



KSCHOOL – MASTER EN DATA SCIENCE

TFM

Plataforma de predicción de la producción eléctrica de una instalación fotovoltaica

Prediction platform for the electricity production of a photovoltaic installation

Author: Alejandro Ruiz Berciano

Madrid, July 2021

Content

IMAGE INDEX	4
1. INTRODUCTION	5
2. OBJECTIVE	6
3. REQUIREMENTS AND PREPARATION.....	7
3.1. REQUIREMENTS	7
3.2. EXECUTION INSTRUCTIONS AND FOLDER STRUCTURE	7
4. METHODOLOGY	9
4.1. OPERATING SCHEME	10
4.2. DATA OBTENTION	11
4.2.1. Obtaining data from AEMET stations	11
4.2.2. Historical data obtention	12
4.3. DATA PREPARATION FOR MODEL TRAINING	16
4.3.1. Data cleansing	16
4.3.2. Data preparation	22
4.4. PREDICTION MODEL TRAINING	26
4.4.1. Data standardization	26
4.4.2. PCA	26
4.4.3. Model training	27
4.5. USER INTERFACE	35
4.5.1. Necessary functions to obtain electricity production	35
4.5.2. Obtaining electricity production.....	37
4.5.3. Streamlit interface	39
5. CONCLUSIONS	47

Image Index

Image 1: Folder structure of the TFM folder	7
Image 2: Folder structure of the data folder	8
Image 3: MDP Methodology.....	9
Image 4: TFM operating diagram	10
Image 5: Generation of the rows with the days in columns for the historical weather data	20
Image 6: Generation of rows with days in columns for historical weather forecast data	21
Image 7: Correlation matrix of the different input variables to the prediction models.....	24
Image 8: Cumulative explained variance versus number of components used.....	27
Image 9: Correlation matrix of the different input components to the prediction models.....	27
Image 10: Average values of solar radiation per hour	29
Image 11: Predicted vs. Actual Values of Different Tested Machine Learning Algorithms	31
Image 12: MAE of the different tested machine learning algorithms.....	32
Image 13: R^2 of the different machine learning algorithms tested.....	32
Image 14: Predicted vs. Actual Values of Different Tested Machine Learning Algorithms	33
Image 15: MAE of the different tested machine learning algorithms.....	34
Image 16: R^2 of the different machine learning algorithms tested.....	34
Image 17: Regular operation of the TFM system	37
Image 18: Hourly electricity consumption curves of the different standard profiles.....	38
Image 19: Main screen of the Streamlit App	41
Image 20: Orientation selection in Streamlit.....	41
Image 21: Inclination selection in Streamlit.....	42
Image 22: Location selection in Streamlit.....	42
Image 23: Entering peak power in Streamlit.....	43
Image 24: Selection of consumption habits in Streamlit.....	44
Image 25: Prediction of electricity production in Streamlit	44
Image 26: Estimated profile of photovoltaic consumption and production for the next day in Streamlit	45
Image 27: Estimated compensation of surpluses (€) for the next day in Streamlit.....	46
Image 28: Downloading data by the user in Streamlit	46
Table 1: Structure of the AEMET meteorological stations dataset.....	12
Table 2: Structure of the AEMET radiation stations dataset	12
Table 3: Structure of the dataset of solar radiation of the day before the call	13
Table 4: Structure of the dataset of solar radiation two days before the data collection day	14
Table 5: Structure of the dataset of hourly climate data for the five days prior to the call.....	15
Table 6: Structure of the dataset of weather forecast for the two days after the call	15
Table 7: Clean dataset structure of climate data from previous 5 days.....	17
Table 8: Structure of the clean dataset of climatological predictions for the next two days	18
Table 9: Structure of the clean dataset of radiation data from the previous day.....	19
Table 10: Structure of the clean radiation data dataset from two days before	20
Table 11: Structure of the model training dataset	26
Table 12: Average values of solar radiation per hour.	28
Table 13: Results of the overfitting check	35
Table 14: Average values of contracted power and electricity consumption	38

1. Introduction

The electricity sector has always been in constant evolution. The process of liberalization of the sector that began in the 1980s was only the beginning of a process of changes that modify the structure of generation are, with the exception of hydroelectric plants, not very mature, this sector as technical and regulatory advances are taking place. In recent years, global climate trends and advances in electricity generation technologies have led to a proliferation of clean and renewable energies. These forms generally modular technologies, so that their generation capacity is small in comparison with conventional generation plants, and typically have dispersed location. Thus, distributed renewable energy sources (DER) appear in the panorama, which represent a challenge for the distribution companies and the organizers of the electricity markets, since they break the traditional generation-transport-distribution-consumption scheme and are difficult to dispatch due to the difficult prediction of its power generation. These changes of decentralization and clean energy involve new forms of network management and the operation of electricity markets, challenges that the current regulation is trying to solve.

Thus, the networks and electrical systems are in constant change due to the continuous technological developments and the existence of an increasingly electrified and changing demand in the use of a good that is considered basic and necessary. Currently there is a long series of disruptive elements in the sector that will make it change in the coming years: RES, climate change policies, demand electrification, etc. The analysis of data, artificial intelligence, the “Internet of things”, the blockchain and other technologies and innovations can suppose a great advance in the sector if optimally applied to improve the prediction of intermittent generation, the sale of energy between pairs, demand management, etc.

On the other hand, all these trends in the sector lead to an increasing use of distributed electricity generation resources for self-consumption. In the Spanish case, the use of shared self-consumption has recently been legally approved. However, the need for an efficient and as fair way as possible to distribute the energy generated among the different users is evident.

Self-consumption facilities must play an important role in this energy transformation. After the approval of RD 244/ 2019 the door is open for an unprecedented development of shared self-consumption in Spain. After the elimination of the “sun tax” and the administrative simplification the legalization of this type of facilities, the situation is very favorable for the proliferation of this type of systems. This is expected to reduce energy costs while contributing to a cleaner and renewable electricity system. However, for this situation to be optimal, algorithms and ways to use this shared energy must be designed so that all parties benefit.

Thus, currently, the electric sector is going through substantial changes in its structure that should lead to a more decentralized system and where the operating costs of the different utilities are lower so that their profit margins grow, thus encouraging them to try to give better customer service. The future is also based on a system where clean and renewable energy is very present, helping to eliminate, along with the electrification of demand, the current problems of global warming and climate change. Therefore, for domestic users to be able to make efficient use of generation resources and to obtain the highest possible economic compensation for unconsumed spillages, accessible and reliable tools are required that allow them to know in advance the production that they will be able to obtain from its photovoltaic plants.

2. Objective

The purpose of this work will be to develop a model that allows domestic owners of photovoltaic installations to know in advance a reliable estimate of the production of electrical energy that they will be able to have at each hour of the day, so that they can take proper advantage of this generation. These type of prediction systems are usually paid and developed *ad hoc* for the characteristics of a specific station, or dependent on the connection to the inverter of each plant. Thus, it is intended to provide in this work a free tool focused on predicting the electrical production of a photovoltaic panel system for any location in Spain. The aim is to encourage domestic consumers to make more efficient use of available generation resources, so as to save on their electricity bill, and reduce the cost of electricity production in the market for other users of the grid, while reducing the emission of greenhouse gases.

3. Requirements and preparation

3.1. Requirements

This project has been developed in the Linux operating system (Ubuntu), specifically in the kernel **xubuntu 20.04**. To execute this project, it will be necessary to have installed the last version of **Anaconda**. Most libraries used by this project are included by this distribution.

In order to have the same libraries and versions necessary for the execution of the project, you can update your environment (“env_name”) with the attached **environment.yml** file:

```
conda env update --name <env_name> --file environment.yml
```

Beyond the base *Conda* package, the following installations have been run:

```
pip install ipynb
pip install python-crontab
conda install croniter -y
pip install --upgrade tensorflow
pip install streamlit-drawable-canvas
pip install streamlit
pip install geopandas
pip install streamlit_folium
```

3.2. Execution instructions and folder structure

In addition, in order to replicate the project on another machine, the GitHub repository that contains it must be cloned: <https://github.com/ruizber23/Proyecto-fin-de-master>.

The folder structure is as follows:

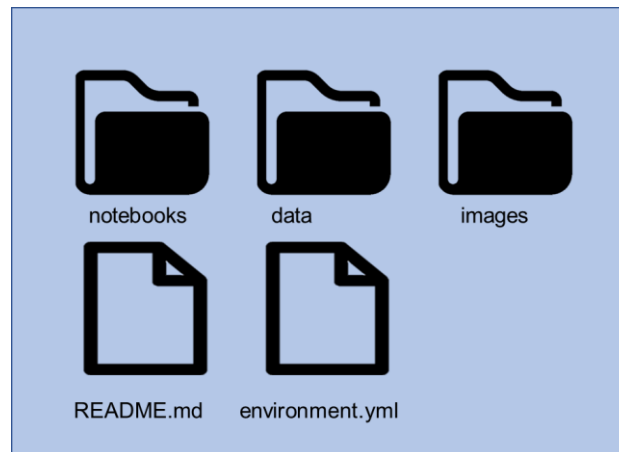


Image 1: Folder structure of the TFM folder

- **notebooks**: contains the jupyter notebooks (.ipynb) needed to run the project.
- **data**: contains the necessary data for training the model. These should be downloaded from the attached shared folder and extracted into it. Throughout this document the origin and function of these files will be explained. This folder will contain:
 - **Modelo**: this folder contains the necessary files for the use of the prediction models.

- Historicos: this folder contains the clean data, so that it can be used for the training of the models.
- Rad_SODA: this folder contains the daily CAMS solar radiation files.
- Clima_OW: this folder contains the daily weather data files from the previous 5 days obtained from OpenWeather.
- Pred_OW: this folder contains the next 48 hour daily weather forecast files obtained from OpenWeather.
- Rad_AEMET: this folder contains the daily solar radiation files of AEMET.
- estaciones.csv: data to be downloaded from AEMET.
- estaciones_rad.csv: data to be downloaded from AEMET.
- perfiles_consumo.csv: self-created file. Standard profiles of consumption of 6 generic user types. Consumption of each hour of the day and profile as part of the total (1).
- dateInfo.txt: text file that records the success or failure of daily data downloads.

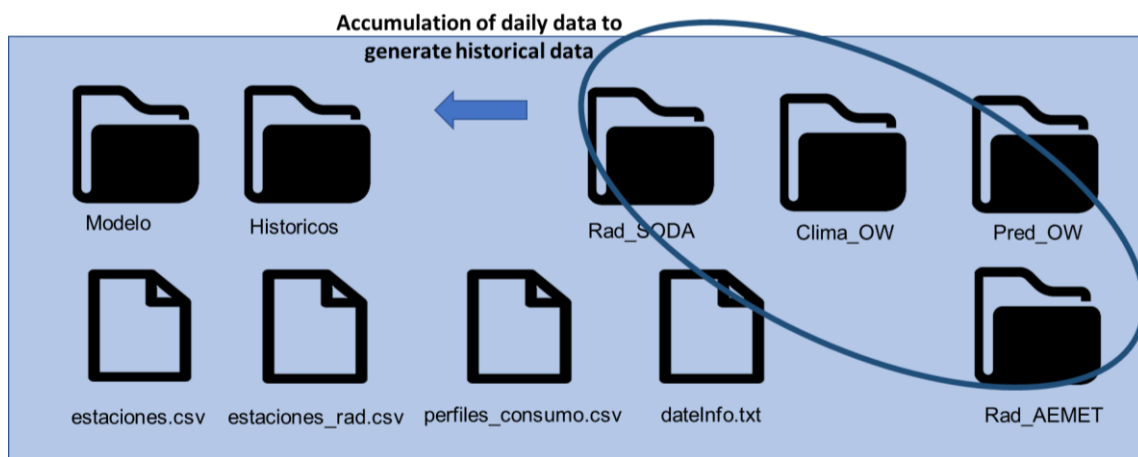


Image 2: Folder structure of the data folder

- **images:** contains the images used in the *Streamlit* interface and in the README file.
- **environment.yml:** in order to quickly reproduce the project environment, with all its packages and versions, this file has been generated.

All of this must be in a folder called *TFM*. In the case of the machine where the project was developed, the directory was: `/home/dsc/git/TFM /`.

At the beginning of the different notebooks the following commands are executed:

```
%cd /home/dsc/git/TFM/
directorio = '/home/dsc/git/TFM/'
```

If you want to run a notebook, just change this directory in these cells, replacing it with the one where everything is downloaded (TFM folder).

4. Methodology

The methodology used in this project will be the Model Development Process (MDP), which consists of four phases that are based on the knowledge acquired in the previous ones. The four phases proposed by the model are the following:

- Formulation of the problem: In this phase the problem to be treated is defined and the data with which to work is defined and prepared (acquisition, cleaning, ...)
- Modeling: In this phase, a prototype is generated that will allow us to make sure that what is being done is what is wanted.
- Settings. Adjustment phase of the models, to choose the most suitable one.
- Maintenance and monitoring. In this phase, the performance of the models is monitored in production environments.

