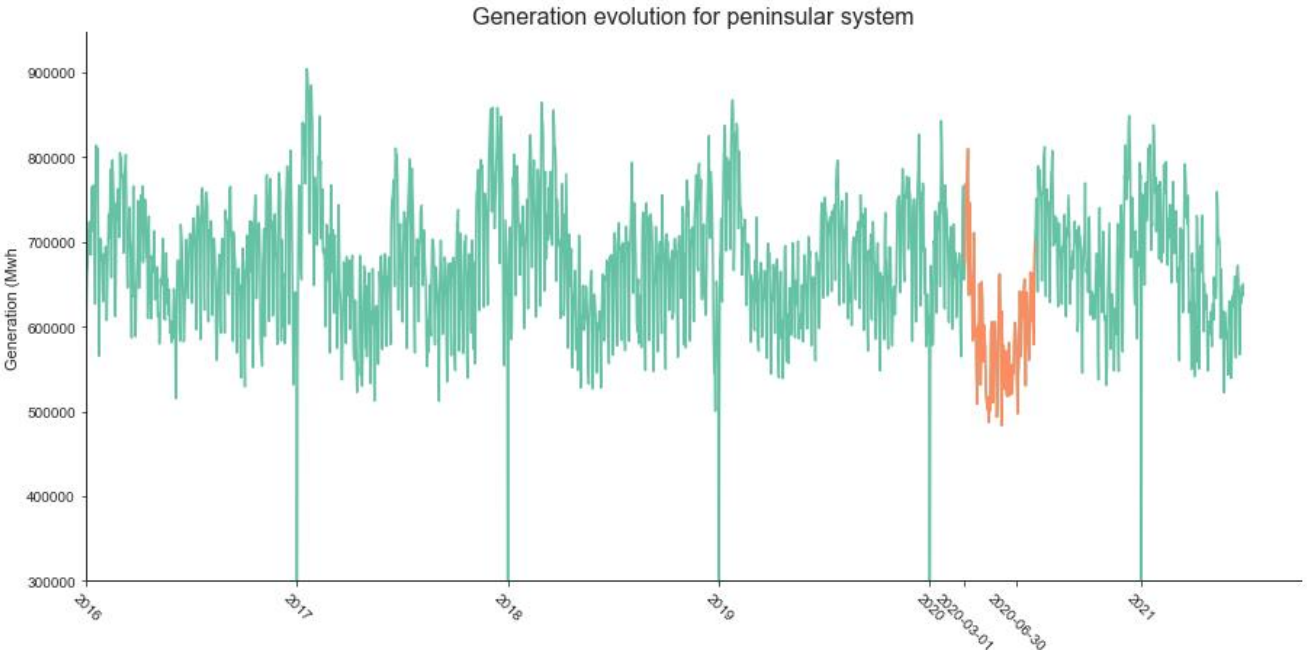
La distribución de las rachas máximas de viento es similar a la de la velocidad media, concentrándose los valores entorno a la media, con una desviación pequeña y con algunos valores máximos muy alejados, de hecho, la media se sitúa en 12.99, la desviación en 5.03, el percentil 75 en 16 y el valor máximo en 44.4

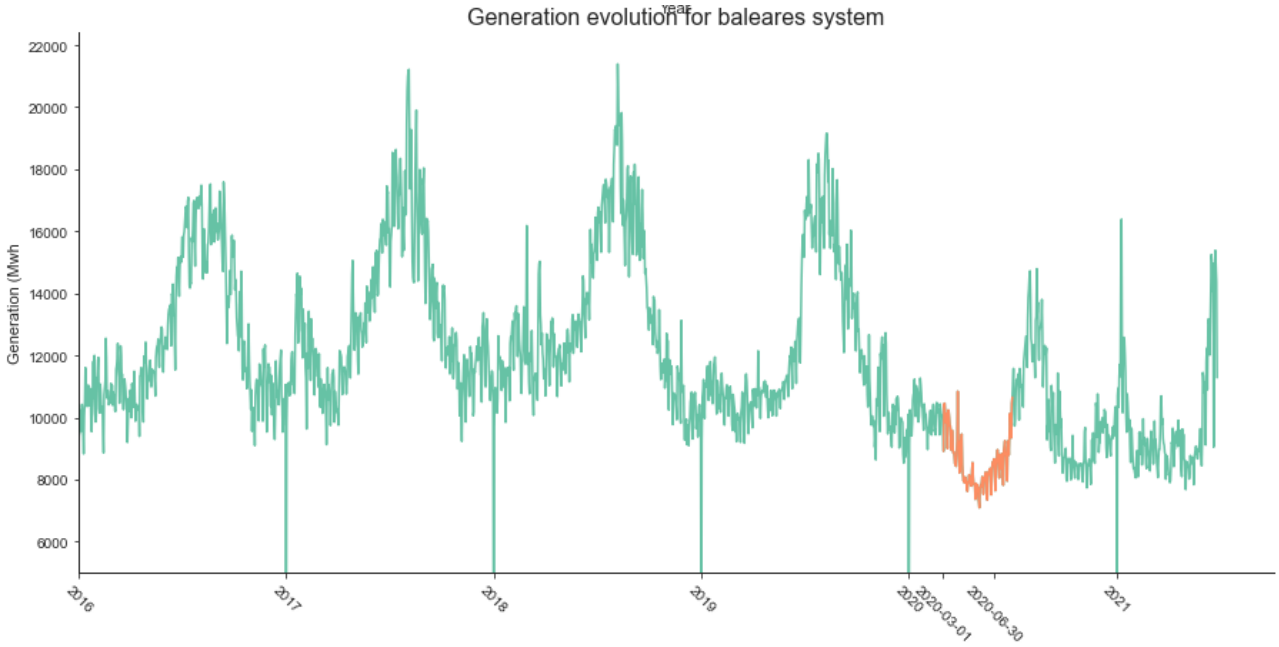


Calidad del dato

Valores de generación durante COVID-19

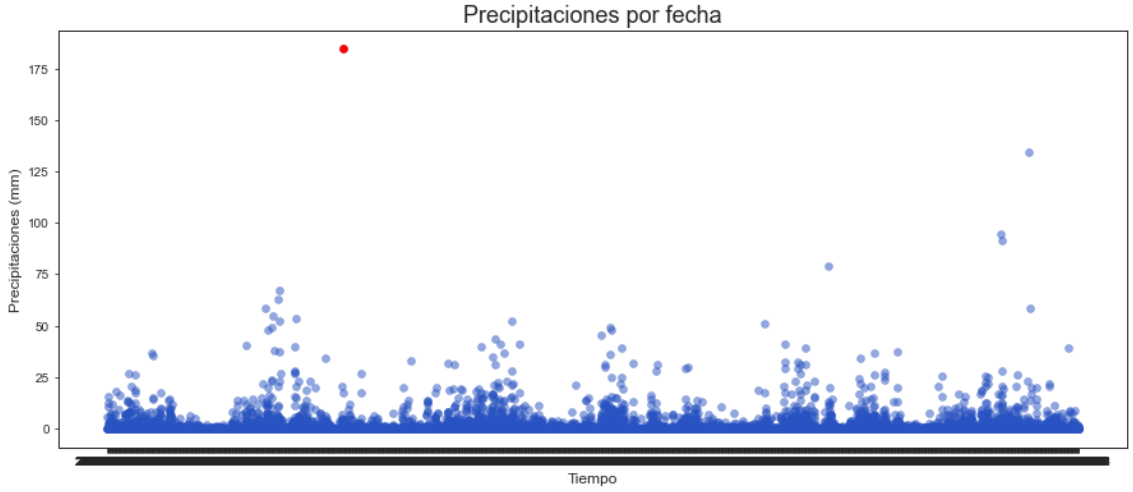
Analizando los valores para la generación eléctrica para los distintos sistemas eléctricos se observa como los datos tienen mucha estacionalidad, cambiando mucho según la época del año, además para los meses del confinamiento debido al virus del COVID-19 se ve como se ha producido un gran descenso de la generación que, aunque tiene cierta recuperación en los siguientes meses no llega a niveles de antes del confinamiento. Debido a esto tome la decisión de eliminar los datos del 15/03/2020 al 28/06/2020 que fue el tiempo que duro el confinamiento en la mayor parte de las CCAA. No se han eliminado datos posteriores para reflejar que la recuperación no ha sido completa, como se puede ver para península y Baleares:





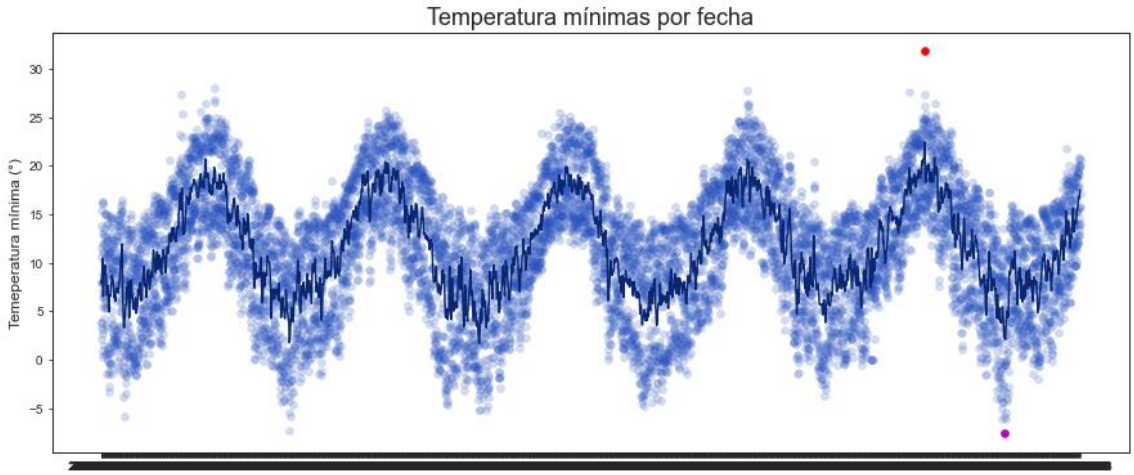
Outliers

Para comprobar la calidad del dato se han revisado los outliers y al ser información meteorología se ha contrastado contra el dato publicado en la web de AEMET, siendo los valores correctos. Para las precipitaciones hay un máximo de 184.4mm para Ceuta para el 28/04/2017. De hecho este fenómeno provoco inundaciones en la ciudad (https://elpueblodeceuta.es/art/18750/inundaciones-en-varias-zonas-de-la-ciudad-por-las-fuertes-y-persistentes-lluvias)



	fecha	system	Holiday	weekday	tmin	presMin	prec	tmed	sol	velmedia	tmax	presMax	racha
2017-04-28	ceuta	0	4	14.5	998.7	184.4	15.4	0.0	4.7	16.3	1002.7	11.9

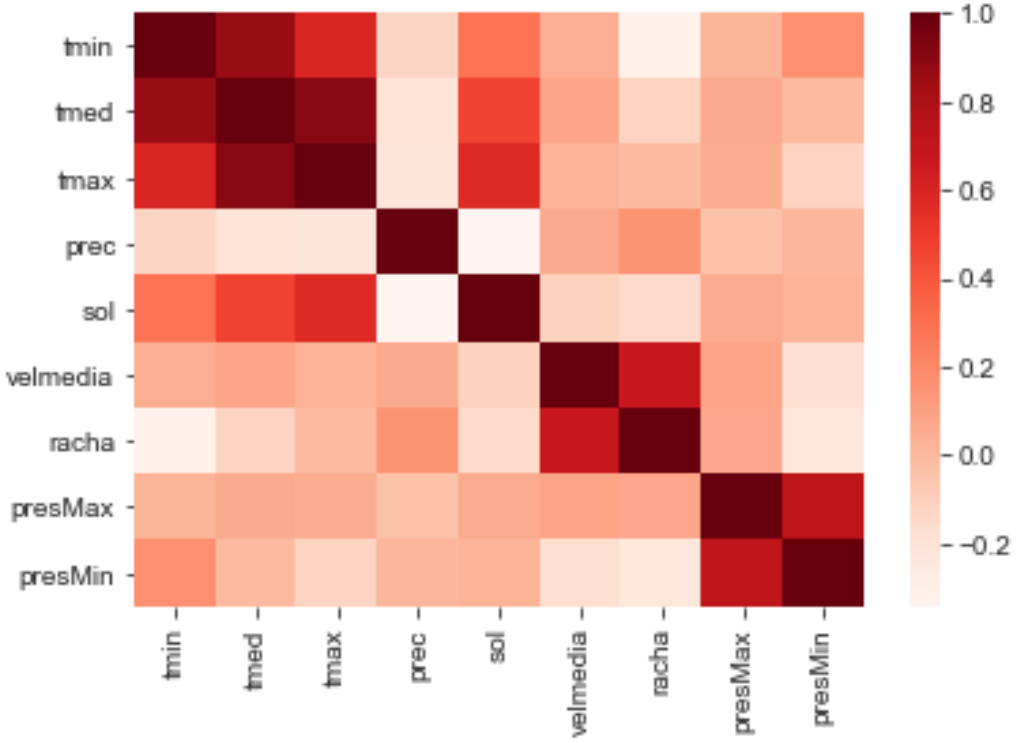
También es notable la temperatura mínima existiendo un día para el cual la temperatura mínima fue de 31.8 grados en Melilla el 02/08/2020.



fecha	system	Holiday	weekday	tmin	presMin	prec	tmed	sol	velmedia	tmax	presMax	racha
2020-08-02	melilla	0	6	31.8	1005.7	0.0	35.4	12.1	4.7	39.1	1008.3	11.7

Matriz de correlación

Con el fin de analizar las correlaciones entre las variables con más detalle, se ha revisado la correlación entre las variables meteorológicas y se han representado en una matriz.



	tmin	tmed	tmax	prec	sol	velmedia	racha	presMax	presMin
tmin	1.000000	0.861894	0.599205	-0.121157	0.288536	0.048719	-0.313343	0.023976	0.167520
tmed	0.861894	1.000000	0.908930	-0.190148	0.475156	0.085610	-0.119845	0.063569	-0.002920
tmax	0.599205	0.908930	1.000000	-0.212737	0.579667	0.026509	-0.000746	0.052660	-0.112294
prec	-0.121157	-0.190148	-0.212737	1.000000	-0.341196	0.063641	0.159817	-0.035045	0.018802
sol	0.288536	0.475156	0.579667	-0.341196	1.000000	-0.106090	-0.145863	0.057532	0.027393
velmedia	0.048719	0.085610	0.026509	0.063641	-0.106090	1.000000	0.684285	0.092813	-0.165299
racha	-0.313343	-0.119845	-0.000746	0.159817	-0.145863	0.684285	1.000000	0.074465	-0.224448
presMax	0.023976	0.063569	0.052660	-0.035045	0.057532	0.092813	0.074465	1.000000	0.726678
presMin	0.167520	-0.002920	-0.112294	0.018802	0.027393	-0.165299	-0.224448	0.726678	1.000000

Con estos datos podemos ver que existe una fuerte correlación entre los datos de tmed y tmin, tmed y tmax, a la vista de estos resultados intentaremos evitar el uso de las variables tmed por si pudiese empeorar el resultado de las regresiones.

Además de las correlaciones mencionadas anteriormente, existen fuertes correlaciones entre PresMax y PresMin, que se va a obviar por que las dos variables nos dan el rango en el que se mueven la presión para un día, lo cual aporta información al modelo. Y la correlación entre velocidad media y racha de máxima que, aunque existe correlación del 0.684, no me parece suficiente como para tomar ninguna decisión al respecto.

Preprocesamiento de campos objetivo

Antes de comenzar con la gestión de variables de los modelos, se va realizar el tratamiento de las variables del target del modelo. Además, como la información que queremos obtener es relativa a la generación de energías renovables vamos a eliminar la información del sistema eléctrico de Ceuta, puesto que sólo contiene tecnologías de generación no renovables.

Creación de nuevas variables

Para poder unir la información de generación electica con la información meteorológica se crean las variables día, mes y año a partir del campo fecha, además se eliminan los registros que hacen referencia a la generación total.

