Energy Data Science - Report 1

Kacper Aleksander

26th of October, 2024

Contents

1 Introduction

This report is about analyzing energy production data collected from some PV panel near Tallinn. Such data is important to get a deeper understanding of supply of solar energy. By looking for patterns, and correlations with other variables, we can predict solar generation in the future. From my point of view, this might be some very useful information for the Demand Response system, in which consumers of energy match their demand with current energy supply.

2 Data collection

Provided data consists of the following features: raw, temperature, modified1, modified2, modified3. It might be worth considering to add other weather features from some external source, for example cloudiness, which intuitively seems important for solar panels. In this report, however, I focused only on the provided data.

First, I checked if there are any missing values.

```
raw               0
temperature       0
modified1      1191
modified2      1375
modified3      1137
```

There were missing data in the "modified" columns. After that, I used a pairplot to see the correlations between all of the features. Hence, I noticed an unusual correlation between "temperature" and "modified3", that I explore in the next section (Figure 4).

The correlation matrix (Figure 1) shows, that if a modified value is not missing, it is always the same as the raw temperature data. In real life, this might mean that there were 3 data-collecting sensors (corresponding to modified1, modified2, and modified3), and that the raw energy production values came directly from those sensors. So that whenever there was missing data from one of the sensors, the raw feature could be filled with any other sensor.

Based on this knowledge, I deleted the modified1, modified2, and modified3 features in the following sections, as they were essentialy incomplete duplicates of raw data. I handled the missing data just for the purpose of this report.

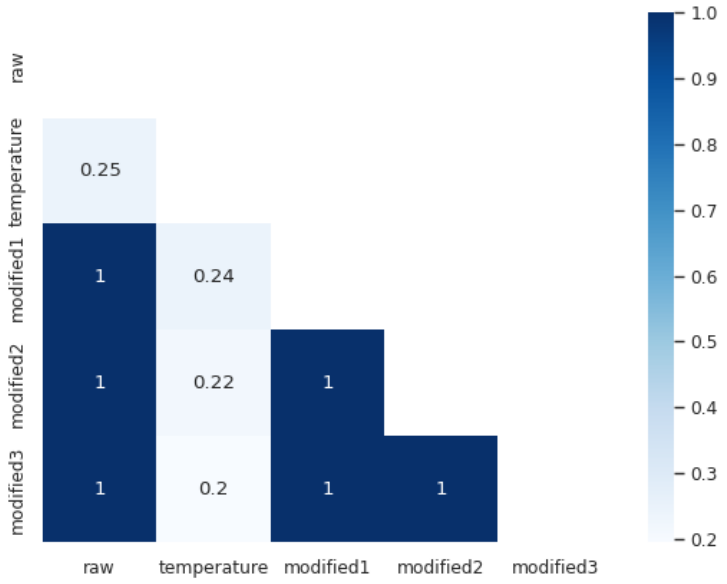

Figure 1: Correlation matrix

3 Data cleaning

3.1 Understanding missing data types

I plotted each of the missing features against temperature, to see if there is any pattern (Figures 2, 3 and 4).

There didn't seem to be any particular pattern of missing values of the features modified1 and modified2. Thus, I assumed that those missing values are classified as missing completely at random (MCAR), as they aren't influenced by any external variable or the data itself.

It is clear that when it comes to modified3, there is some maximum temperature that sets the limit for recording data over that temperature. This temperature is 21°C. Because temperature is an external factor to energy production, those values are classified as missing at random (MAR).

