Daily Market Price Forecast

Master Degree Final Project

Fernando Rodríguez Paler

Content

1. Summary

This Master's degree final project is a study about the forecast of the Electricity Market Price.

Electricity prices in Europe are set daily (every day of the year) at 12 noon, for the twenty-four hours of the following day, in what is referred to as the Daily Market. The price and volume of energy over a specific hour are determined by the point at which the supply and demand curves meet, according to the marginal pricing model adopted by the EU.

When the Daily Market Price is set, some of the factors that most affect are:

- Electricity demand: The total energy consumption for all purposes. The larger the electricity demand, the higher the Market Price will be.
- Renewable energy generation: Renewable energy have priority in the power grid and therefore always come first in the merit order. The larger of renewable energy generation, the lower the Market Price will be.
- Date and time: The Market Price will be increased in summer and winter, in addition peak values usually take place in the early morning and late afternoon.

1.1 Objective

The objective of this repository is to develop a Machine Learning model that allows to predict the Daily Market Price for 48 hours Forecast Horizon.

For this, it will be necessary to obtain the information considered relevant for the development of the model.

Once this information is available, it will be necessary to pre-process it, with the objective of adapting it to the requirements of the model.

Next, a Machine Learning model based on the Random Forest Regressor algorithm will be developed.

Finally, the project will have a visualization chapter developed with Tableu with which the results of the developed model can be seen graphically.

2. Data acquisition

To develop this model, two sources were mainly used.

2.1. Bank Holidays

On the one hand, is necessary information about the holidays that takes place in Spain, for this is used a file called *“Bank_Holidays.csv”* that contains the information of the existing holidays in the last 4 years (since January 1, 2014 from December 31, 2017).

The file *"Bank_Holidays.csv"* can be found in *"/DailyMarketPriceForecast/Bank_Holidays/Data/"* and consists of the following variables:

Variable	Description
Date	*Date in format dd/mm/yyyy*
Day	*Weekday (Monday to Sunday)*
Title	*Name of the bank holiday*
Bank Holiday	*True (1) or False (0)*

The complete dataset is made up of 4 variables and 55 samples, one for each bank holiday that took place during that period.

2.2. ESIOS

On the other hand, to access the information of REE, it is done through an API REST service to ESIOS (System Operator Information System). By using this service, you can download all the information in the system.

To access the system, it is necessary to request a token by email.

The indicators available in ESIOS can be found in the Excel file called *"indicators.xlsx"*. This file consists of two fields:

Variable	Description
Name	*Name of the indicator as it appears on the ESIOS website*
Indicator	*Number associated with the name of the indicator used to download the indicator information*

As seen in the previous section, the parameters that have most influence in the Daily Market Price are:

Indicator name	Indicator number
Previsión diaria de la demanda eléctrica peninsular	*Id: 460*
Previsión de la producción eólica nacional peninsular	*Id: 541*
Generación prevista Solar	*Id: 542*
Precio medio horario componente mercado diario	*Id: 805*

In this way, information can be downloaded through the REST API service. Since the Electric Market Price is a value that is updated hourly, to develop the model it's necessary that the rest of the parameters maintain the same scale, so the script is configured to perform the data download of the indicators previously mentioned with hourly frequency.

To obtain the information from ESIOS, it is necessary to execute the Jupyter Notebook file *"Get_Indicators.ipynb"*, which is available in *"/DailyMarketPriceForecast/ESIOS/"*. Once the file is executed, four different *".csv"* files will be generated (one for each indicator) in the path *"/DailyMarketPriceForecast/ESIOS/Data/"*. The files obtained have a similar structure, formed by two fields:

Indicator date	Indicator value
Previsión diaria de la demanda eléctrica peninsular	*Id: 460*
Previsión de la producción eólica nacional peninsular	*Id: 541*

The complete dataset is made up of 2 variables and 35065 samples.

3. Preprocessing

It is necessary to make some transformations in the datasets in order to implement the desired Machine Learning model.

As the dataset related to bank holidays has daily frequency, the information must be converted to hourly frequency in order to make a merge with the rest of the datasets. In addition, four series related to the dates ('Hora', 'Dia', 'Mes', and 'Dia semana') will be created to analyze the contribution of these variables to the model.