MDP :: Model Development Process

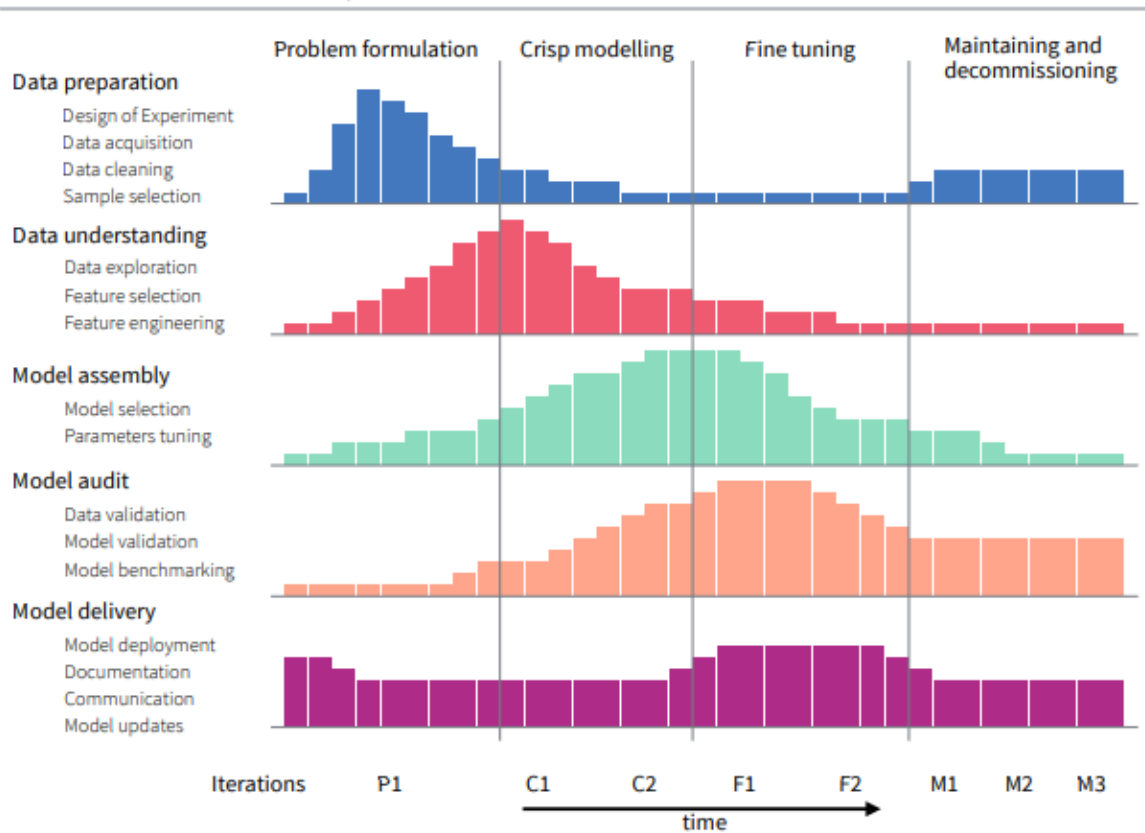


Image 3: MDP Methodology

4.1. Operating scheme

The following image summarizes the work flow (development) of the project:

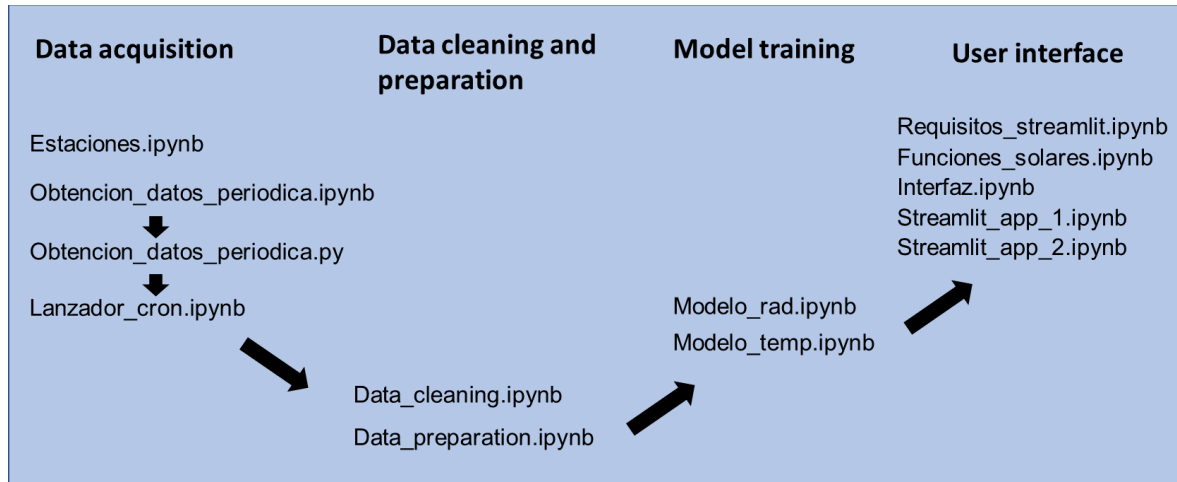


Image 4: TFM operating diagram

The process that has been followed for the development of this TFM is as follows:

- In the first place, the necessary data have been obtained for the elaboration of the models.
 - In the **Estaciones.ipynb** notebook the lists of meteorological and radiation stations from AEMET are downloaded¹.
 - In **Obtencion_datos_periodica.ipynb** daily data are obtained for the prediction models. Runs daily via notebook **Lanzador_cron.ipynb**, in its forma: **Obtencion_datos_periodica.py**.

In a first attempt to generate a prediction system, daily historical meteorological data were used, since there is a large historical base. However, we want to use an hourly database to generate a more accurate model (with equal amount of data). For this reason, the script that downloads hourly data is executed every day through the **Lanzador_cron.ipynb** notebook to generate, by accumulating these daily files, a historical database for training the prediction models.

- Afterwards, the data from the models has been cleaned up and prepared for training. Notebooks: **Data_cleaning.ipynb** y **Data_preparation.ipynb**.
- Once these data are obtained, the solar radiation on a horizontal plane prediction model is trained in the **Modelo_rad.ipynb** notebook, and the ambient temperature prediction model in the **Modelo_temp.ipynb** notebook. Thus, two models are generated, tested and trained, which will predict the solar radiation and the ambient temperature in the desired location on an hourly basis.
- Finally, there is a series of notebooks that allow the user interface to be displayed locally through a *Streamlit* app, where any owner of a photovoltaic installation can enter its data and obtain a prediction of electricity production for the next day. Notebooks: **Script_funcional.ipynb**, **Requisitos_streamlit.ipynb**, **Funciones_solares.ipynb**, **Interfaz.ipynb**, **Streamlit_app_1.ipynb** y **Streamlit_app_2.ipynb**.

¹ <http://www.aemet.es/es/portada>

Different locations within the national territory have been used to obtain the data and train the models. These points are the AEMET weather stations, which are obtained in the script **Estaciones.ipynb**. These locations have been selected because they are adequately distributed throughout the different areas of the country. Since these stations do not coincide with the AEMET radiation stations, from which data will be extracted, this does not pose any problem when training the models.

4.2. Data obtention

The data that will be available each day for the operation of the interface will be:

- Solar radiation data from two days before: will be obtained from CAMS Radiation Service, a photovoltaic information system of the European Commission.
- Hourly weather data for the last 5 days: Obtained from OpenWeather.
- Weather forecast data for the next 48 hours: Obtained from OpenWeather.
- Hourly solar radiation data for the previous day for the different AEMET radiation stations: Obtained from Aemet Open Data.

Therefore, in order to train the different prediction models, these data have been obtained on a daily basis for different locations, in order to generate a historical base of them.

Data collection has been divided into two stages, which are detailed in the following subsections:

4.2.1. Obtaining data from AEMET² stations

Notebook: Estaciones.ipynb

This notebook is used to download the data from the AMET³ (Agencia Estatal de Meteorología) stations:

- Weather stations: These are stations that record daily values of different meteorological variables. As they are distributed throughout the country, they are used as a varied group of locations from which to obtain data, as example geographical points when developing the models and obtaining the different data. The obtained dataset has the following structure:

Dataset column name	Field description	Example value
latitud	Station latitude	413515N
provincia	Station province	BARCELONA
altitud	Station altitude (m)	74

² <http://www.aemet.es/es/portada>

³ <https://opendata.aemet.es/centrodedescargas/productosAEMET>

indicativo	Identification code of the AEMET weather station	0252D
nombre	Station location	ARENYS DE MAR
longitud	Station longitude	023224E

Table 1: Structure of the AEMET meteorological stations dataset

It is saved as **estaciones.csv**, in the *data* folder.

- Radiation stations: They measure the values of solar radiation every day. The obtained dataset has the following structure:

Dataset column name	Field description	Example value
Estación	Station location	A Coruña
indicativo	Identification code of the AEMET weather station	1387
latitud	Station latitude	432157N
longitud	Station longitude	082517W

Table 2: Structure of the AEMET radiation stations dataset

For each station, its coordinates and the indicative code of the station are obtained, after cleaning the dataset and eliminating repeated stations. It is saved as **estaciones_rad.csv**, in the *data* folder.

Thus, the different climatic stations will be used as geographical points for the training of the prediction models. Each point will have the coordinates and the indicative code of one of these stations.

4.2.2. Historical data obtention

Notebooks: Obtencion_datos_periodica.ipynb, Lanzador_cron.ipynb

This notebook will be used to obtain the historical data required for the prediction models:

- Solar radiation of the day before the call⁴: These data are only available for the different radiation stations. These are accumulated hourly data (in true solar time) of global, direct, diffuse and infrared solar radiation on a horizontal surface. This data is also obtained from the AEMET Opendata portal. The fields obtained for each day are, among others:

Dataset column name	Field description	Example value
fecha	Date	06/04/2021

⁴ <https://opendata.aemet.es/centrodedescargas/productosAEMET>

Estación	Name of the radiation station from which the data is obtained	ALBACETE
Indicativo	Identification code of the radiation station from which the data is obtained	8178D
Tipo	Measured variable (Global/Diffuse/Direct/Erythematic UV/Infrared)	GL
GL/DF/DT	Global/Diffuse/Direct hourly radiation accumulated between: (indicated hour -1) and (indicated hour). Between 5:00 and 20:00. True Solar Time. Units: (10 * kJ/m ²)	6
UVB	Semi-hourly radiation accumulated between: (hour: indicated minutes - 30 minutes and (hour: indicated minutes) between 4:30 and 20:00. True Solar Time. Variables: Erythematic Ultraviolet Radiation (J/m ²)	190
IR	Hourly radiation accumulated between (indicated hour -1) and (indicated hour) between 1:00 and 24:00. True Solar Time. Variables: Infrared radiation (10 * kJ/m ²)	99

Table 3: Structure of the dataset of solar radiation of the day before the call

It will not be necessary to transform the time, since the true solar time corresponds approximately to UTC.

They are saved as “**rad_aemet_{date}**” in the *Rad_AEMET* folder of the *data* folder.

- Solar radiation of two days before the day of data collection⁵: These data are obtained from the CAMS Radiation Service portal of the European Union. They are data in UTC time. Provides solar radiation for any date up to 2 days before the call. In addition, a column has been added with the identifier of the meteorological station for whose location the data was taken. The fields obtained for each day are:

Dataset column name	Field description	Example value
dateBegins	Beginning of the time period with the format "yyyy-mm-ddTHH:MM:SS.S/yyyy-mm-ddTHH:MM:SS.S"	2021-04-18T01:00:00.0
dateEnds	End of the time period with the format "yyyy-mm-ddTHH:MM:SS.S/yyyy-mm-ddTHH:MM:SS.S"	2021-04-18T02:00:00.0
toa	Irradiation on horizontal plane at the top of atmosphere (Wh/m ²) computed from Solar Geometry 2	90.31
cs_ghi	Clear sky global irradiation on horizontal plane at ground level (Wh/m ²)	40.29
cs_bhi	Clear sky beam irradiation on horizontal plane at ground level (Wh/m ²)	20.17
cs_dhi	Clear sky diffuse irradiation on horizontal plane at ground level (Wh/m ²)	20.12
cs_bni	Clear sky beam irradiation on mobile plane following the sun at normal incidence (Wh/m ²)	186.84

⁵ <http://www.soda-pro.com/web-services/radiation/cams-radiation-service>

ghi	Global irradiation on horizontal plane at ground level (Wh/m ²)	33.43
bhi	Beam irradiation on horizontal plane at ground level (Wh/m ²)	10.5
dhi	Diffuse irradiation on horizontal plane at ground level (Wh/m ²)	22.93
bni	Beam irradiation on mobile plane following the sun at normal incidence (Wh/m ²)	75.0727
reliability	Proportion of reliable data in the summarization (0-1)	0.8667
estacion	Identifying code of the weather station for whose location the data is obtained	0252D

Table 4: Structure of the dataset of solar radiation two days before the data collection day

They are saved as “**rad_soda_{date}**” in the *Rad_SODA* folder of the *data* folder.