Categorización target

Con el fin de tener una variable para cada tipo de tecnología de generación además de variables para el total de generación renovable y el total de generación no renovable. Realizo un Onehotencoder de las variables 'Tecnologia' y 'Renov_norenov' y posteriormente lo vuelvo a unir a los valores de fecha, sistema y generación (MwH), de esta manera en cada registro tengo la generación para un día, sistema y tecnología, marcando con un ‘1’ la tecnología de generación.

Generacion_Mwh	fecha	system	x0_Carbón	x0_Ciclo combinado	x0_Cogeneración	x0_Eólica	x0_Fuel + Gas	x0_Hidroeólica	x0_Hidráulica	x0_Motores diésel	x0_Nuclear
29281.000	2016-01-01	peninsular	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

Una vez tenemos el dato en este formato mediante una función previamente definida de la librería utils, agrupo el dato por fecha y sistema, informando cada variable de tecnología y renovable o no renovable con el % de generación correspondiente

system	fecha	Generacion_Mwh	x0_Carbón	x0_Ciclo combinado	x0_Cogeneración	x0_Eólica	x0_Fuel + Gas	x0_Hidroeólica	x0_Hidráulica	x0_Motores diésel	x0_Nuclear	re
baleares	2016-01-01	8814.678	0.343993	0.258372	0.007846	0.001254	0.0	0.0	0.0	0.115979	0.0	

De esta manera ya tenemos los datos en el formato deseado para utilizarlos como target de las regresiones, por ultimo unimos esta información por fecha y sistema con los datos meteorológicos para tener nuestra tabla preparada para los distintos modelos de datos.

Featuring

Codificación de variables

Una de las variables de entrada de los modelos de regresión debe de ser el sistema eléctrico, esta variable puede tener los valores:

- Melilla
- Baleares
- Canarias
- Peninsular

Para poder utilizar esta variable como entrada del modelo y puesto que no tiene muchas categorías distintas se utiliza OnehotEncoder para generar una variable binaria para cada uno de los posibles valores.

Tratamiento de fechas

Al utilizar fechas y no plantear los modelos como series temporales, tenemos que conseguir dar valores continuos a los valores de fechas, para ello lo realizamos mediante el uso de cosenos.

Para pasar las fechas aun formato en que el salto entre el ultimo día de un mes y el primer día del siguiente sea continuo, uso los cosenos de los días y meses.

Para ello situó los valores de los 31 días en ángulos iguales calculándolos como:

$$Dia(x) = \cos\frac{2\pi x}{31}$$

Para los meses situo cada mes en:

$$Mes(x) = \cos\frac{2\pi x}{12}$$

Y para los días de la semana:

$$Weekday(x) = \cos\frac{2\pi x}{7}$$

Además, la variable del año tiene un valor excesivamente grande comparado con el resto de valores, pero he preferido no añadirla al escalado para no dar más peso a los años con más valores, puesto que para 2020 hemos eliminando casi 3 meses por el confinamiento COVID-19 y realizando un escalado normal también restaríamos importancia al año en curso. Por lo que finalmente se realiza:

$$year(x) = \log(x)$$

Train-test Split

Para realizar la separación entre train y test se ha creado una función en la librería utils, que separa los datos en función de la fecha y las variables que se le pasen como features y target. Se ha utilizado un 85% de los datos para entrenar al modelo y un 15% para realizar el test del modelo.

De este modo evitamos utilizar datos futuros para realizar el entrenamiento del modelo. Para hacerla más reutilizable he añadido la posibilidad de obtener un set de datos de validación.

Escalado

Para evitar que ciertos valores tengan mucho peso dentro del modelo y que los valores tengan un peso relativo sobre su variable se realiza un escalado de las variables que se van a utilizar como entrada del modelo: tmin, presMin, prec, sol, tmax, presMax, racha, velmedia y tmed.

Esta última variable, tmed, tiene una fuerte correlación con tmin y tmax y su uso en los modelos finalmente de regresión se ha eliminado por empeorar el modelo.

No se ha realizado ningún método de reducción de dimensionalidad puesto que el número de variables es relativamente pequeño (16)

Elección de los modelos

En una primera aproximación, se optó por crear una red neuronal con 17 valores de salida (1 por cada tecnología) pero los valores de salida, eran pésimos, seguramente el problema sea que no hay valores suficientes para entrenar la red, ya que disponemos de datos de menos de 2 años para cada sistema.

Visto los resultados obtenido con la red neuronal se optó por un modelo de ML tradicional, como métricas para evaluar los modelos sea optado por usar el error cuadrático medio (RMSE), error absoluto medio (MAE) y R2. Además, con R2 podemos tener una idea de si el modelo tiene overfitting.

Como modelo naif se ha optado por usar un modelo de regresión lineal el cual utilizaremos para comprobar que los modelos mejoran este escenario base.

Como ya se ha detallado anteriormente se van a generar 3 modelos distintos:

- Modelo de generación total por sistema eléctrico.
- Modelo de predicción del porcentaje de generación renovable.
- Modelo de predicción del porcentaje de generación solar fotovoltaica y eólica.

Modelo de generación total por sistema eléctrico.

Este modelo tiene como objetivo obtener el dato de la generación total dada una fecha, sistema electro y las variables meteorológicas.

Como features tenemos los siguientes campos:

- Holiday
- Weekday
- Tmin
- presMin
- prec
- sol
- tmax
- presMax
- racha
- velmedia
- year
- day
- month
- x0_baleares
- x0_canarias
- x0_melilla
- x0_peninsular

Y como target tenemos la **Generacion_Mwh**, además en este conjunto de datos se añaden las variables que definen el sistema eléctrico para posteriormente poder obtener los resultados por sistema eléctrico.

Para empezar, se entrenaron los modelos ajustando los parámetros de entrada intentando obtener las mejores meticas posibles para ello se ha utilizado la función de sklearn GridSeachCV, como se van a utilizar varias métricas se ha utilizado la opción refit para entrenar de nuevo el mejor modelo usando la métrica RMSE.

Primero se empezaron utilizando modelos sencillos como KNeighborsRegressor y DecisionTreeRegressor para los que ya se obtuvieron valores mejores que con la aproximación realizada con Deep Learning.

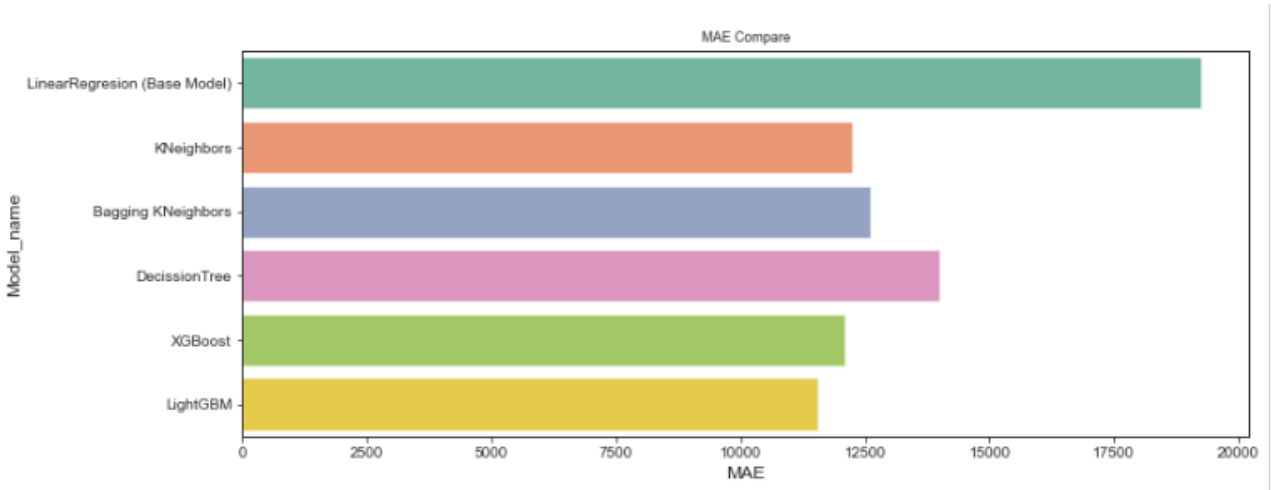
Posteriormente se optó por utilizar modelos completos con boosting, los modelos de este tipo utilizados han sido XGBoost y LigthGBM, los dos se basan en arboles de decisión y he obtenido valores muy similares con ellos.