On the other hand, the ESIOS information comes from different files, so it must be combined in a single DataFrame whose variables will be the date in hourly format and also each of the indicators used in ESIOS.

Once the preprocessing of the data is finished, it must be analyzed if there are outliers in the dataset. It is observed that there are only outliers in two series of the complete DataFrame (0.97% and 4.45% with respect to the total number of samples in the series, respectively).

A Machine Learning model based on the Random Forest algorithm will be developed, it should not be forgotten that this model is very robust in the presence of outliers because the model isolates them in small regions of the feature space. Then, since the prediction for each leaf is the average (in the case of regression), being isolated in separate leaves, outliers won't influence the rest of the predictions (in the case of regression for instance, they would not impact the mean of the other leaves).

Despite the presence of a small number of outliers, it is decided not to act on them because it is considered that these samples have a normal behavior within the dataset.

4. Modelling

For the development of the model, a Random Forest Regressor algorithm has been implemented.

The Random Forest algorithm is an ensemble of Decision Trees, trained with the "bagging" method, which means that the process of finding the root node and splitting the feature nodes will run randomly.

The Random Forest algorithm is considered one of the most effective machine learning models for predictive. Some of it most important features are:

- It runs efficiently on large data bases.
- It can handle tabular data with numerical features, or categorical features with fewer than hundreds of categories.
- It gives estimates of what variables are important.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

To train the model, the dataset has been divided into 80% train and 20% test.

4.1. Hyperparameter tuning

Once the model is decided, before training the model it is necessary to obtain the best parameters of the Random Forest Regressor model with a hyperparameter tuning.

The Randomized Search method has been chosen. With these method, the parameters of the estimator are optimized by cross-validated search over parameter settings. In contrast to GridSearchCV, not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions.

4.2. Model evaluation

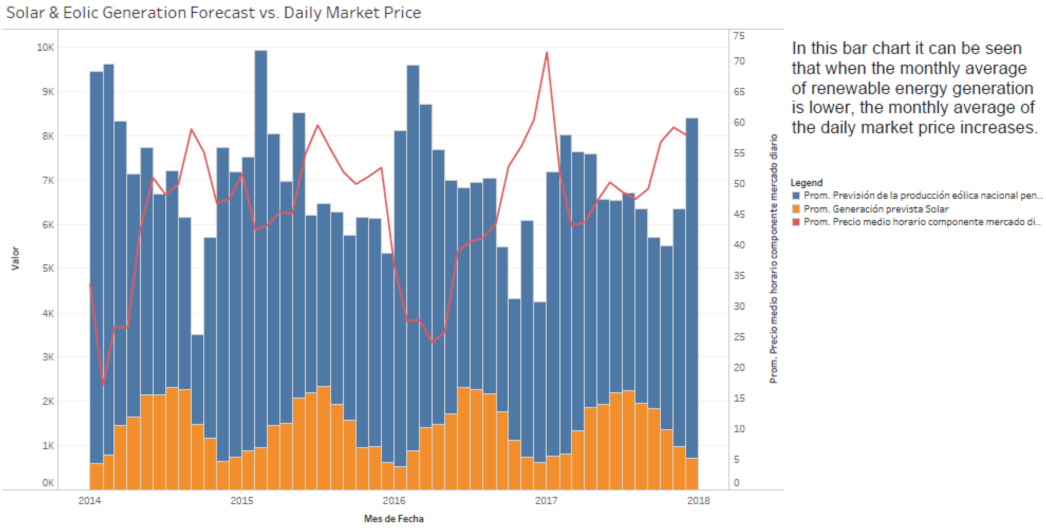
Once the hyperparameter tuning is done, the model is trained with the best parameters of the Random Forest Regressor, and it is tested with the test dataset (20% of total samples) to obtain the results of the prediction.

5. Visualization

A visualization module has been developed with Tableau software, through which numerous types of interactive graphics can be implemented.

In order to develop Tableau Workbooks, a data source must be imported. In this case, two different Tableau Workbooks have been developed.