CAMS Radiation Service only allows 100 calls per day to its API for each registered account. Therefore, and taking into account that a call is already made each time the *Streamlit* app (frontend) is used to predict the electricity production of the next day, it is recommended **not to run this part of the notebook more than once a day**, in order not to exceed the number of calls per account (entails blocking).

- Hourly weather data for the five days prior to the call⁶: This data is obtained from the OpenWeather portal (thanks to a student license that allows a large number of calls per day). The data is in UTC time. The fields obtained are:

Dataset column name	Field description	Example value
date	Date	03/04/2021
hour	Hour of historical data, UTC	21:00
fecha_prediccion	Date the call was made	06/04/2021
estacion	Identification code of the climatic station for whose location the data is obtained	0252D
temp	Temperature. Units: kelvin	288.13
feels_like	Temperature. This accounts for the human perception of weather. Units: kelvin.	287.09
pressure	Atmospheric pressure on the sea level, hPa	1013
humidity	Humidity, %	72
dew_point	Atmospheric temperature below which water droplets begin to condense and dew can form. Units: kelvin	283.13
clouds	Cloudiness, %	0
visibility	Average visibility, metres	10000

⁶ <https://openweathermap.org/api/one-call-api#history>

wind_speed	Wind speed. Wind speed. Units: m/s	1.54
wind_deg	Wind direction, degrees (meteorological)	90
wind_gust	Wind gust. Units: m/s	1.96
we	Weather id	501
rain.1h	Precipitation volume, mm	1.02
snow.1h	Snow volume for last hour, mm	0.6

Table 5: Structure of the dataset of hourly climate data for the five days prior to the call

They are saved as "**clima_ow_ {date}**" in the *Clima_OW* folder of the *data* folder.

- Weather forecast for the two days after the call⁷: This data is obtained from the OpenWeather portal (thanks to a student license that allows a large number of calls per day). The data is in UTC time. The hourly weather forecast for the 48 hours following the call is accessed. The fields obtained are:

Dataset column name	Field description	Example value
date	Date	07/04/2021
hour	Hour of historical data, UTC	13:00
fecha_prediccion	Date the call was made	07/04/2021
estacion	Identification code of the climatic station for whose location the data is obtained	0252D
temp	Temperature. Units: kelvin	283.25
feels_like	Temperature. This accounts for the human perception of weather. Units: kelvin.	282.12
pressure	Atmospheric pressure on the sea level, hPa	1022
humidity	Humidity, %	69
dew_point	Atmospheric temperature below which water droplets begin to condense and dew can form. Units: kelvin	277.83
uvi	UV index	3.6
clouds	Cloudiness, %	99
visibility	Average visibility, metres	10000
wind_speed	Wind speed. Wind speed. Units: m/s	0.64
wind_deg	Wind direction, degrees (meteorological)	14
wind_gust	Wind gust. Units: m/s	1.68
pop	Probability of precipitation	0
we	Weather id	804
rain.1h	Rain volume for last hour, mm	0.12
snow.1h	Snow volume for last hour, mm	0.6

Table 6: Structure of the dataset of weather forecast for the two days after the call

They are saved as "**pred_ow_ {date}**" in the *Pred_OW* folder of the *data* folder.

⁷ <https://openweathermap.org/api/one-call-api>

In order to generate a historical database with which to train the models, it is necessary to obtain the four groups of data mentioned for each day and location (meteorological stations). For this reason, to be able to execute it daily automatically using cron, this notebook is converted to .py format (**Obtencion_datos_periodica.py**). Afterwards, its daily execution is automated using the **Lanzador_cron.ipynb** script.

4.3. Data preparation for model training

In order to be able to correctly develop the different prediction models, a previous processing, cleaning and preparation of the downloaded data in an appropriate format is required. In addition, all useful data must be gathered in a single dataframe.

4.3.1. Data cleansing

Notebook: Data_cleaning.ipynb

First, the so-called “useful hours” are defined (4:00-20:00), which define a time range outside of which radiation will always be zero. Therefore, only data that is within this time range is taken.

- Climate data from 5 previous days

This is the hourly weather data for the 5 days prior to the call. They are in UTC time. Each hour (X:00) contains the data associated with the starting time (X:00 - X:59).

Data is studied:

- The rows (observations) that correspond to stations that are not in the downloaded AEMET meteorological station list are eliminated, looking for call signs that are not present in it. The data is filtered to leave those within the useful hours range. As the data is downloaded in daily files, all the days that have been obtained are concatenated in a single dataset so that a historical data is available.
- It is detected that the visibility and wind_gust fields have a percentage of null data (NAs) of 29 y 67% respectively, so the missing data is filled in with the mean value of the field in question. The “rain.1h” and “snow.1h” fields have more than 92% of missing data, so they are removed from the dataset.
- Possible duplicate rows are eliminated.
- Finally, a function is generated that encompasses the entire process.

The remaining fields after cleaning are:

Dataset column name	Field description	Example value
date	Date	04/04/2021
hour	UTC time	4
fecha_prediccion	Date the call was made	05/04/2021
estacion	Identification code of the climatic station for whose location the data is obtained	0252D
temp	Temperature. Units: kelvin	284.3

feels_like	Temperature. This accounts for the human perception of weather. Units: kelvin.	282.56
pressure	Atmospheric pressure on the sea level, hPa	1014
humidity	Humidity, %	93
dew_point	Atmospheric temperature below which water droplets begin to condense and dew can form. Units: kelvin	283.21
clouds	Cloudiness, %	0
visibility	Average visibility, metres	10000
wind_speed	Wind speed. Units: m/s	2.57
wind_deg	Wind direction, degrees (meteorological)	280
wind_gust	Wind gust. Units: m/s	4.78241474
we	Weather id ⁸	800

Table 7: Clean dataset structure of climate data from previous 5 days

They are saved as "**historicos_climaticos_clean.csv**" in the *Historicos* folder of the *data* folder.

- Weather predictions for the next 2 days

This is the hourly weather forecast for the 48 hours following the call. They are in UTC time. Each hour (X:00) contains the data associated with the starting time (X:00 - X:59).

Data is studied:

- The rows (observations) that correspond to stations that are not in the downloaded AEMET meteorological station list are eliminated, looking for call signs that are not present in it. The data is filtered to leave those within the useful hours range. As the data is downloaded in daily files, all the days that have been obtained are concatenated in a single dataset so that a historical data is available.
- The hour field is converted to numeric.
- It is detected that the "rain.1h" and "snow.1h" fields have more than 88 % of missing data, so they are removed from the dataset.
- Possible duplicate rows are eliminated.
- Finally, a function is generated that encompasses the entire process.

The remaining fields after cleaning are:

Dataset column name	Field description	Example value
date	Date	05/04/2021
hour	UTC time	5
fecha_prediccion	Date the call was made	05/04/2021
estacion	Identification code of the climatic station for whose location the data is obtained	0252D
temp	Temperature. Units: kelvin	285.54

⁸ <https://openweathermap.org/weather-conditions#Weather-Condition-Codes-2>

feels_like	Temperature. This accounts for the human perception of weather. Units: kelvin.	284.66
pressure	Atmospheric pressure on the sea level, hPa	1017
humidity	Humidity, %	70
dew_point	Atmospheric temperature below which water droplets begin to condense and dew can form. Units: kelvin	279.91
uvi	UV index	0
clouds	Cloudiness, %	65
visibility	Average visibility, metres	10000
wind_speed	Wind speed. Wind speed. Units: m/s	1.18
wind_deg	Wind direction, degrees (meteorological)	10
wind_gust	Wind gust. Units: m/s	1.43
pop	Probability of precipitation	0
we	Weather id	803

Table 8: Structure of the clean dataset of climatological predictions for the next two days

They are saved as "**predicciones_climaticas_clean.csv**" in the *Historicos* folder of the *data* folder.

- Radiation data from the previous day

These data are only available for the different AEMET radiation stations.

It is hourly data (true solar time) accumulated from global, direct, diffuse and infrared radiation. Each hour (X:00) contains the data associated with the hour that ends ((X-1):00 - X:00).

Data is studied:

- The rows (observations) that correspond to stations that are not in the downloaded AEMET radiation station list are eliminated, looking for callsigns not present in it. As the data is downloaded in daily files, all the days that have been obtained are concatenated in a single dataset so that a historical data is available.
- The downloaded data has a structure such that each row corresponds to a day, the different columns being the hourly values of the different fields. Therefore, a dataset is generated in which each row corresponds to a time and each column to a field.
- Diffuse and direct radiation fields, which are components of the global radiation, are eliminated.
- The hour field is converted to a numeric type and the fields expressed in 10 KJ/m² and J/m² to W/m².
- It is detected that the IR and UVB fields have a percentage of null data (NAs) of 28 and 37% respectively, so the missing data are filled in with the mean value of the field in question. The GL field has 3.8% missing data, so these hours are removed from the dataset.
- Possible duplicate rows are eliminated.
- Finally, a function is generated that encompasses the entire process.

The remaining fields after cleaning are:

Dataset column name	Field description	Example value
fecha	Date	05/04/2021
hora	Hour	5
estacion	Name of the weather station for whose location the data is obtained	A CORUÑA
indicativo	Identification code of the climatic station for whose location the data is obtained	1387
GL	Cumulative global hourly radiation (W/m ²)	682.3
UVB	Erythematic Ultraviolet Radiation (W/m ²)	0.13
IR	Infrared radiation (W/m ²)	266.666667

Table 9: Structure of the clean dataset of radiation data from the previous day

They are saved as "**rad_aemet_clean.csv**" in the *Historicos* folder of the *data* folder.

- Radiation data from two days before

This is the hourly solar radiation data for any date up to 2 days before the call. They are in UTC time. Each hour (X:00) contains the data associated with the starting time (X:00 - X:59).

Data is studied:

- The rows (observations) corresponding to stations that are not in the downloaded AEMET meteorological station list are eliminated, looking for callsigns not present in it. As the data is downloaded in daily files, all the days that have been obtained are concatenated in a single dataset so that a historical data is available.
- The "dateEnds" fields (which indicates the end of the corresponding hour), "bhi", "dhi", "bni", "toa", "reliability" and the clear sky radiation fields ("cs_XXXX") are eliminated. These data are not relevant since what is important is to know the real global radiation on the horizontal surface.
- Rows with NAs are eliminated.
- From the original date column, the date and hour fields are obtained, separately. The data is filtered to leave those within the useful hours range.
- Possible duplicate rows are eliminated.
- Finally, a function is generated that encompasses the entire process.

The remaining fields after cleaning are:

Dataset column name	Field description	Example value
date	Beginning of the time period with the format "yyyy-mm-ddTHH:MM:SS.S"	05/04/2021 4:00:00

ghi	Global irradiation on horizontal plane at ground level (Wh/m ²)	0
estacion	Identification code of the climatic station for whose location the data is obtained	0252D
hora	Hour	4
fecha	Date	05/04/2021

Table 10: Structure of the clean radiation data dataset from two days before

They are saved as "**rad_soda_clean.csv**" in the *Historicos* folder of the *data* folder.

- Generation of the rows with the days in columns for the historical weather data

For each day that data is obtained, a dataset is archived with the observations of the five days prior to the call. However, each hour of each day appears as a separate row, so the number of rows for each download is 24 hours multiplied by five days. But we want to have a dataset where each row is one of the 24 hours (then filtered to leave only 16 useful hours) of the day, and where the different columns are the data corresponding to that hour for each day (observations of each field for the day before, observations of each field for the day before this, etc.). That is, for each hour of the day, 5 columns for each field, corresponding to its values for that hour for each of the five previous days. Therefore, a new dataset is generated with the data ordered in this format. The different variables saved (the same as the historical weather data, removing the date variable) will have the format "variable_d-x", where x is the number of days prior to the day the data was obtained. Example: *temp_d-1* (temp field for the previous day), *pressure_d-2* (pressure field for two days before), etc.

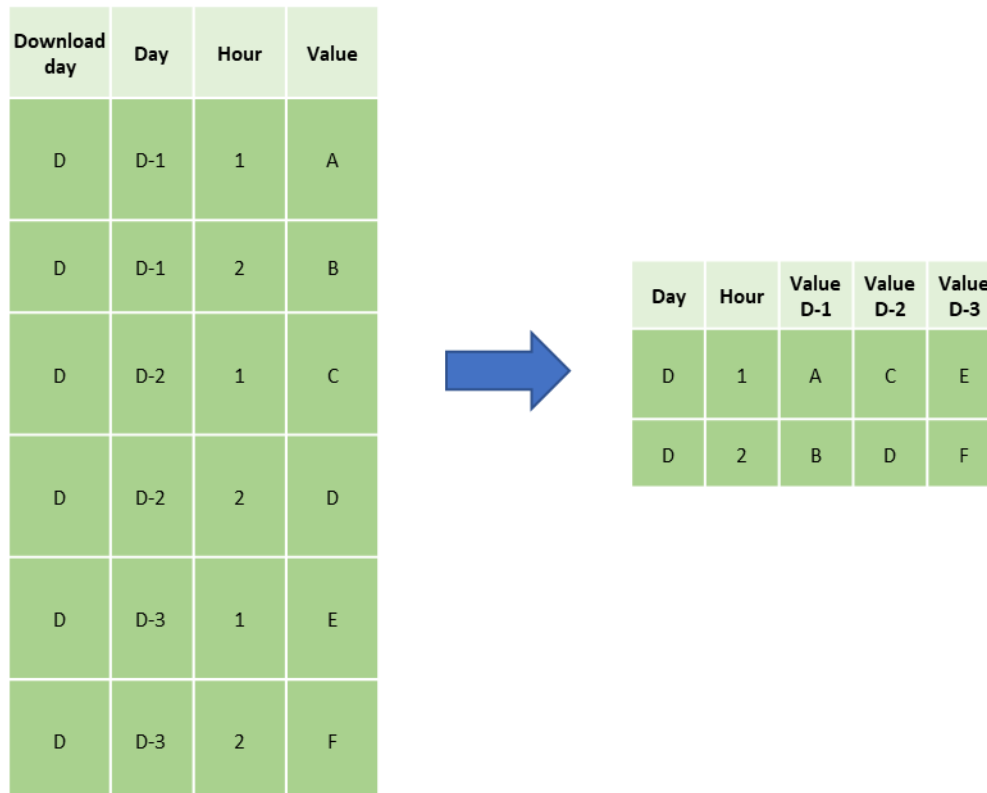


Image 5: Generation of the rows with the days in columns for the historical weather data

They are saved as "**clima_por_horas.csv**" in the *Historicos* folder of the *data* folder.