Por último, en vista del buen resultado obtenido con el algoritmo KNeighbors, se probó a realizar un algoritmo de bagging sobre este regresor, para ello utilice la función de sklearn BaggingRegressor.

Evaluación

Para evaluar los modelos se midieron las métricas MAE, RMSE y R2 contra el conjunto de datos de test y posteriormente se visualizaron las predicciones de cada uno de los modelos utilizando este mismo conjunto de datos para comprobar si existía overfitting

La comparativa de resultado de MAE es la siguiente:

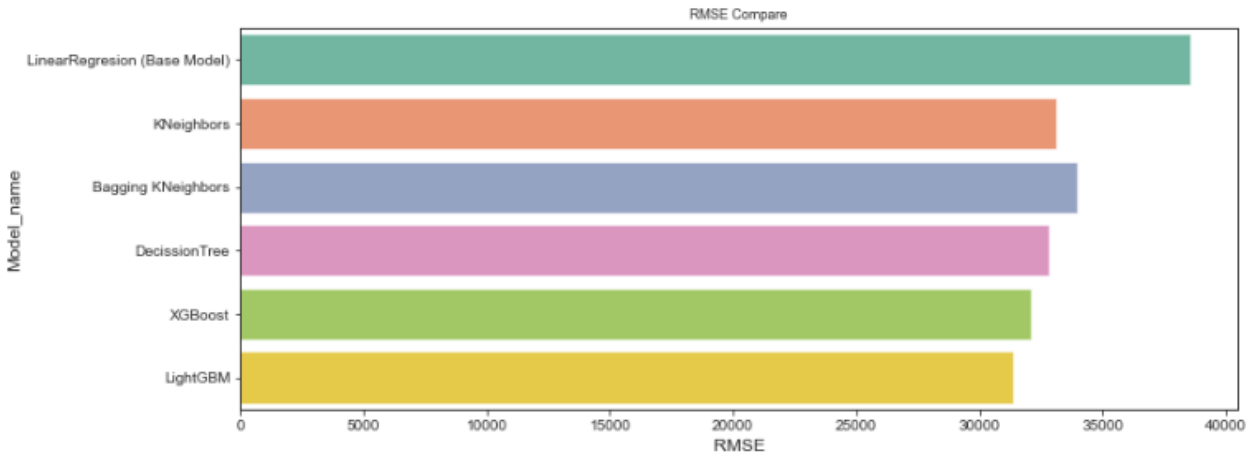


Como se puede observar midiendo el error absoluto medio, todos los modelos son mejores que el modelo naif, la regresión lineal.

Sorprendentemente el modelo K-neighbors es casi tan bueno cono el modelo de boosting XGBoost y el mejor que modelo basado en bagging de k-neighbors, lo cual me sorprendió bastante.

Como conclusión podemos ver que usando como métrica MAE el mejor modelo es el basado en LightGBM.

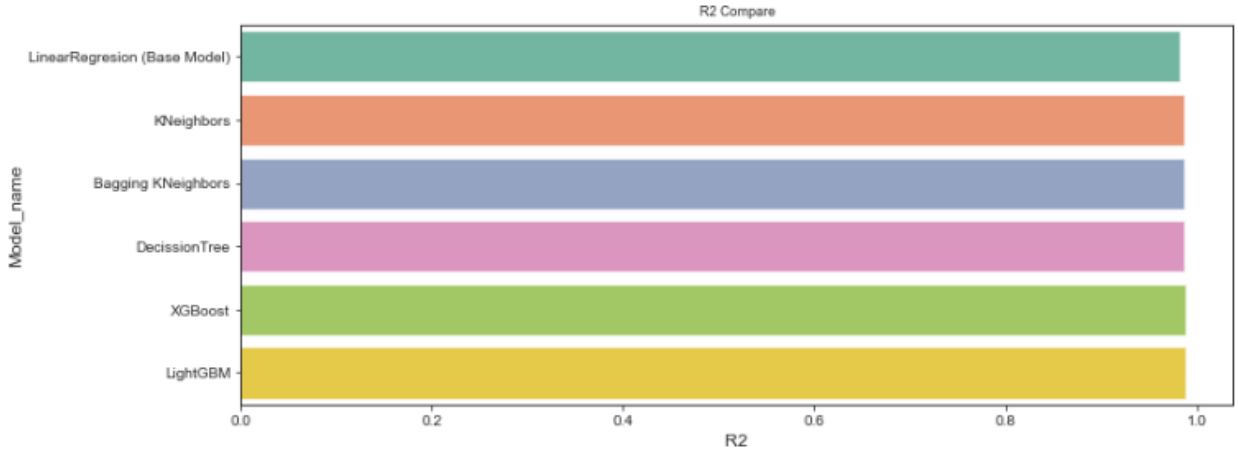
La comparativa usando RMSE difiere en los resultados, siendo estos:



Se puede ver como en utilizando como métrica RMSE el peor modelo después de la regresión lineal es el modelo de k-neighbors con bagging, mientras que en este caso DecisionTreeRegressor es mejor que los modelos basados en k-neighbors

Del mismo modo que con MAE, el mejor modelo es el basado en LightGBM.

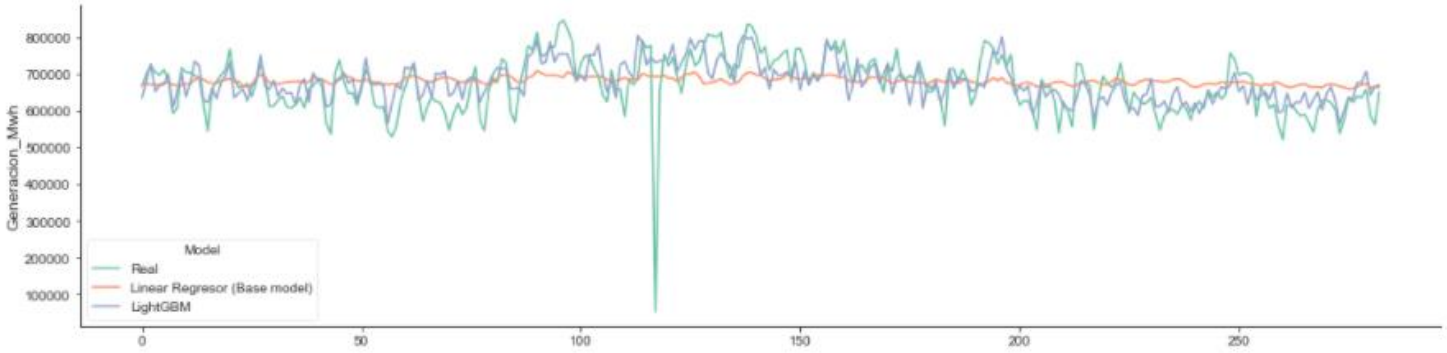
Por último, se comparó el resultado de la métrica R2:



Como se puede ver los valores son muy cercanos a 1, lo cual, aunque parece indicar que los modelos son buenos, también es un síntoma de posible overfitting.

Además, el resultado para casi todos los modelos es muy similar, para comprobar si existe overfittig se visualizaron los datos de la predicción de cada uno de los modelos sobre el conjunto de test contra los datos reales.

Como se puede ver para el mejor modelo obtenido, el basado en LightGBM, con los datos del sistema peninsular generaliza bien, pero sin llegar a existir overfitting:



Como último paso se creó un pipeline incluyendo todos los pasos del feature engineering y los mejores parámetros del modelo basado en LigthGBM, cuyos parámetros fueron:

- Max_depth=5
- N_estimators=129
- Learning_rate=0.09
- Subsample=0.3
- colsample_bytree=0.9

Esta pipeline se entrenó con todo el conjunto de datos para posteriormente ser exportado y usado en la aplicación realizada en streamlit.

Por último, además de realizar la prueba visual para comprobar el overfitting, se comparó el best_score del modelo utilizando los datos de entrenamiento y utilizando todos los datos, comprobando que el resultado con todo el conjunto de datos era peor, confirmando que no existe overfitting.

- Resultado usando datos de train: - -28673.3275
- Resultado usando todos los datos: -29720.6864

Modelo de predicción del porcentaje de generación renovable.