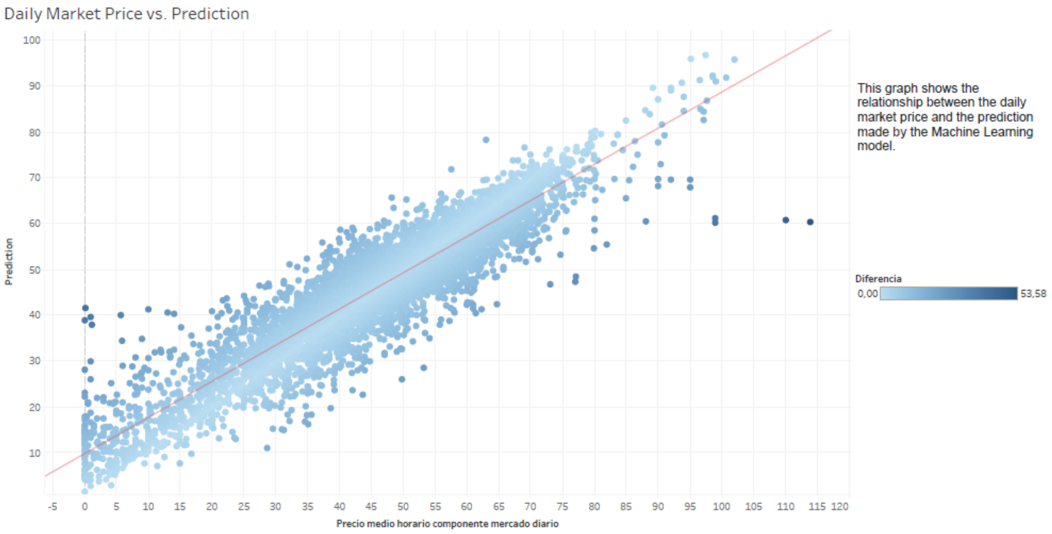
For the first one, the input data of the Machine Learning model *"DataFrame.csv"* have been used to see interactively how the Daily Market Price is affected with the different model variables:



Solar & Eolic Generation Forecast vs. Daily Market Price

In this bar chart it can be seen that when the monthly average of renewable energy generation is lower, the monthly average of the daily market price increases.

For the second Tableau Workbook, the results of the Machine Learning model *"Prediction_Results.csv"* have been used to observe the relationship between the test data and the prediction:



Daily Market Price vs. Prediction

This graph shows the relationship between the daily market price and the prediction made by the Machine Learning model.

6. How to Run

1. Processing Bank Holidays data

Execute "Bank_Holidays.ipynb".

```
$ jupyter notebook "./Bank_Holidays/Bank_Holidays.ipynb"
```

2. Getting ESIOS data

Execute "Get_Indicators.ipynb".

```
$ jupyter notebook "./ESIOS/Get_Indicators.ipynb"
```

3. Preproccesing and Modelling

Execute "Preprocessing_and_Modelling.ipynb".

```
$ jupyter notebook "./Model/Preprocessing_and_Modelling.ipynb"
```

4. Visualization

Execute "Visualization1.twbx" and "Visualization2.twbx".

7. About the technologies and libraries

The technologies used to develop this project are:

- *Python*: Programming language used for Preprocessing, Data Analysis and Modelling. During the project, two applications have been used to work with Python:
    - *Jupyter Notebook*: Application used for Preprocessing and Modelling.
    - *Spyder IDE*: Application mainly used in ESIOS module to import the libraries that allow extracting data from ESIOS API REST.
- *Tableau*: Software that produces interactive data visualization, used in the Data Visualitation module.

The main libraries used in this project are:

- *functools*: The functools module is for higher-order functions. Functions that act on or return other functions.
- *matplotlib*: Plotting library which produces publication quality figures.
- *numpy*: NumPy is the fundamental package for scientific computing with Python that provides a multidimensional array object, various derived objects, and an assortment of routines for fast operations on arrays.
- *pandas*: Pandas is an easy-to-use data structures and data analysis tools for the Python programming language.
- *seaborn*: Visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.
- *sklearn*: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It provides efficient tools for data mining and data analysis, and is built on NumPy, SciPy, and matplotlib.

8. About the author

https://www.linkedin.com/in/fernandorodriguezpaler