- Generation of the rows with the days in columns for the historical data of weather prediction

Cada vez que se descargan los datos se obtienen las predicciones climáticas para las 48 horas siguientes a estas (no 2 días completos). Por ello, se decide utilizar una media de los dos valores obtenidos para cada hora del día, ya que siempre se dispondrá de dos por cada hora (en el caso de descargar los archivos al comienzo de un día, se podría tener 2 días completos, pero en el resto de ocasiones se tendrá parte del día en el que se realiza la llamada, el día siguiente completo, y parte del siguiente, hasta sumar 48 horas). Así, se genera un nuevo dataset que, para cada hora del día, cuenta con un valor de predicción medio por cada campo. Las diferentes variables guardadas (las mismas que los datos históricos de predicción de clima, quitando la variable *fecha*) tendrán el formato "*variable_pred*". Ejemplo: *temp_pred* (valor medio de los dos valores del campo *temp* descargados), etc.

In a similar way to what happens in the previous case, for each day that data is obtained, a dataset is archived with the observations of the 48 hours following the call. However, each hour of each day appears as a separate row, so the number of rows for each download is 48 hours. However, we want to have a dataset where each row is one of the 24 hours (then filtered to leave only 16 hours) of the day, and where the different columns are the mean of the data corresponding to that hour for each one of the two days downloaded. Each time the data is downloaded, the climate predictions are obtained for the 48 hours following these (not 2 full days). For this reason, it was decided to use an average of the two values obtained for each hour of the day, since there will always be two for each hour (in the case of downloading the files at the beginning of a day, we could have 2 full days, but on other occasions, there will be part of the day the call is made, the next full day, and part of the next, up to 48 hours). Thus, a new dataset is generated that, for each hour of the day, has an average prediction value for each field. The different saved variables (the same as the historical weather prediction data, removing the date variable) will have the format "default_variable". Example: *temp_pred* (average value of the two values of the downloaded temp field), etc.

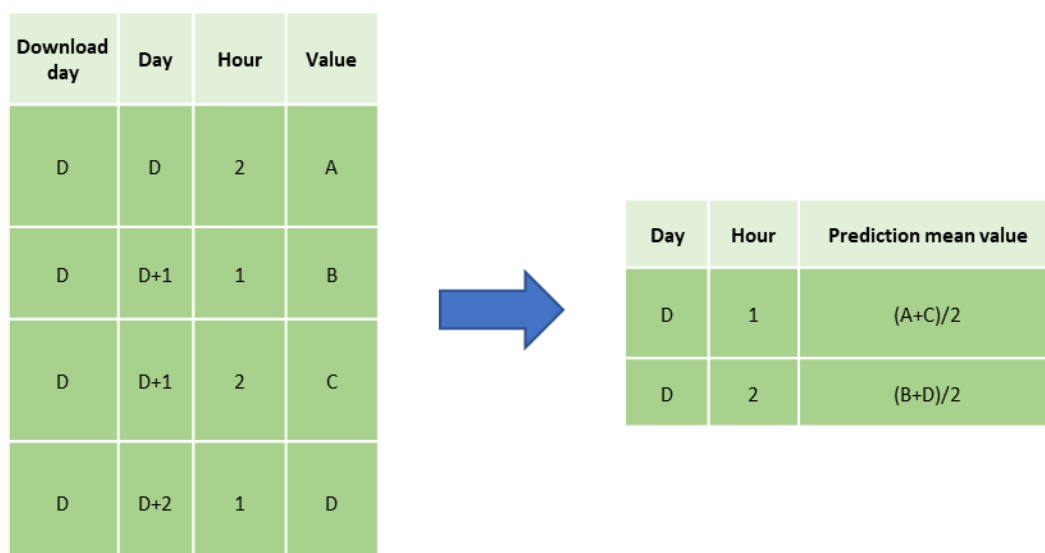


Image 6: Generation of rows with days in columns for historical weather forecast data

They are saved as "**pred_por_horas.csv**" in the *Historicos* folder of the *data* folder.

4.3.2. Data preparation

Notebook: Data_preparation.ipynb

In this notebook, the different data sets are prepared and put together so that they can be used to generate prediction models. In order to join the different data sets, each observation will be associated with a date. In the case of variables used to predict, this date will be the day for which it is wanted to make the prediction based on them. In the case of target variables, this date will be the day they were observed, since they will be the real result of the prediction for that date.

- Historical weather data

The historical climate data dataset is prepared.

- The data are analyzed, checking their distributions and mean values, in order to check that they make sense and that no errors of concept have been made. This is done with the dataset prior to final processing (**Generation of the rows with the days in columns for the historical weather data**), where all the rows correspond to an hour of a specific day.
- A column is added with the date of the day for which it is wanted to predict the solar radiation from these data (day after they were obtained). Thus, this dataset can be joined with the rest of the data sets, taking these dates as a reference.
- The necessary columns are renamed and Kelvin degrees are converted to centigrade.
- A field called “indicativo_rad” is also added. This field represents the indicative code of the AEMET radiation station closest to the location of each observation (to the corresponding meteorological station in the case of historical data, taken for these locations). This call sign will be used to associate the radiation of the previous day taken at the nearest radiation station to the data of each observation.
- In addition, a new dataset is generated with the ambient temperature field, associated with the actual date of data recording. This dataset collects this target variable, therefore, although they are the same data previously associated with the day after it was obtained (for the prediction of solar radiation), in this case they are associated with the date of its registration.

- Weather forecast data

The weather forecast data dataset is prepared.

- The data are analyzed, checking their distributions and mean values. This is done with the dataset prior to final processing (**Generation of the rows with the days in columns for the historical weather forecast data**), where all the rows correspond to an hour of a specific day.
- A column is added with the date of the day for which it is wanted to predict solar radiation from these data (day after they were obtained). Thus, this dataset can be joined with the rest of the data sets, taking these dates as a reference.
- The necessary columns are renamed and Kelvin degrees are converted to centigrade.

- Solar radiation data for the previous day at AEMET radiation stations

The weather forecast data dataset is prepared.

- The data are analyzed, checking their distributions and mean values.

- A column is added with the date of the day for which you want to predict solar radiation from these data (two days after the day they were obtained). Thus, this dataset can be joined with the rest of the data sets, taking these dates as a reference.
- The necessary columns are renamed.
- Solar radiation data

The dataset is prepared.

- The necessary columns are renamed and the date column is removed.
- The data collection date column itself is the date column that is used as a reference to join the data sets.
- In addition, a dataset is generated where the same hourly solar radiation data is associated with the date of two days later, in order to obtain a dataset of the radiation variable two days before the day to be predicted, used in the generation of prediction models.
- Join of datasets

The datasets to join are:

- *df_soda*: Target variable nº1. Global radiation on horizontal plane in W/m² for each hour and weather station.
- *df_objetivos*: Target variable nº2. Average temperature in C°.
- *df_clima*: Climate data for the 5 days prior to the prediction (day prior to the day to be predicted).
- *df_pred*: Weather predictions for the day the prediction is made (day before the day to be predicted) and the day after.
- *df_aemet*: Hourly radiation data in W/m² of the day before the prediction is made (day before the day to be predicted) for each hour and weather station.
- *df_rad_2*: Hourly radiation data in W/m² for two days before the forecast is made (day before the day to be predicted) for each hour and weather station.

The unnecessary columns are eliminated and the different datasets are joined, associating the rows (observations) of each one that share the date, hour and indicative of the meteorological station (radiation station in the case of radiation data from the previous day). It is recalled that the locations of these stations (associated with their indicative code) are the locations taken as examples to obtain the historical data.

Then, the data are studied: number of observations, correlation between variables, NAs...

It is determined that there are variables with a high level of correlation, so **it will be necessary to apply dimensionality reduction on the dataset**. It is verified that the different variables to be used in the prediction do not follow a normal distribution, so **a process of normalization/standardization of the data will also be necessary**.

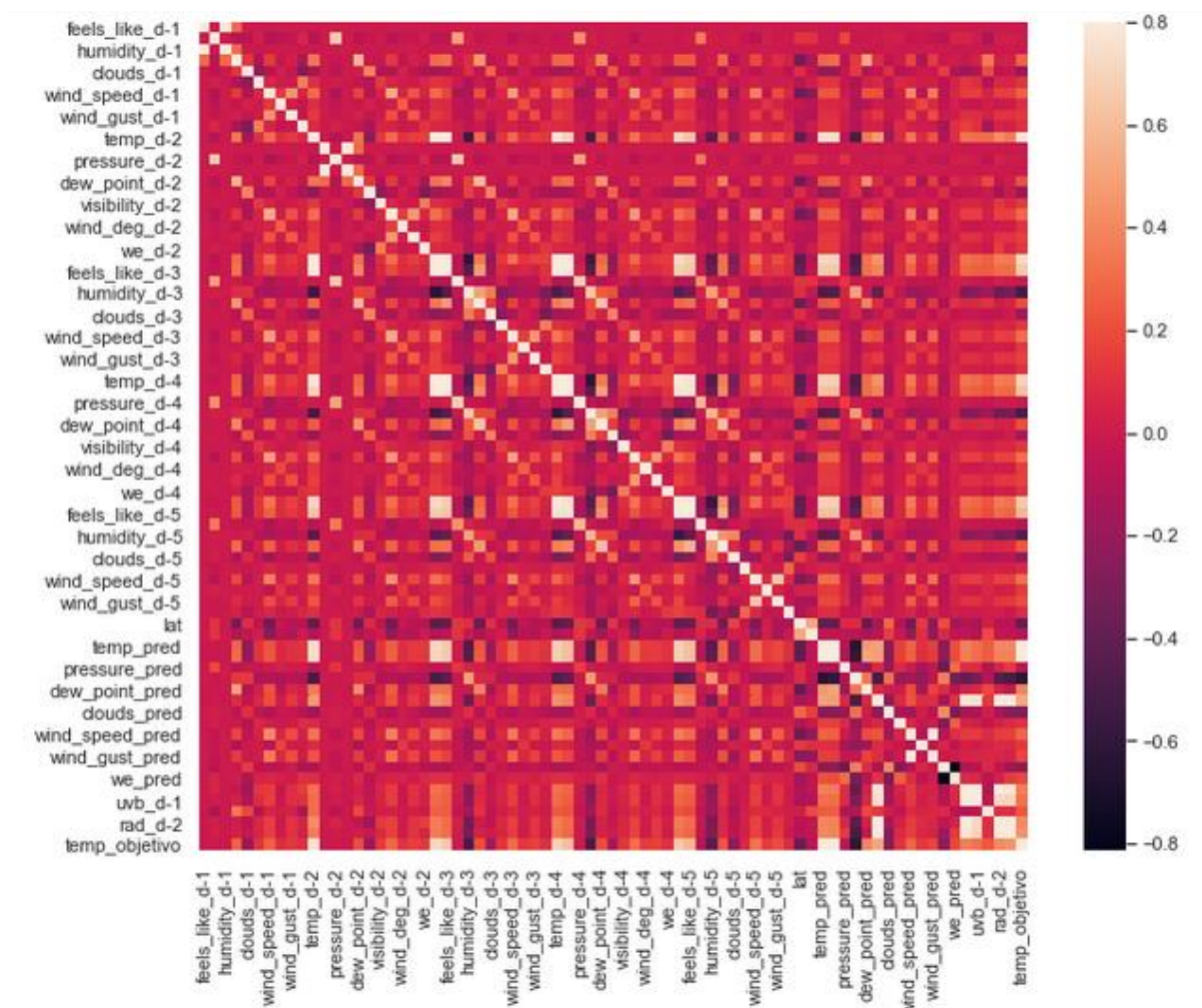


Image 7: Correlation matrix of the different input variables to the prediction models

Finally, the entire process is included in a function. The data set used for the training of the models has the following structure:

Dataset column name	Field description	Example value
fecha_rad	Date of the day to which the solar radiation data corresponds (date to be predicted)	08/04/2021
hora	Hour	4
indicativo	Identification code of the climatic station for whose location the data is obtained	76
temp_d-X/temp_pred	Temperature in centigrade X days before the data download / predicted mean value for the next two hourly values	11.41
feels_like_d-X/feels_like_pred	Wind chill in centigrade X days before data download / predicted mean value for the next two hourly values	8.39

pressure_d-X/pressure_pred	Atmospheric pressure on the sea level, hPa, X days before the data download / predicted mean value for the next two hourly values	1009
humidity_d-X/humidity_pred	Humidity,%, X days prior to data download / predicted mean value for the next two hourly values	87
dew_point_d-X/dew_point_pred	Atmospheric temperature below which water droplets begin to condense and dew can form X days before data download / predicted mean value for the following two hourly values Units: kelvin	282.48
clouds_d-X/clouds_pred	Cloudiness,%, X days prior to data download / predicted mean value for the next two hourly values	20
visibility_d-X/visibility_pred	Average visibility, meters, from X days before the data download / predicted average value for the next two hourly values	7000
wind_speed_d-X/wind_speed_pred	Wind speed (m/s) of X days before the data download / predicted mean value for the next two hourly values	4.12
wind_deg_d-X/wind_deg_pred	Wind direction, degrees (meteorological) of X days before the data download / predicted mean value for the next two hourly values	330
wind_gust_d-X/wind_gust_pred	Wind gust (m / s) of X days before the data download / predicted mean value for the next two hourly values	4.78241474
we_d-X/we_pred	Weather id of X days before the data download / predicted mean value for the next two hourly values	801
lat	Latitude	41.2927778
lon	Longitude	2.07
uvi_pred	UV index, predicted mean value for the next two time values	5
pop_pred	Probability of precipitation, predicted mean value for the next two time values	6
rad_d-1	Cumulative global hourly radiation (W/m ²) of the day before the data download	12
uvb_d-1	Erythematic Ultraviolet Radiation (W/m ²) of the day before the data download	63
ir_d-1	Infrared radiation (W/m ²) of the day before the data download	330.246106

rad_d-2	Global irradiation on horizontal plane at ground level (Wh/m ²) two days prior to data download	56
Ghi	Global irradiation on horizontal plane at ground level (Wh/m ²) to predict	63
temp_objetivo	Ambient temperature to be predicted (degrees centigrade)	8.28

Table 11: Structure of the model training dataset

It is saved as "**datos_modelo.csv**" in the *Historicos* folder of the *data* folder.

4.4. Prediction model training

Notebooks: Modelo_rad.ipynb y Modelo_temp.ipynb

For the regular operation of the application, a prediction model is required that, based on its input data, is capable of predicting hourly solar radiation and ambient temperature for the day after the call is executed. With these predictions, and by means of a model of the operation of a photovoltaic installation and its characteristics, it will be possible to calculate the expected electricity generation per hour of the installation. This prediction model will be generated using the hourly data already mentioned in the data collection section:

- Solar radiation of the day before the call, for AEMET radiation stations
- Solar radiation two days before the day of data collection
- Weather data for the five days prior to the call
- Weather forecast for 48 hours after the call

With the historical set of these accumulated data, these prediction models have been generated and trained, after cleaning and preparing the data.

First, the data is divided into *test* set and *train* set. 30% of these are taken as a *test* set, to test the validity of the models.

4.4.1. Data standardization

Standardization scales each input variable separately by subtracting the mean and dividing by the standard deviation to change the distribution so that it has a mean of zero and a standard deviation of one. For this, the *StandardScaler* method of *sklearn* is used on the train set of the variables with which to make the prediction. The generated scaler is saved and used on the test set.

It is verified that the input variables to the model follow a normal distribution, with zero mean and variance 1.

Scalers are saved as "**scaler_rad.pkl**" and "**scaler_temp.pkl**" in the *Modelo* folder of the *data* folder.

4.4.2. PCA

PCA is applied to the train dataset. Principal Component Analysis (PCA) is a technique used to describe a data set in terms of new uncorrelated variables (components). The PCA looks for the projection according to which the data is best represented in terms of least squares. This converts a set of observations of possibly correlated variables into a set of values of variables without linear

correlation called principal components. It is determined that, although the number of variables in the input set is 75, 62 components will be sufficient, since the accumulated explained variance saturates when using this number of variables (more components will not contribute much to the performance of the model).

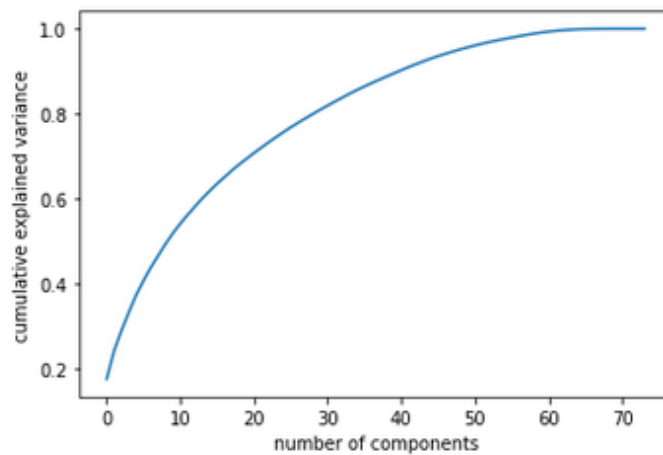


Image 8: Cumulative explained variance versus number of components used

The generated PCA is saved and used on the test set. It is found that the 62 remaining components, new input variables of the model, do not correlate with each other:

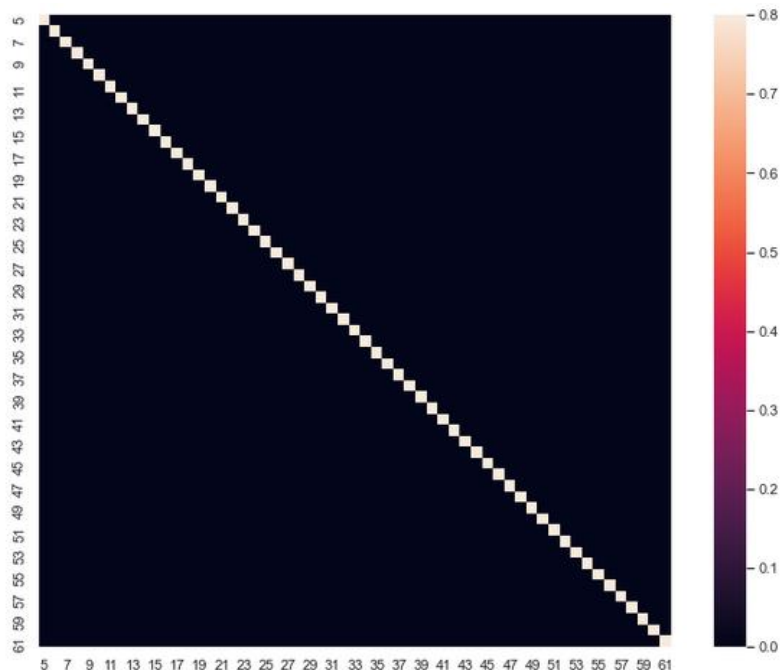


Image 9: Correlation matrix of the different input components to the prediction models

The PCAs are saved as "**pca_rad.pkl**" and "**pca_temp.pkl**" in the *Modelo* folder of the *data* folder.

4.4.3. Model training

The metrics to use will be:

- Mean absolute error (MAE): the error is calculated as an average of absolute differences between the target values and the predictions. All individual differences are weighted equally in the average. It has the scale of the target variable.

- Mean square error (MSE): measures the mean squared error of the predictions. For each point, calculates the square difference between the predictions and the target, and then averages those values.
- Root Mean Square Error (RMSE): is the square root of MSE. It has the scale of the target variable.
- R-squared (r^2): it is closely related to the MSE, but has the advantage of being free of scale. It is always between $-\infty$ and 1.

$$R^2 = 1 - \frac{\text{MSE}(\text{model})}{\text{MSE}(\text{baseline})}$$

$$\text{MSE}(\text{baseline}) = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2$$

The tested machine learning algorithms are:

- Base model: A first model is generated as the basis for comparisons. This simply gets the hourly mean value of the target variable, assigning the predicted value based on the hour of the input data.

Hour	Global irradiation on horizontal plane at ground level (Wh/m ²)
4	0.92
5	34.64
6	142.38
7	295.24
8	453.65
9	594.06
10	704.68
11	784.56
12	802.50
13	739.94
14	657.54
15	533.12
16	387.85
17	231.12
18	89.79
19	9.76

Table 12: Average values of solar radiation per hour.

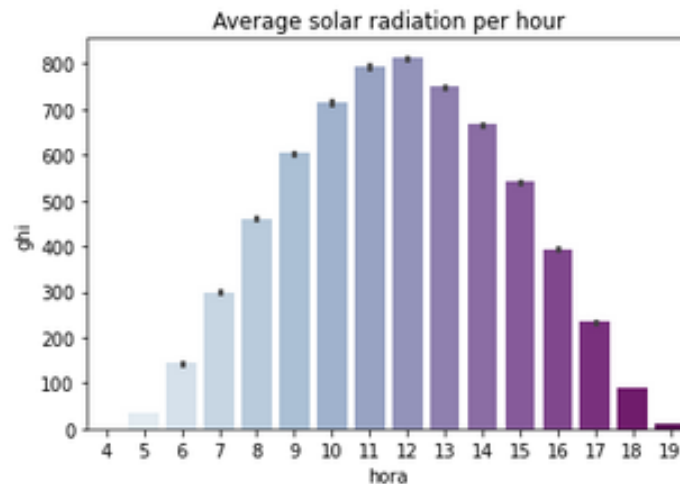


Image 10: Average values of solar radiation per hour

- Linear regression: It is used to estimate real values based on continuous variables. Here, the relationship between the independent and dependent variables is established by a line. This line of fit is known as the regression line and is represented by a linear equation: $Y = a * X + b$, where Y is the variable to predict and X is the input data set.
- k-Nearest Neighbors: Can be used for both classification and regression problems. kNN is a simple algorithm that stores all available cases and classifies new cases by majority vote of their k neighbors. The case assigned to the class is the most common among its k closest neighbors measured by a distance function.
- Decision tree: It is a type of supervised learning algorithm that works for both categorical and continuous dependent variables. It is a predictive model that divides the space of the predictors by grouping observations with similar values for the response or dependent variable.
- Gradient boosting, is a machine learning technique used for regression analysis and for classification problems, which produces a predictive model in the form of a set of weak prediction models, typically decision trees. Builds the model in a stepwise fashion like other *boosting* methods do, and generalizes them allowing arbitrary optimization of a differentiable loss function.
- Random forest: it is a combination of predictor trees such that each tree depends on the values of a random vector tested independently and with the same distribution for each of these. It is a substantial modification of *bagging* that builds a long collection of uncorrelated trees and then averages them.
- GridSearchCV: is a class available in *scikit-learn* that allows to systematically evaluate and select the parameters of a model. By indicating a model and the parameters to test, you can evaluate the performance of the former based on the latter through cross-validation. This function helps to loop through predefined hyperparameters and fit your estimator (model) in your training set.

Therefore, it will be applied to the previously tested models, with r^2 as the metric:

- Linear regression.
- k-Nearest Neighbors.
- Decision tree.
- Gradient boosting.
- Random forest with Regressor RandomizedSearchCV: RandomizedSearchCV implements a "fit" and "score" method. The estimator parameters used to apply these methods are optimized by a cross-validation search on the parameter settings. Unlike GridSearchCV, not all parameter values are tested, but rather a fixed number of parameter settings are sampled from the specified distributions. The number of parameter settings that are tested is given by *n_iter*.
- TensorFlow: It is Google Brain's second-generation machine learning system, released as open-source software on November 9, 2015. TensorFlow is an end-to-end open-source platform for machine learning.

The metrics produced by the different algorithms are analyzed, trying to identify the one that generates the lowest error:

- Solar radiation prediction results: a figure is shown below where the predicted values are observed against the real ones.