Con este modelo se busca obtener el porcentaje de energía generada que tendrá como tecnología de generación una renovable.

Para ello partimos de las siguientes variables como fe atures:

- Olida
- Wendy
- Tomín
- pres Min
- pre
- sol
- temas
- prisma
- racha
- demedia
- yesar
- Day
- monto
- x0_baleares
- x0_canarias
- x0_melilla
- x0_peninsular

Y como tenemos **x1_Renovable**, que es el porcentaje de energía generada con tecnologías renovables. Del mismo modo que para el modelo de generación total como modelo a batir se optó por una regresión lineal.

La metodología llevada a cabo fue la misma que para el modelo anterior, pero en este caso viendo los resultados del caos anterior no se realizó el desarrollo del modelo mediante bagging.

Por lo tanto, se utilizaron los siguientes algoritmos:

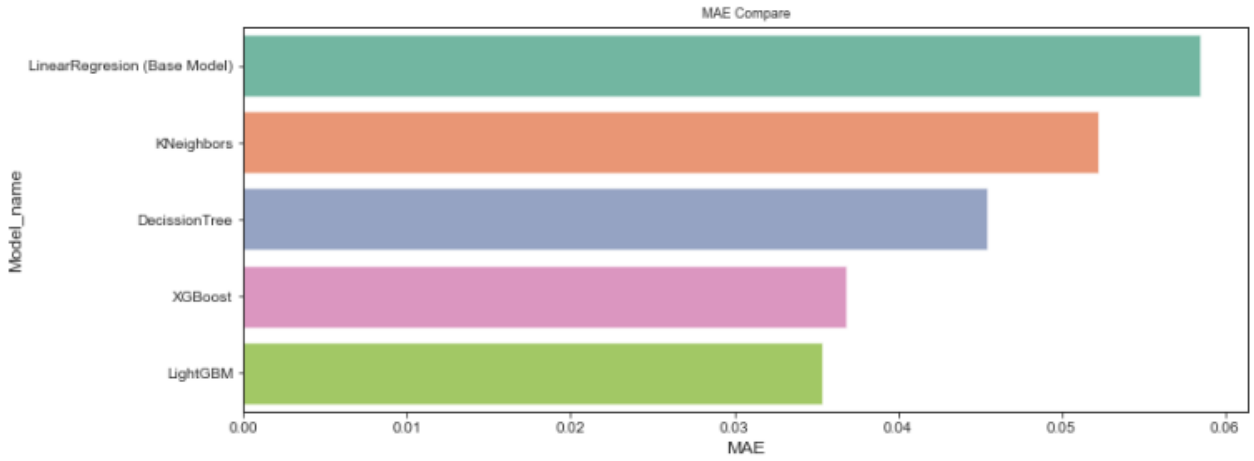
- KNeighborsRegressor
- DecissionTreeRegressor

- XGBoostRegressor
- LightGBMRegressor

Evaluación

Para evaluar los modelos se midieron las métricas MAE, RMSE y R2 contra el conjunto de datos de test y posteriormente se visualizaron las predicciones de cada uno de los modelos utilizando este mismo conjunto de datos para comprobar si existía overfitting

La comparativa de resultado de MAE es la siguiente:

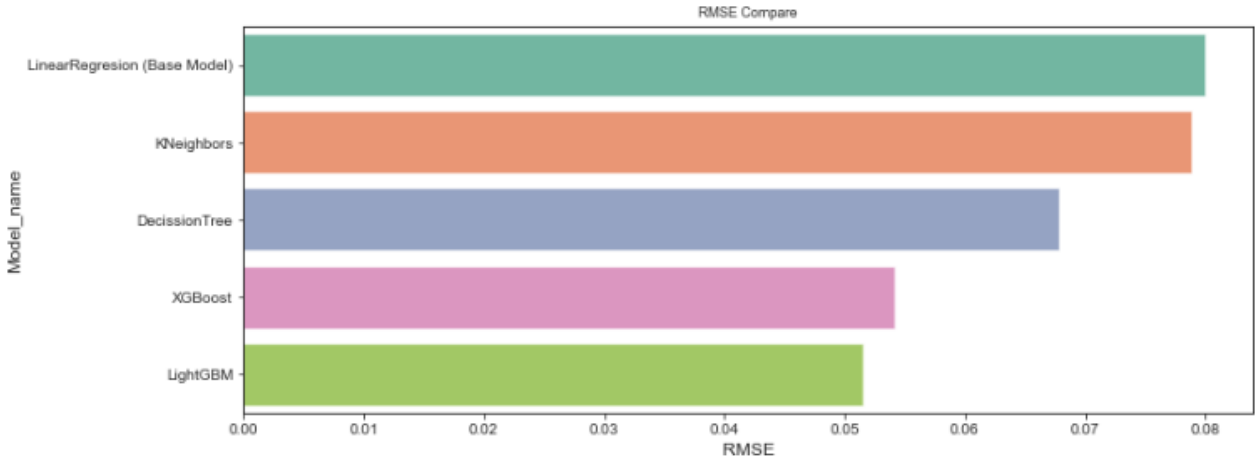


Como se puede observar midiendo el error absoluto medio, todos los modelos son mejores que el modelo naif, la regresión lineal.

Al contrario que en el modelo anterior, el modelo que peor se ajusta al resultado deseado es el kneighbors, siendo los modelos de boosting bastante mejores que el resto.

Como conclusión podemos ver que usando como métrica MAE el mejor modelo es el basado en LightGBM.

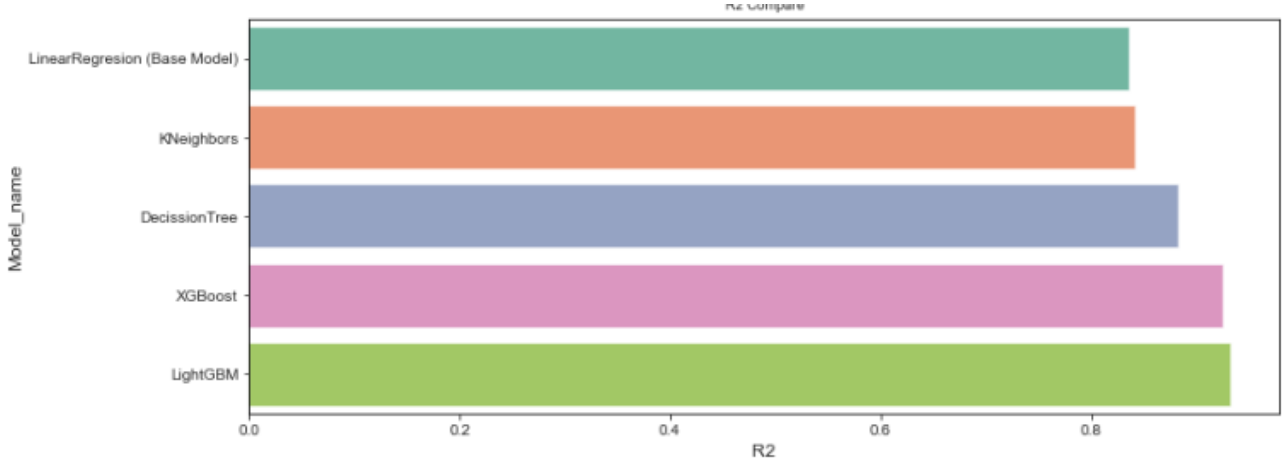
La comparativa usando RMSE difiere en los resultados, siendo estos:



Se puede ver como utilizando como métrica RMSE el peor modelo después de la regresión lineal es el modelo de k-neighbors, que obtiene una métrica poco mejor que este.

