Microgrids

Forecasting assignment

Jonathan Dumas
jdumas@uliege.be

Forecasting assignment

Learning objectives

Through this assignment, it is aimed for the students to be able to:

- Produce **point** forecasts;
- Produce **probabilistic** (quantile) forecasts;
- Perform **verification** of point & probabilistic forecasts

Forecasting assignment

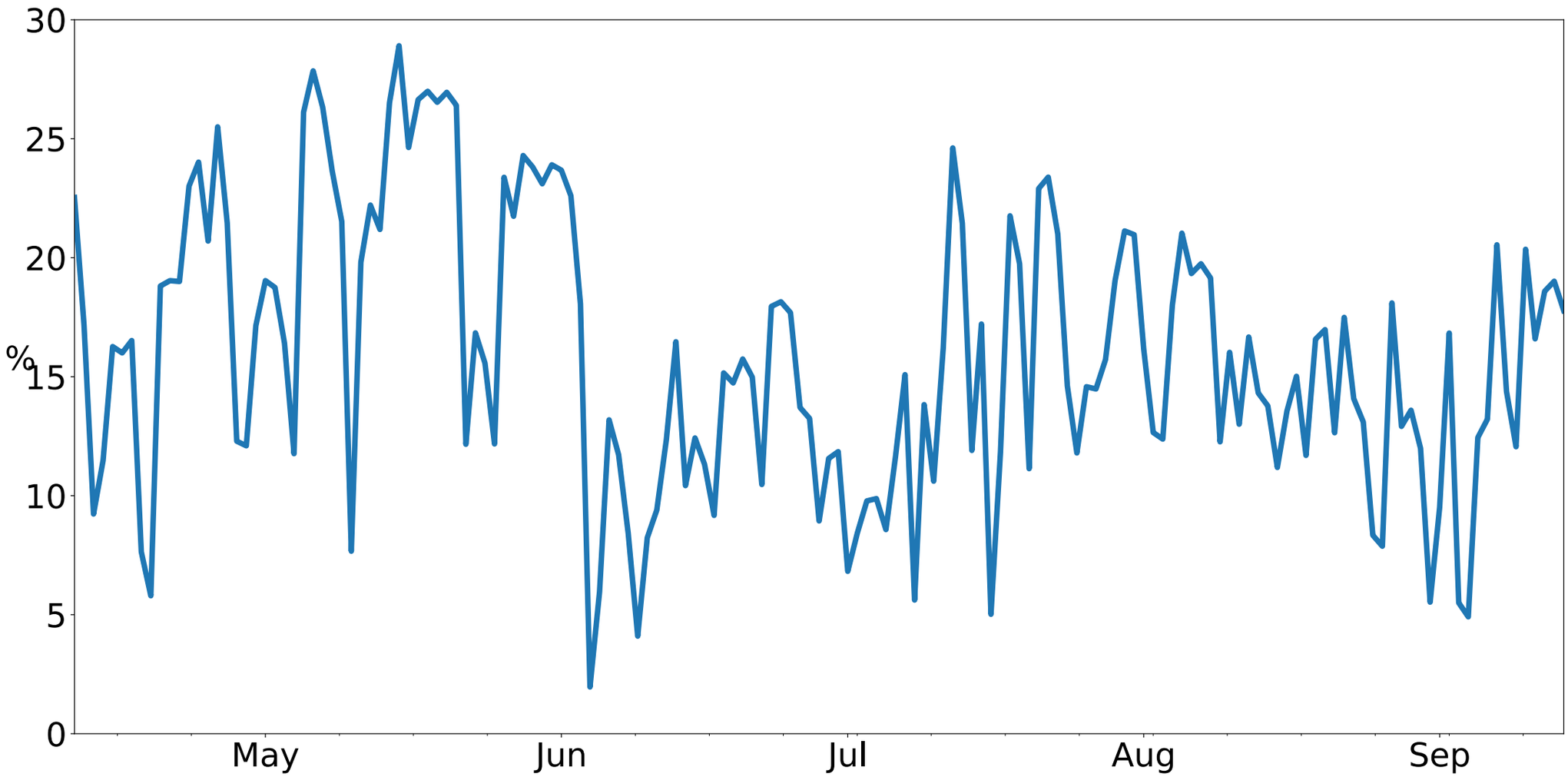
Case study: PV parking rooftops from Liège university

PV installation of 466.4 kWp

https://www.uliege.be/cms/c_7726266/fr/2500-m-de-panneaux-photovoltaiques-bientot-en-fonction-sur-le-campus-du-sart-tilman

Forecasting assignment

Daily energy per day of the dataset



Daily energy PV generation normalized by the daily energy produced by the total installed capacity (466.4 * 24 kWh).

Forecasting assignment

Dataset inspection

Plot the PV generation observations.

Plot the weather forecasts: irradiance and air temperature.

Use the file « data_inspection.py ».

Forecasting assignment

Point forecasts

1. Implement a **persistent** model to be used as a benchmark: D-1 = D in the file « persistance_model_TODO.py ».
2. Implement a **linear regression** model from the Python scikit-learn library in the file « MLR_point_TODO.py ».
3. Implement a **Gradient Boosting Regressor** (GBR) from the Python scikit-learn library in the file « GBR_point_TODO.py ».
4. Try to optimize the **GBR hyper-parameters**.
5. Perform the **visual inspection** of point forecasts, and compute scores. Comment the results. You can use « score_comparison.py ».
6. Change the random parameter to **build a new pair of learning and testing sets**. How do the scores behave? Comment the results. *WARNING: at this stage, you cannot change the GBR hyper-parameters selected at point 4.*
7. Discuss the validation strategy. Would it be possible to adopt another strategy? What would be the pros and cons?

Forecasting assignment

Quantile forecasts

1. Implement a **Gradient Boosting Regressor** (GBR) from the Python scikit-learn library and change the loss function to produce quantiles in the file « GBR_quantile_TODO.py ».
2. Try to optimize the **GBR hyper-parameters**.
3. Perform the **visual inspection** of probabilistic forecasts, and compute scores. Comment the results. You can use « score_comparison.py ».
4. Change the random parameter to **build a new pair of learning and testing sets**. How do the scores behave? Comment the results. *WARNING: at this stage, you cannot change the GBR hyper-parameters selected at point 2.*

Forecasting assignment

Rules for assignment completion and submission

1. Use the groups created for the precedent assignments.

2. When submitting your report, please indicate on **the report title page** the names of the group students.

3. Write a **short report** (max 4 pages, 11pt font).

4. Submit your code and your report on Ecampus.