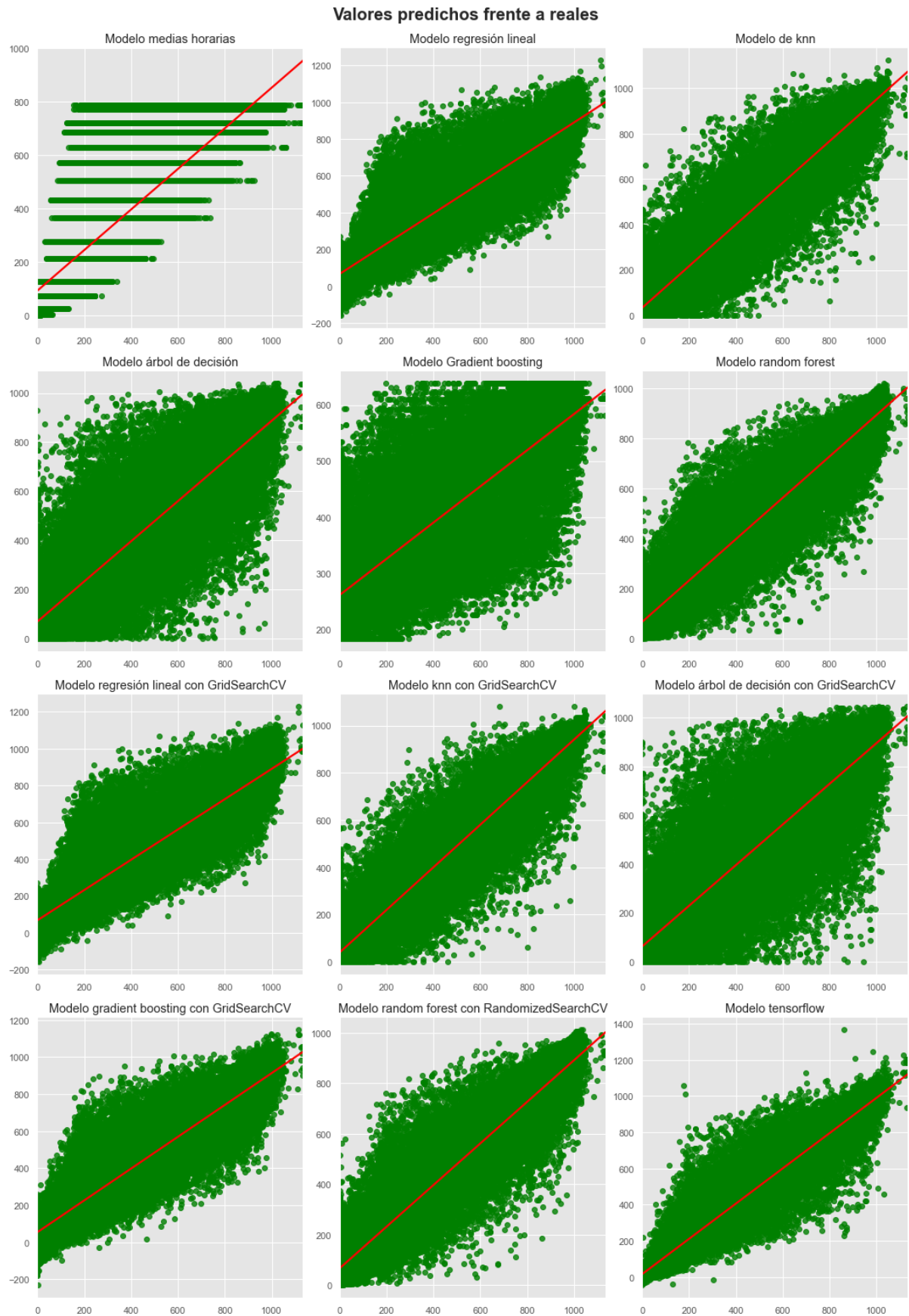


Image 11: Predicted vs. Actual Values of Different Tested Machine Learning Algorithms

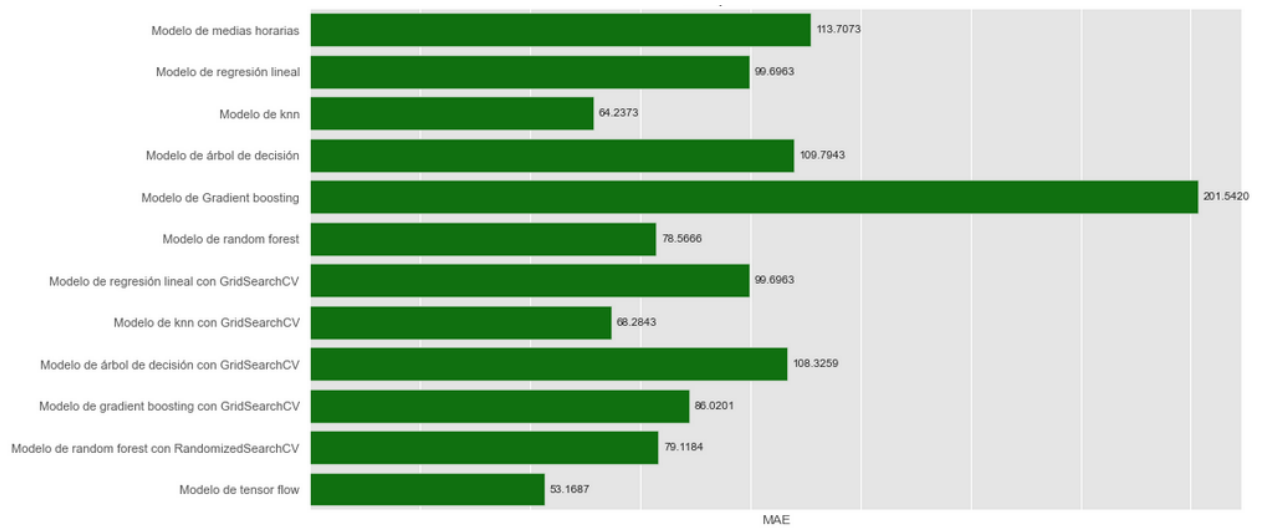


Image 12: MAE of the different tested machine learning algorithms

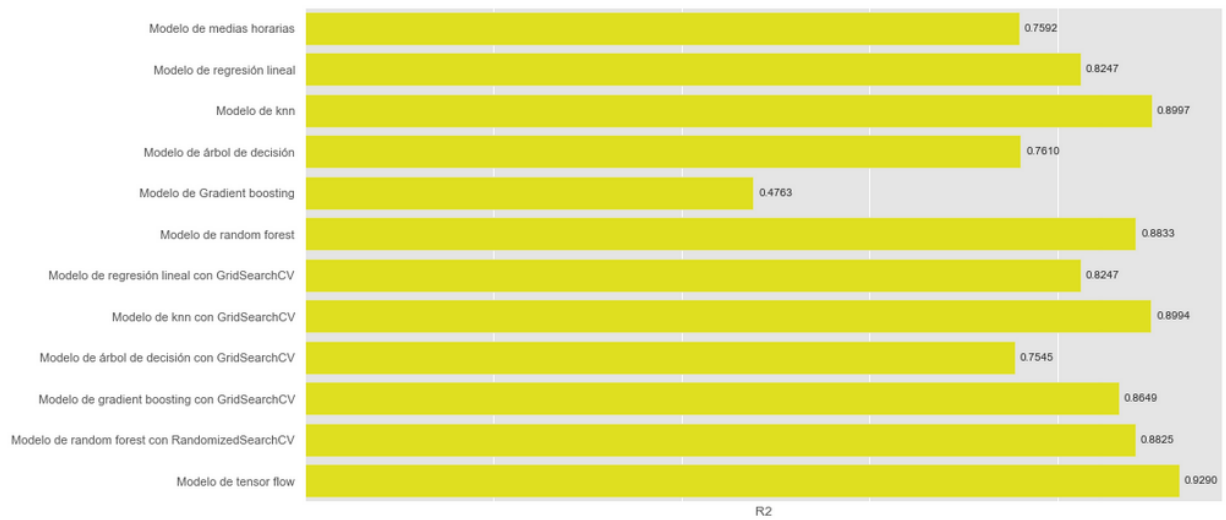


Image 13: R² of the different machine learning algorithms tested

- Ambient temperature prediction results: a figure is shown below where the predicted values are observed against the real ones.

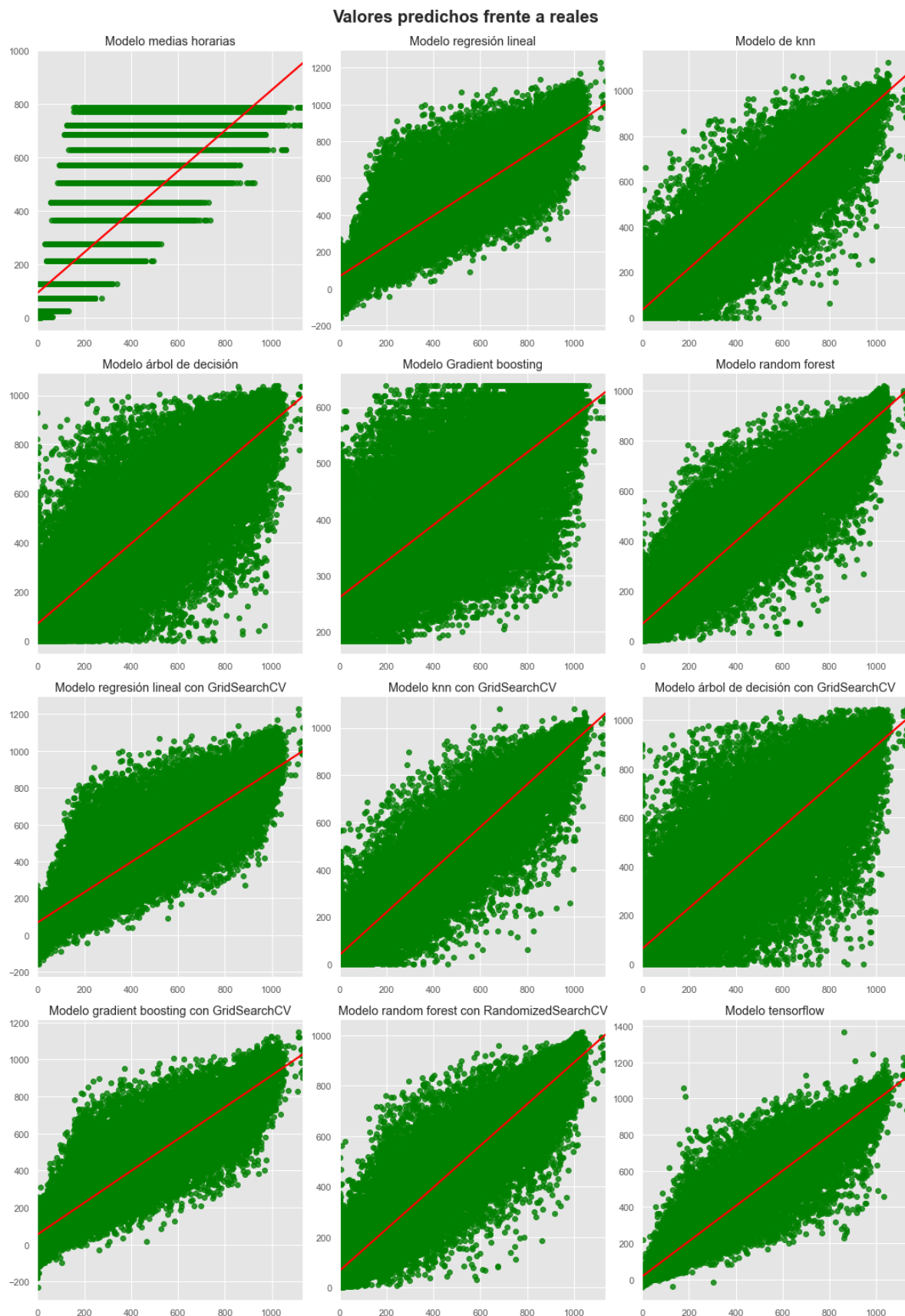


Image 14: Predicted vs. Actual Values of Different Tested Machine Learning Algorithms

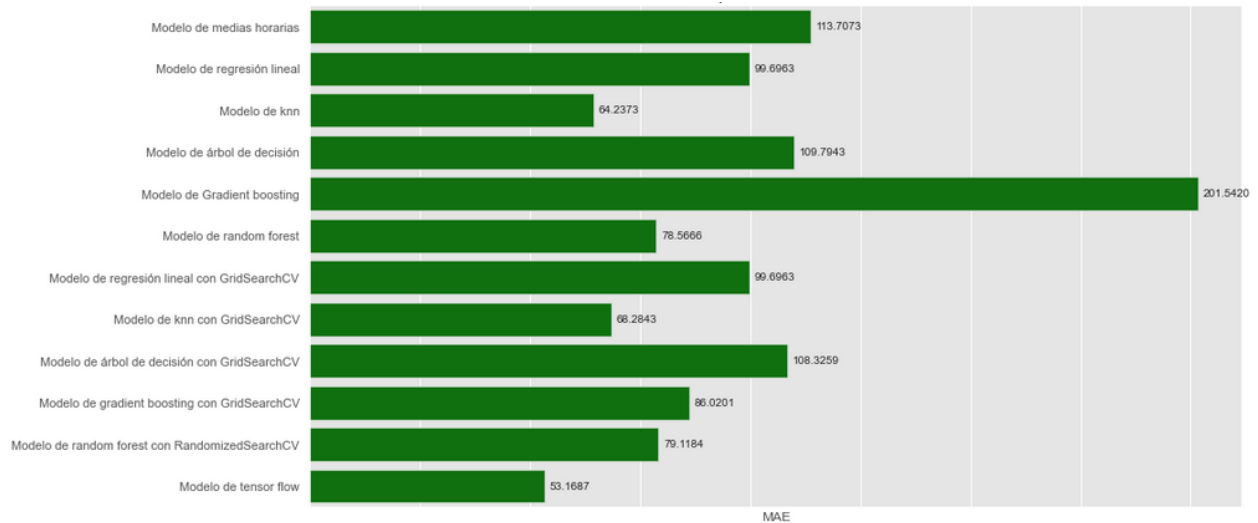


Image 15: MAE of the different tested machine learning algorithms



Image 16: R² of the different machine learning algorithms tested

It is concluded that for both the prediction of solar radiation and that of ambient temperature, the machine learning algorithm that returns the best metrics is the **k-Nearest Neighbors**, which has a higher r^2 (90%) than the rest of the models and a lower mean error (17% of the mean value).