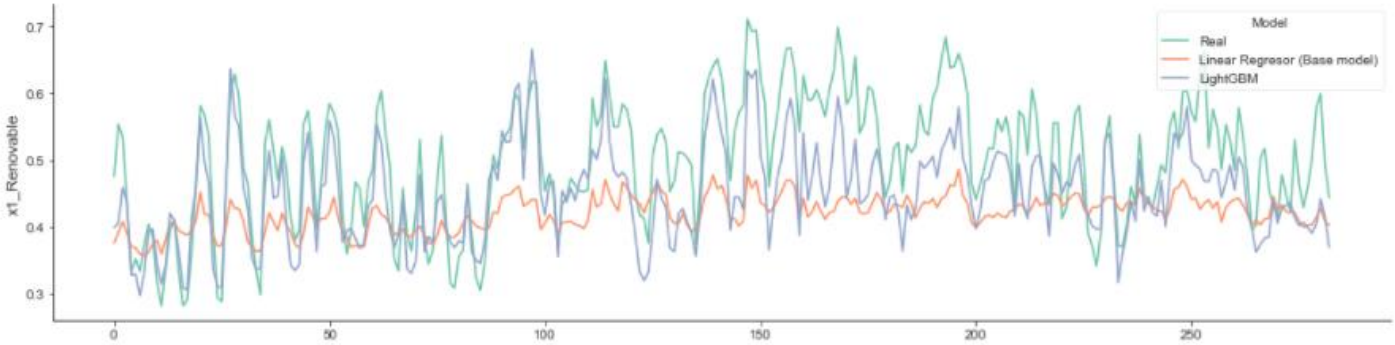
Además, de manera similar a los resultados obtenidos con MAE, los modelos de boosting son superiores al resto, siendo el mejor modelo el LightGBM.

Por último, se comparó el resultado de la métrica R2:



Como se puede ver todos los valores son superiores al 0.8 y los valores de los modelos de XGBoost y LightGBM son cercanos a 1, esto indica que los modelos predicen bastante bien, pero puede haber un problema de overfitting, para comprobar si existe se visualizaron los datos de la predicción de cada uno de los modelos sobre el conjunto de test contra los datos reales.

Realizando la comparativa contra el sistema peninsular y usando el mejor modelo obtenido (LightGBM), podemos ver como no existe overfitting:



Como último paso se creó un pipeline incluyendo todos los pasos del feature engineering y los mejores parámetros del modelo basado en LigthGBM, cuyos parámetros fueron:

- Max_depth=5
- N_estimators=99
- Learning_rate=0.08
- Subsample=0.5
- colsample_bytree=0.9

Esta pipeline se entrenó con todo el conjunto de datos para posteriormente ser exportado y usado en la aplicación realizada en streamlit.

Por último, además de realizar la prueba visual para comprobar el overfitting, se comparó el best_score del modelo utilizando los datos de entrenamiento y utilizando todos los datos, comprobando que el resultado con todo el conjunto de datos era peor, confirmando que no existe overfitting.

- Resultado usando datos de train: -0.04217
- Resultado usando todos los datos: -0.04504

Modelo de predicción del porcentaje de generación solar fotovoltaica y eólica.

Con este modelo se busca obtener el porcentaje de energía generada que tendrá como tecnología de generación Solar fotovoltaica y Eólica, todos los modelos deberán ser multioutput y como salida tendrán un array de 2 posiciones.

Para ello partimos de las siguientes variables como features:

- Holiday
- Weekday
- Tmin
- presMin
- prec
- sol
- tmax
- presMax
- racha
- velmedia
- year
- day
- month
- x0_baleares
- x0_canarias
- x0_melilla
- x0_peninsular

Y como target obtenemos **x0_Solar fotovoltaica y x0_Eólica**, que son los porcentajes de energía generada con tecnología Solar fotovoltaica y Eólica. Del mismo modo que para el modelo de generación total como modelo a batir se optó por una regresión lineal, el gran cambio para estos modelos es que tiene 2 valores de salida para llevar a cabo esto se ha optado por utilizar el paquete de sklearn MultiOutputRegressor, el cual nos da un algoritmo para convertir los modelos de regresión en multi salida.

La metodología llevada a cabo fue la misma que para en el modelo de porcentaje de generación renovable, pero invocando a los distintos tipos de regresores dentro de la llamada a MultiOutputRegressor().

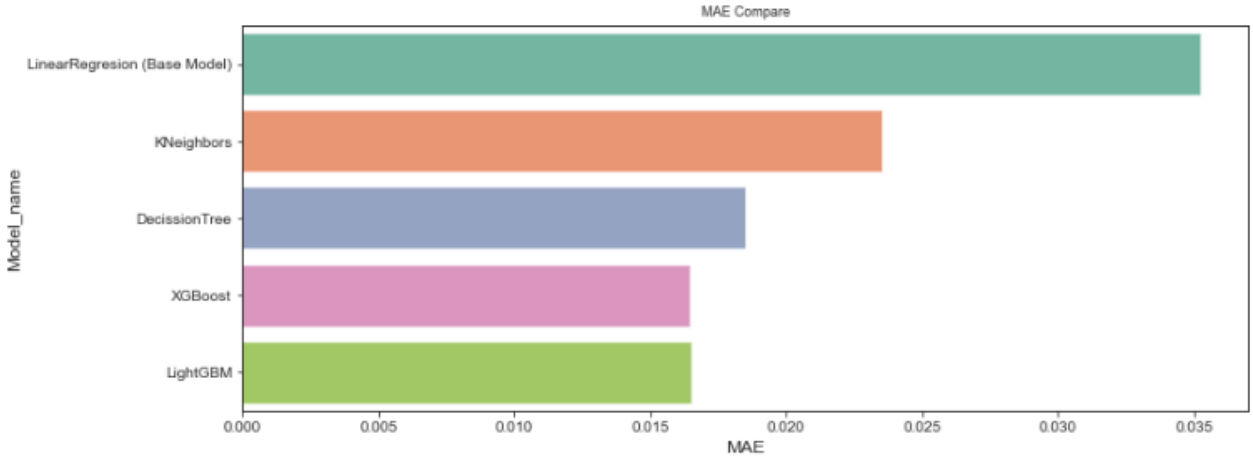
Por lo tanto, se utilizaron los siguientes algoritmos:

- KNeighborsRegressor
- DecissionTreeRegressor
- XGBoostRegressor
- LightGBMRegressor

Evaluación

Para evaluar los modelos se midieron las métricas MAE, RMSE y R2 contra el conjunto de datos de test y posteriormente se visualizaron las predicciones de cada uno de los modelos utilizando este mismo conjunto de datos para comprobar si existía overfitting

La comparativa de resultado de MAE es la siguiente:

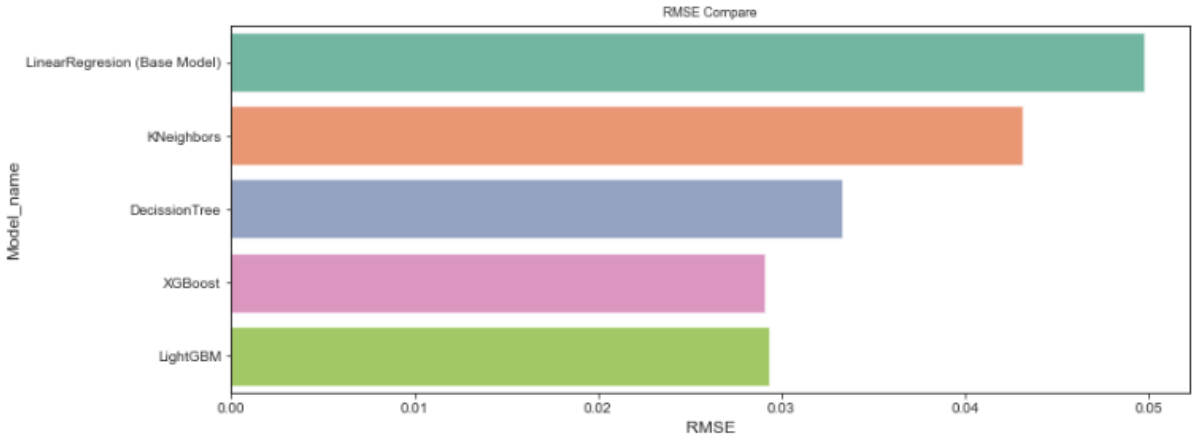


Como se puede observar midiendo el error absoluto medio, todos los modelos son mejores que el modelo naif, la regresión lineal.

Del mismo modo que para el modelo de porcentaje de generación renovable, el modelo que peor se ajusta al resultado deseado es el k-neighbors, siendo los modelos de boosting bastante mejores que el resto.

Como conclusión podemos ver que usando como métrica MAE el mejor modelo es el basado en XGBoost, que es ligeramente superior al LightGBM.

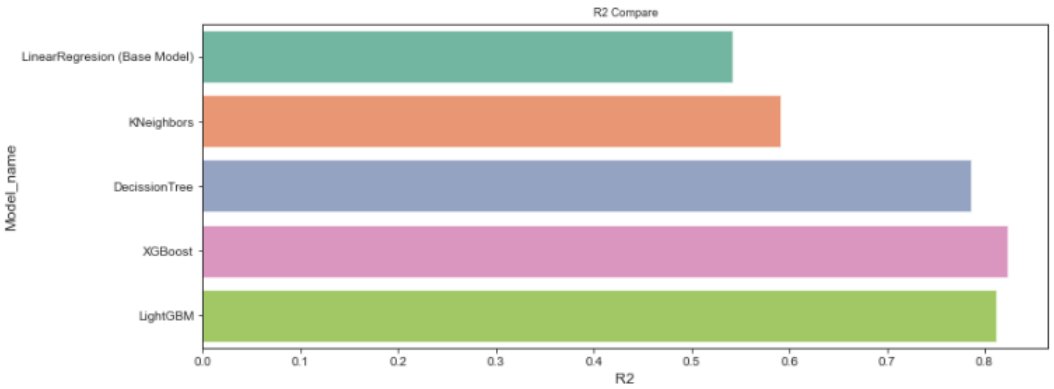
La comparativa usando RMSE difiere en los resultados, siendo estos:



Se puede ver como utilizando como métrica RMSE el peor modelo después de la regresión lineal es el modelo de k-neighbors junto con DecissionTree.

Además, de manera similar a los resultados obtenidos con MAE, los modelos de boosting son superiores al resto, siendo el mejor modelo el XGBoost.

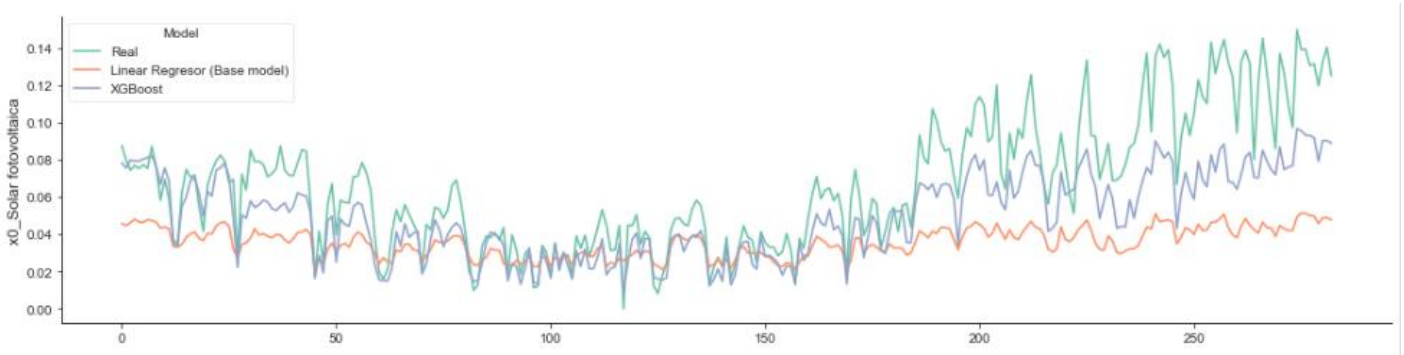
Por último, se comparó el resultado de la métrica R2:

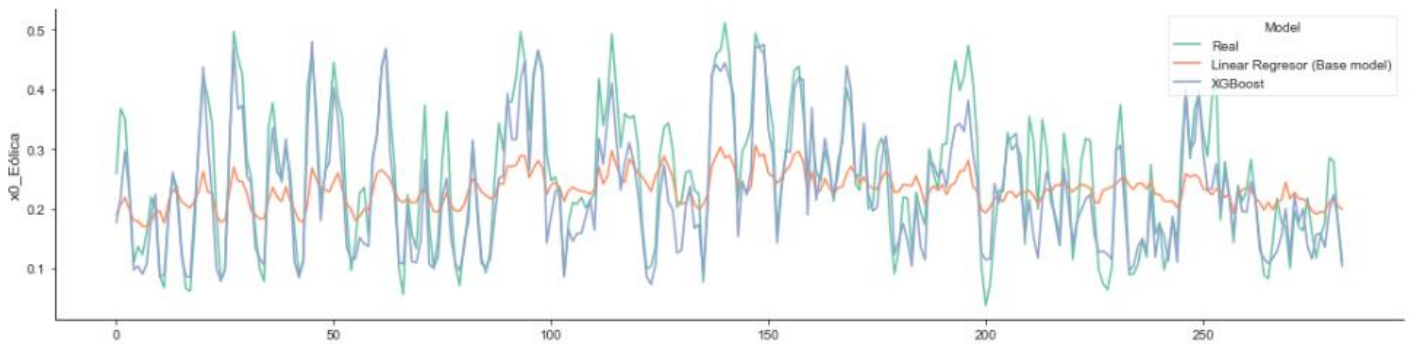


Con esta métrica a diferencia de los modelos anteriores podemos ver que solo los modelos XGBoost y LightGBM, son superiores al 0.8, este resultado, aunque no es excesivamente alto indica que estos 2 modelos predicen bastante bien,

Para comprobar si existe overfitting se visualizan los datos de la predicción de cada uno de los modelos sobre el conjunto de test contra los datos reales.

Realizando la comparativa contra el sistema peninsular para cada una de las salidas del modelo y usando el mejor modelo obtenido (XGBoost), podemos ver como no existe overfitting:





Como último paso se creó un pipeline incluyendo todos los pasos del feature engineering y los mejores parámetros del modelo basado en XGBoost, cuyos parámetros fueron:

- Max_depth=5
- N_estimators=59
- Learning_rate=0.11
- Subsample=0.7
- colsample_bytree=0.9

Esta pipeline se entrenó con todo el conjunto de datos para posteriormente ser exportado y usado en la aplicación realizada en streamlit.

Por último, además de realizar la prueba visual para comprobar el overfitting, se comparó el best_score del modelo utilizando los datos de entrenamiento y utilizando todos los datos, comprobando que el resultado con todo el conjunto de datos era peor, confirmando que no existe overfitting.

- Resultado usando datos de train: -0.01643
- Resultado usando todos los datos: -0.01921

Conclusiones

Manual de usuario

Con el fin de crear una interfaz para mostrar los resultados de los distintos modelos desarrollados, se ha creado una aplicación Python usando streamlit.

Esta aplicación despliega una web, desde la que se puede obtener el resultado de las predicciones en función de unos parámetros de entrada meteorológicos configurables por el usuario.