It is checked if the model is overfitted. To do this, a random 60% of the data set is taken 5 times. The same procedure used to generate the model using k-Nearest Neighbors (including train-test division, scaling, and PCA) is applied to each of these new data sets. Then the mean of the r^2 metric of the 5 generated models is obtained. If the model is not overfitted, the result of this mean metric should be similar to the one obtained for the original model.

Since it is not too different, the model is taken as valid. The mean r^2 obtained is somewhat lower, there may be a slight overtraining, probably due to the reduced time range from which data have been obtained.

	R^2	Mean R^2 from overtraining analysis
Solar radiation prediction model	0.89	0.83
Ambient temperature prediction model	0.96	0.66

Table 13: Results of the overfitting check

The adjusted r^2 also results in a similar value. Finally, the models are saved as "model_rad.pkl" and "model_temp.pkl" in the *Modelo* folder of the *data* folder.

4.5. User interface

The interactive interface (*frontend*) for the end users was made through *Streamlit*.

4.5.1. Necessary functions to obtain electricity production

Notebook: Funciones_solares.ipynb

In order to obtain the electrical production of a photovoltaic installation from the data of ambient temperature and solar radiation on the earth's surface (in our case, predictions), a model of the photovoltaic installation is required that, taking into account the **inclination**, the **orientation** and the **peak power** of the installation; returns the electricity generation obtained.

This model will also require functions that determine the value of a series of variables dependent on the day of the year (eccentricity of the earth, declination, angle of sunrise, etc.), as well as functions to determine direct, diffuse solar radiation. and reflected from the characteristics of the installation and the global solar radiation. All these functions are found in the notebook **Funciones_solares.ipynb** and are obtained from "A C Library for Solar Radiation Modelling"⁹.

Regarding the model of a photovoltaic installation (function also present in the mentioned notebook), to obtain the hourly data of generation of the photovoltaic installation, the hourly data of solar radiation (W/m^2) and ambient temperature ($^{\circ}C$) are needed, in addition to other data of the photovoltaic installation as orientation/azimuth ($^{\circ}$), peak power of the installation (kW) or inclination ($^{\circ}$).

Afterwards, the hourly energy generation of the installation will be obtained from these values. Based on the peak power of the installation, the power generated by the photovoltaic panels is calculated by means of the Osterwal method¹⁰, being:

- G : Incidental solar irradiation in the module (W/m^2).
- G_{ref} : Incidental solar irradiation in reference conditions ($1000 W/m^2$).
- γ : Coefficient of losses by temperature, fixed ($-0.48 \text{ }^{\circ}C$).
- W : Average wind speed (3.5 m/s).

⁹ https://inis.iaea.org/search/search.aspx?orig_q=RN:38106953

¹⁰ C. Osterwald, Translation of Device Performance Measurements to Reference Conditions, Solar Cells, 18 (3-4), pp. 269-279, 1986.

- T : Temperature of the photovoltaic installation. In order to obtain the temperature, the King model¹¹ is applied, used in¹², which is formulated as follows:

$$T = T_{amb} + G \cdot e^{(m+n \cdot W)} \quad (1)$$

- m : Empirical dimension coefficient of impact of irradiation on temperature, with value - 3.56¹².
- n : Empirical coefficient that describes the cooling of the installation due to wind, with value -0.079 s/m¹².
- T_{ref} : Reference temperature of the photovoltaic installation (25 °C).

The calculation will be, therefore:

$$P_m = P_{ref} \cdot \frac{G}{G_{ref}} \cdot [1 + \gamma \cdot (T_m - T_{ref})] \quad (2)$$

However, at this generated power it will be necessary to discount the losses that are caused in the DC circuit, by the efficiency of the inverter and in the AC circuit. Those that are not calculated will be assumed, based on¹³:

- Losses by angular reflectance and variation of the radiation spectrum in the photovoltaic module: 3%.
- Residual losses in the wiring: 2%.
- Losses due to the dispersion of parameters in the generator: 2%.
- Losses due to tracking errors of the maximum power point of the inverter: 2%.
- Losses for inverter's performance: To obtain this performance we apply the Jantsch equation¹⁴.

$$\eta_{inv} = \frac{\frac{P_{DC}}{P_{peak}}}{\frac{P_{DC}}{P_{peak}} + (b_0 + b_1 \cdot (P_{DC}/P_{peak}) + b_2 \cdot (P_{DC}/P_{peak})^2)} \quad (3)$$

Where b_0 , b_1 y b_2 are dimensionless parameters of value 0.02, 0.002 and 0.03 respectively. P_{DC} is the input power to the inverter from the DC circuit.

To obtain the inverter's output power:

$$\eta_{inv} \cdot P_{DC} = P_{AC} \quad (4)$$

- Loss in the AC wiring: 1 %.

¹¹ D. King, W. Boyson y J. Kratochvil, Photovoltaic array performance model., Sandia National Laboratories. Report SAND2004-3535, 2004.

¹² P. Mora Segado, Contribución al estudio de la temperatura de módulos FV de diferentes tecnologías en condiciones de sol real, Universidad de Málaga, Servicio de Publicaciones y Divulgación Científica, 2015.

¹³ P. Mora Segado, J. E. Carretero-Rubio y M. Sidrach de Cardona Ortín, Models to predict the operating temperature of different photovoltaic modules in outdoor conditions, Progress in Photovoltaics: Research and Applications, 23 (10), pp. 1267-1282, 2014.

¹⁴ M. Jantsch, H. Schmidt y J. Schmid, Results on the concerted action on power conditioning and control, 11th European Photovoltaic Solar Energy Conference. Montreux, pp. 1589-1592, 1992.

This methodology has been shown to be valid for the calculation of hourly photovoltaic generation in ¹⁵.

4.5.2. Obtaining electricity production

Notebook: Script_funcional.ipynb

For the regular operation of the interface, a code is required to obtain and process the necessary data and then generate the predictions of ambient temperature and solar radiation per hour for a specific location, using previously trained models. Then, taking into account the characteristics of the installation (orientation, inclination and peak power), it will return the electricity production for each hour of the following day using the installation model explained in the previous section.

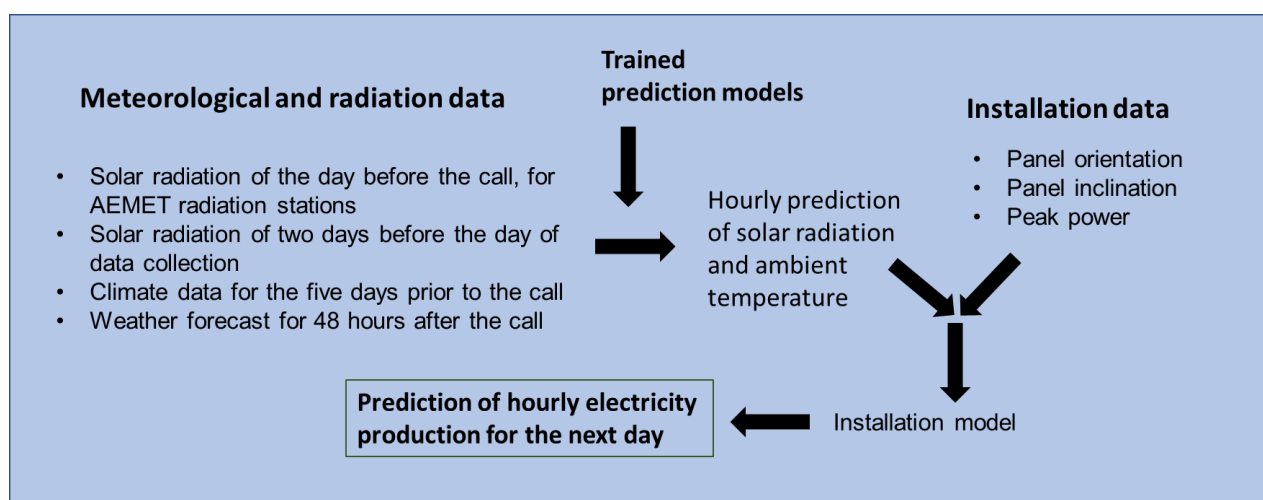


Image 17: Regular operation of the TFM system

To generate this code, the notebook **Script_funcional.ipynb** has been used. However, this will not be used in the operation of the interface, it is only used to experiment and develop the code, since later it must be programmed in a format compatible with *Streamlit*, a tool that will be used to create the user interface (frontend). Therefore, the date, the characteristics of the installation and its location are set by hand as test values.

Thus, this notebook downloads the data necessary to make predictions of solar radiation and ambient temperature. Then clean, join, and process the datasets by following the same steps that were done to train the models (**Data_cleaning.ipynb** and **Data_preparation.ipynb**). However, in this case, the missing values (NA) are replaced by 0. Then, using the trained models, it predicts the hourly values of solar radiation and ambient temperature for the next day. With them, and using the functions script **Funciones_solares.ipynb**, the prediction of electricity production per hour for the next day is obtained.

In addition, it is wanted to show the user an estimate of the income that he will obtain on the predicted day from the sale of his surpluses (economic compensation)¹⁶. For this, the last hourly prices for compensation of surpluses are obtained from the API of esios¹⁷ (Red Eléctrica information system). This price is the one paid to prosumers for the energy generated that they spill into the grid (surpluses, not consumed generation of their photovoltaic installation). In addition, the file **Perfiles_consumo.csv** is used, which contains the typical consumption curves of 6 standard supplies in our country. They consist of 24 values that indicate the relative weight of

¹⁵ C. Cañete Torralvo, Modelos para la caracterización eléctrica de módulos fotovoltaicos en condiciones de sol real, Universidad de Málaga, Servicio de Publicaciones y Divulgación Científica, 2015.

¹⁶ <https://tarifasgasluz.com/autoconsumo/normativa/compensacion-de-excedentes>

¹⁷ <https://www.esios.ree.es/es>

the consumption of that hour over the whole of the day (they are the “shape” of your daily consumption curves). Average consumption data are also used according to the type of home and the average contracted power in Spain.

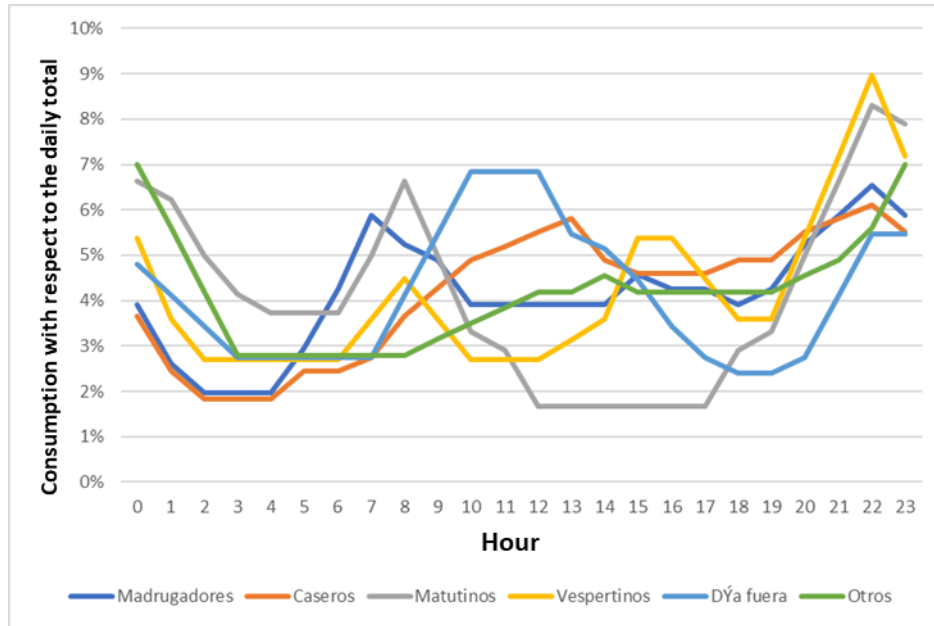


Image 18: Hourly electricity consumption curves of the different standard profiles

Average contracted power (kW)	4.5
Average daily household consumption (kWh)	10.28
Average daily consumption (kWh)	9.24

Table 14: Average values of contracted power and electricity consumption

According to the contracted power that the user enters, his estimated consumption profile will be, for each hour i :

$$Consumption\ hour_i = \frac{Relative\ weight\ standard\ profile_i \cdot Average\ consumption\ [kWh]}{\frac{Contracted\ potency\ [kW]}{Average\ power\ [kW]} \cdot 1000} \cdot \frac{Wh}{kWh}$$

Relative weight standard profile $_i$: Relative weight (as per one) of consumption for hour i with respect to the daily total, depending on the standard profile of the user.