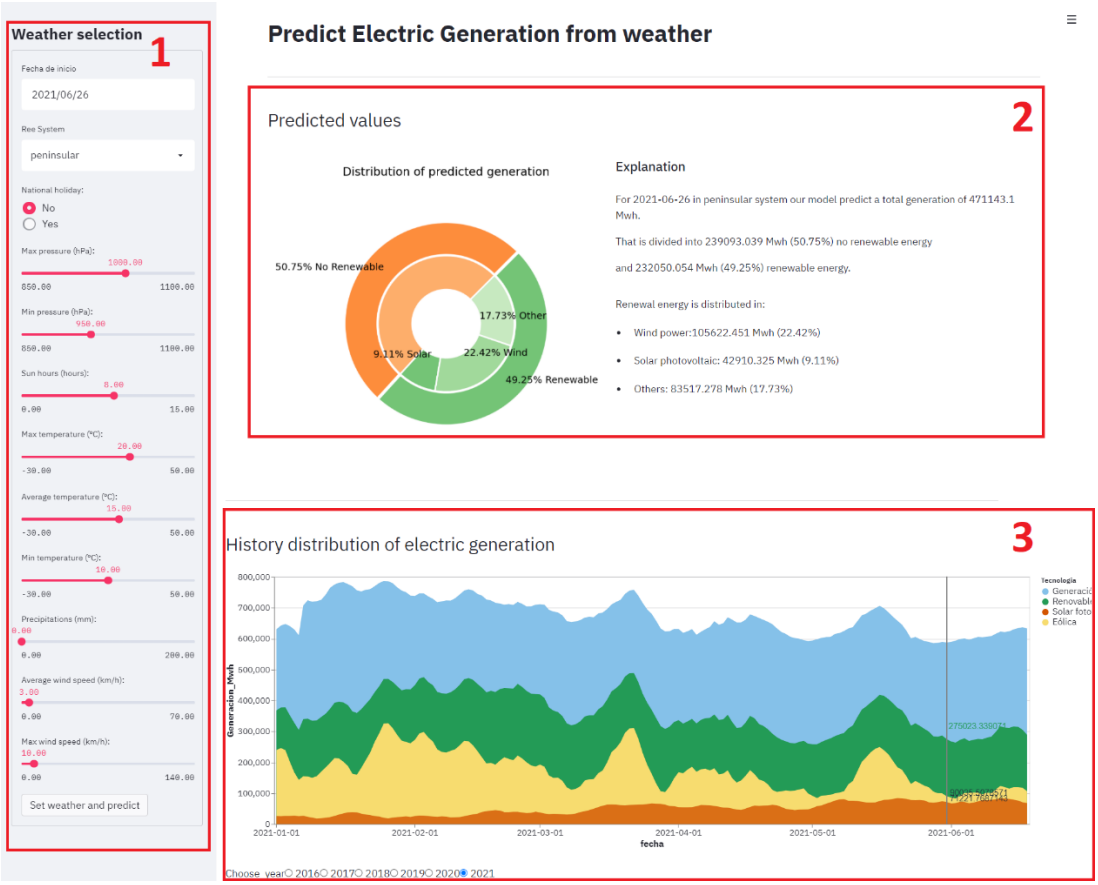
La ejecución de esta aplicación se puede realizar de dos maneras:

- Desde la línea de comando desde la ruta '/Python' del proyecto, ejecutando el siguiente comando: *streamlit run app.py*
- Si el repositorio se ha desplegado en Google Drive, se puede ejecutar mediante el siguiente notebook de /Python, donde se generará una URL pública desde la que se puede acceder a la aplicación: *6-Execute_Streamlit.ipynb*

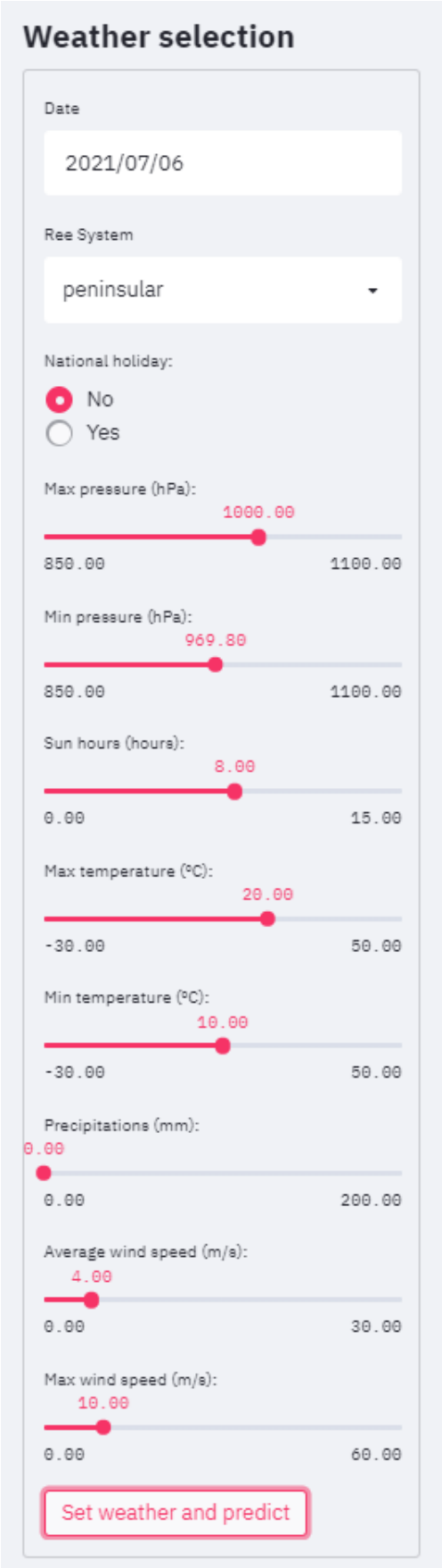
Descripción General

La aplicación se compone de 3 partes principales que se detallan a continuación

1- Menú lateral: Selectores para introducir los datos meteorológicos del usuario.
2- Resultado del modelo: Gráfico y explicación de los resultados obtenidos
3- Gráfico con el histórico: Grafico con el histórico de generación eléctrica por año y tecnología.



Menú Lateral



Con este menú se pueden seleccionar los datos de entrada para obtener los valores predichos por los modelos. Los valores a informar son:

Date: Fecha de los datos de entrada debe de ser mayor o igual a la fecha de ejecución de la aplicación

Ree System: Selector de sistema eléctrico para el que se desea realizar la predicción

National Holiday: Indicador sobre si la fecha es festivo nacional o no lo es.

Max pressuser (hPa): Máxima presión atmosférica en hPa

Min pressuser (hPa): Mínima presión atmosférica en hPa, la máxima presión no puede ser menor que la mínima presión

Sun hours: Horas de sol durante el día

Max temperatura (ºC): Temperatura máxima del día

Min temperatura (ºC): Temperatura mínima del día, no puede ser mayor que la temperatura máxima.

Precipitacions (mm): Precipitaciones acumuladas durante el día

Average wind speed (m/s): Velocidad media del viento

Max wind speed (m/s): Máxima velocidad del viento, no puede ser mayor que la velocidad media.

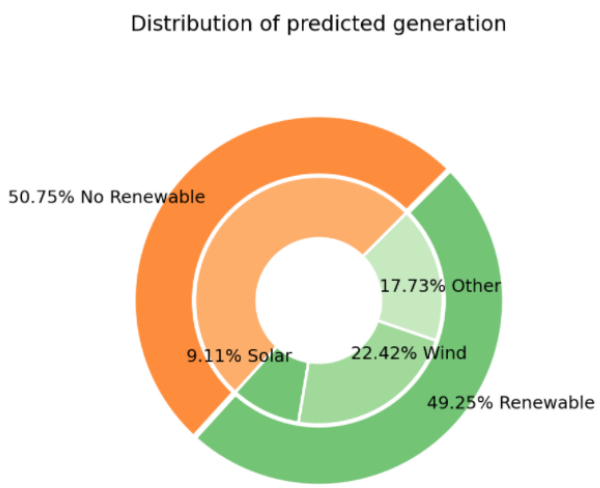
Resultado del Modelo

En esta parte de la web se muestra el resultado con los valores seleccionados en el menú lateral, en la parte izquierda se muestra un gráfico con la distribución de energía No Renovable y Renovable, y dentro de esta tipología de generación, se desglosa el porcentaje de energía de Solar fotovoltaica, el de energía eólica y el del resto de energías renovables.

En la parte derecha se detalla la explicación de los resultados predichos por el modelo incluyendo los valores de la generación en Mwh y los porcentajes de cada valor.

Predict Electric Generation from weather

Predicted values



Explanation

For 2021-06-26 in peninsular system our model predict a total generation of 471143.1 Mwh.

That is divided into 239093.039 Mwh (50.75%) no renewable energy

and 232050.054 Mwh (49.25%) renewable energy.

Renewal energy is distributed in:

- Wind power:105622.451 Mwh (22.42%)
- Solar photovoltaic: 42910.325 Mwh (9.11%)
- Others: 83517.278 Mwh (17.73%)

Gráfico Histórico

Además de los resultados de los modelos, en la aplicación se incluye un gráfico interactivo creado con Altair, que muestra el valor de la generación en Mwh de cada tipología de energía.

Este gráfico representa la información de cada año, mediante un selector se puede elegir el año a mostrar y mediante otro selector el sistema eléctrico que desea visualizarse.

Además, se ha incluido una funcionalidad adicional, al pasar el ratón sobre el gráfico muestra los valores de generación de Renovable, Solar y Eólica para esa fecha.