Then the difference per hour between the estimated consumption and the predicted electricity production is obtained. For the hours in which more electricity is generated than is consumed, the electrical spillage to the grid (the difference between consumption and generation) is multiplied by the previously downloaded hourly prices for compensating surpluses. Thus, the compensation per hour is obtained (€).

4.5.3. Streamlit interface

Notebooks: Requisitos_streamlit.ipynb, Interfaz.ipynb, Streamlit_app_1.ipynb y Streamlit_app_2.ipynb

The user interface, the frontend of the project, has been developed as a Streamlit app. *Streamlit* is a library to create web applications in a simple way.

▪ Prerequisites

The **Requisitos_streamlit.ipynb** notebook should only be run once, to configure everything necessary to run the *Streamlit* app.

Folium will be used to display the interface map in Streamlit. However, the usual Streamlit is static. In order to extract data from it, the Streamlit component must be 'bidirectional'¹⁸. Therefore, in order to extract the coordinates of the point selected on the map by the user, a bidirectional Streamlit component must be created. A bidirectional Streamlit component has two parts:

- An interface, which is built from HTML and any other web technology you want (JavaScript, React, Vue, etc.), and is rendered in Streamlit applications through an `iframe` tag.
- A Python API, which Streamlit applications use to instantiate and communicate with that interface.

To facilitate the bidirectional component creation process, Streamlit has created a React template and a TypeScript-only template in the Streamlit Component Template GitHub repository¹⁹. They also provide some sample components in the same repository. However, it is difficult to add interactivity with the map and send the results to Streamlit.

This project is a use case that is most easily addressed by creating a custom Streamlit component on top of *Leaflet*. For the deployment, another GitHub repository is used that addresses this problem²⁰. Following the steps of the notebook, these repositories are cloned and parts of them are used. A Mapbox *access token* must also be entered in the code²¹. Finally, the app's Theme configuration is saved.

▪ Interface code

The notebook **Interfaz.ipynb** will be used for the general operation of the application. Its execution generates the `interface.py` file that defines the application. It is necessary to run it when it is indicated in the **Streamlit_app_1.ipynb** notebook. Its content adapts the **Script_funcional.ipynb** code to generate a Streamlit app, adding other functionalities:

- Generates a map that allows users to select their location. This data was set as an example value in **Script_funcional.ipynb**.
- Introduces a sidebar with interactive sliders that allow the user to enter the orientation and inclination of their solar panels. Also, if they are not sure of the exact value, it allows them to mark a range of values. If this option is selected, the mean value of the range is taken. They can also clear this range if they finally know the exact value. This data was set as an example value in **Script_funcional.ipynb**.

¹⁸ https://docs.streamlit.io/en/stable/develop_streamlit_components.html#create-a-bi-directional-component

¹⁹ <https://github.com/streamlit/component-template>

²⁰ <https://github.com/andfanilo/streamlit-light-leaflet>

²¹ <https://www.mapbox.com/>

- Allows to enter the user's peak power, or check a box that sets this value at 1 kW (unitary power). In this second case, the electrical production results must be multiplied by the actual peak power. This data was set as an example value in **Script_funcional.ipynb**.
 - In addition, it allows the user to check a box if he wishes to know an estimate of the financial compensation that he will receive for his surpluses. Here, it allows users to enter their contracted power, the type of consumption profile and the type of housing of their supply point. This data was set as an example value in **Script_funcional.ipynb**.
 - Executes the **Script_funcional.ipynb** code, showing the prediction of electrical production of the installation per hour, for the next day. If the user has also marked the option to know an estimate of his compensation for surpluses, his estimated consumption curve is superimposed on the prediction of electricity production. In this case, it is also shown the graph of the estimated compensation (€) that he will receive each hour.
 - It enables the user to download a csv with the data that generates these graphs.
- App execution

Once **Requisitos_streamlit.ipynb** has been executed, in order to run the *Streamlit* app locally, the following steps must be followed:

- The **Interfaz.ipynb** notebook must be run, so that the .py file that defines the interface exists.
- **Streamlit_app_1.ipynb** must be run. In order to use the application, the last cell must be kept running.
- When **this cell** shows: *Compiled successfully!*, wait **a couple of minutes** and execute **Streamlit_app_2.ipynb**. The last cells of both scripts must be kept running at the same time. Wait two minutes before running **Streamlit_app_2.ipynb**, with **Streamlit_app_1.ipynb** running. This avoids charging problems. If an error appears, it will be enough to refresh the page until it disappears, or repeat the process, waiting for more time between the execution of both notebooks.
- A tab will open in the browser with the *Streamlit* app.

▪ Streamlit App Guide

Here is a guide to using the Streamlit app. It consists of two parts, the main screen and the sidebar:



Image 19. Main screen of the Streamlit App

In the sidebar the user is allowed to mark the inclination and orientation of his panels. The option of indicating a range of values is also supported, in case the exact value is not known. The value taken in this case will be the mean. This range can be deleted, so that the value of the main bar is taken back.

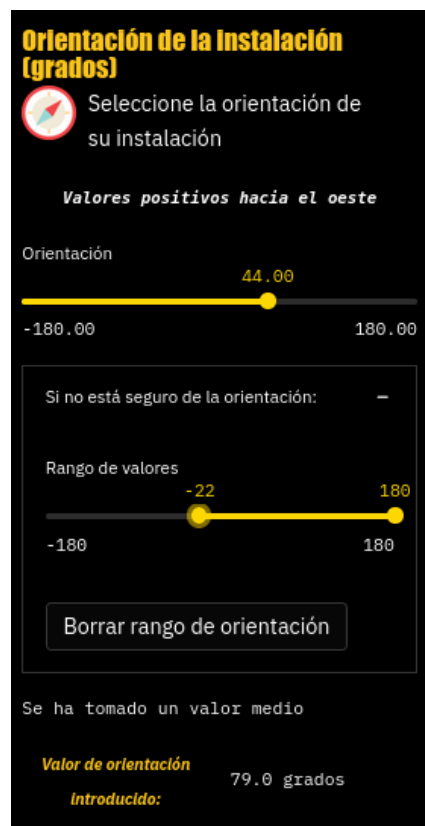


Image 20. Orientation selection in Streamlit

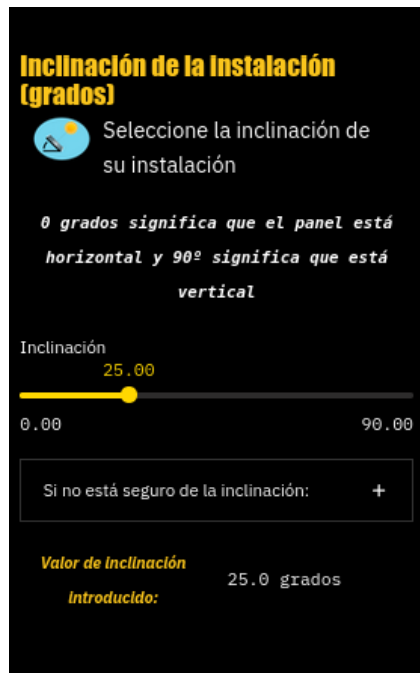


Image 21. Inclination selection in Streamlit

The main screen allows to:

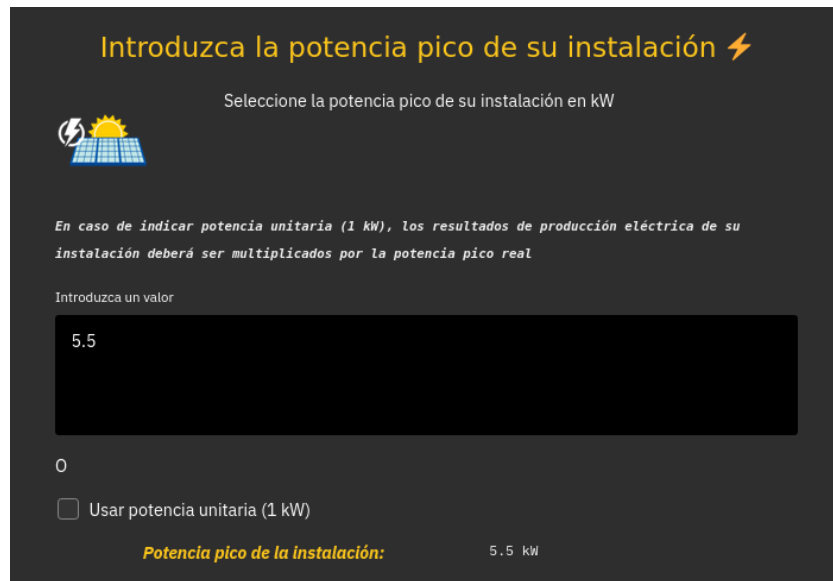
- Select the location of the installation (house with solar panels)



Image 22. Location selection in Streamlit

- Enter the peak power of this installation, in kW (it admits both a semicolon as a decimal separator). It also supports the option of checking a box and setting the power as 1 kW

(unit power). In this second case, the electrical production results must be multiplied by the actual peak power.



The screenshot shows a web interface with a dark background. At the top, the title "Introduzca la potencia pico de su instalación ⚡" is displayed in yellow. Below it, a subtitle "Seleccione la potencia pico de su instalación en kW" is shown. To the left of the subtitle is an icon of a sun and solar panels. A paragraph of text in Spanish explains that if unitary power (1 kW) is indicated, the results will be multiplied by the actual peak power. Below this, there is a text input field labeled "Introduzca un valor" containing the number "5.5". A "0" button is located below the input field. A checkbox labeled "Usar potencia unitaria (1 kW)" is present and unchecked. At the bottom, the text "Potencia pico de la instalación:" is followed by "5.5 kW".

Image 23. Entering peak power in Streamlit

- In addition, it allows the users to check a box if they want to know an estimate of the financial compensation they will receive for their surpluses. Here, it allows users to enter their contracted power, the type of consumption profile and the type of housing of their supply point.

Estimación de compensación por excedentes (€)

Para poder proporcionarle una estimación de la compensación económica que podrá recibir mañana por la energía que vierta a la red (aquella de la producida por sus paneles solares que no consuma) se necesita estimar el perfil de consumo de su vivienda

☒ Deseo recibir una estimación de mi compensación por excedentes

 Seleccione la potencia contratada de su hogar

Introduzca su potencia contratada

Introduzca valor de potencia válido. Ejemplo: 4.5 kW

 Seleccione el tipo de vivienda en la que se encuentra la instalación fotovoltaica

Seleccione el tipo de vivienda en el que se encuentra la instalación fotovoltaica:

Casa

 Seleccione el tipo de comportamiento de los habitantes de su vivienda

Para poder estimar el consumo de su vivienda, se va a realizar una predicción de este en base a los datos medios de la sociedad española. para ello, es necesario conocer a qué horas suele estar ocupada. Por ello, se requiere que indique el tipo de comportamiento típico que tiene el grupo de habitantes de la vivienda. Por ejemplo, si suelen irse de casa muy pronto por la mañana: Madrugadores. Si pasan las tardes en casa, porque trabajan/estudian por la mañana: Mañanas fuera...

Seleccione el horario que más identifique a la gente que habita en la vivienda:

Madrugadores

Image 24. Selection of consumption habits in Streamlit

- Finally, it allows to obtain the prediction of electrical production of the installation per hour, for the next day. If the user has also marked the option to know an estimate of his compensation for surpluses, his estimated consumption curve is superimposed on the prediction of electricity production. Also shown, in this case, is the graph of the estimated compensation (€) that he will receive each hour.

Predicción de la producción eléctrica

Guardar coordenadas y características

Image 25. Prediction of electricity production in Streamlit



Imagen 26. Estimated profile of photovoltaic consumption and production for the next day in Streamlit



Image 27. Estimated compensation of surpluses (€) for the next day in Streamlit

- It allows the user to download the csv files that generate these graphs.

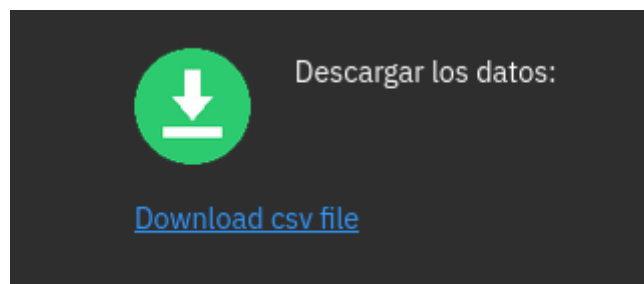


Image 28. Downloading data by the user in Streamlit

An example video of the use of the app is attached below:



<https://youtu.be/fr-S27TEngg>



5. Conclusions

It is concluded, therefore, that the project satisfactorily meets the objective set. A free and intuitive access tool has been developed where users can receive, by entering their installation data, a generation prediction that they will be able to use the next day. However, the accuracy of the prediction models is not perfect. This is due to the fact that a sufficiently broad data history has not been available (all available sources were expensive or non-hourly data), but data have been taken on a daily basis, to generate this historical information through its accumulation. Obviously, this has only happened for a few months, once the necessary notebooks and tools were finished and prepared. Therefore, it is considered that, if this process were maintained over time to have a larger database with which to train the models, the result would be much better.