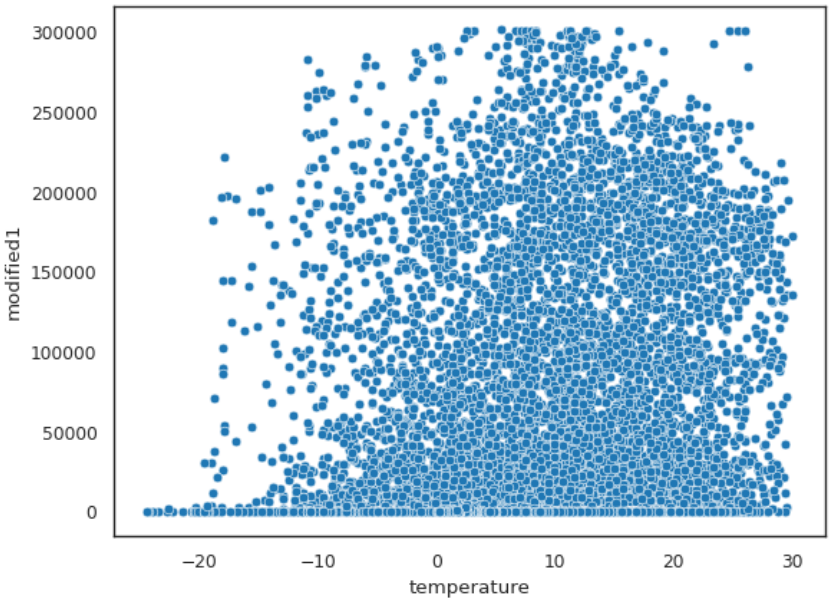

Figure 2: Scatterplot of the temperature and modified1 features

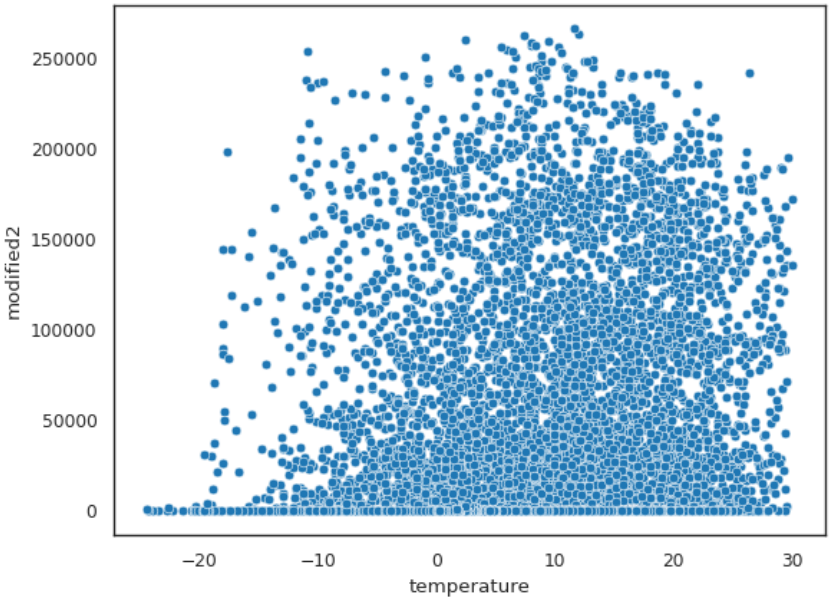

Figure 3: Scatterplot of the temperature and modified2 features

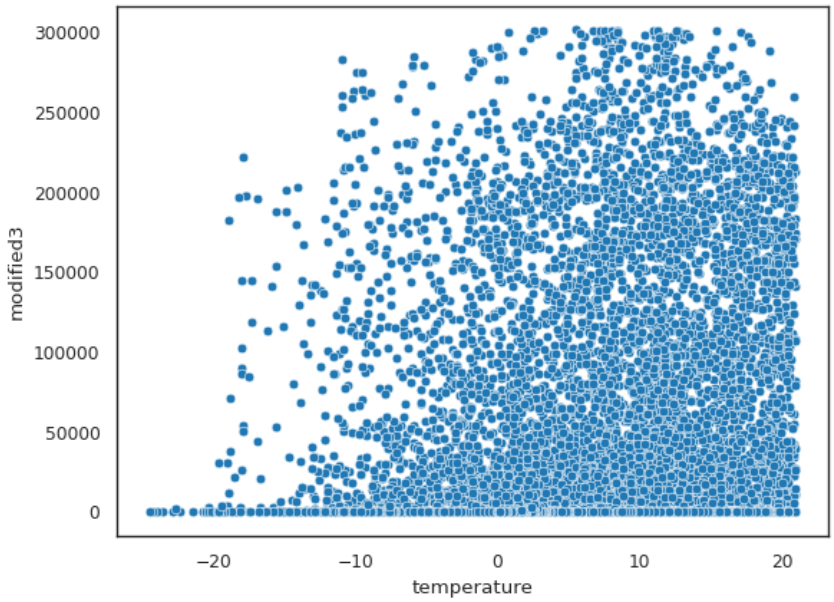

Figure 4: Scatterplot of the temperature and modified3 features

3.2 Dealing with missing values in modified1

I performed deletion, univariate imputation, and multivariate imputation on the "modified1" feature. The reseults are the following:

```
Deletion
Standard error: 129.70

Univariate imputation
Mean absolute percentage error: 3.315%
Root mean squared error: 21323
Standard error: 977.60

Multivariate imputation
Mean absolute percentage error: 0.005%
Root mean squared error: 10
Standard error: 0.12
```

Based on the chosen metrics, multivariate imputation seemed to be the most effective in that case.

4 Data exploration and feature engineering

First, I plotted the raw feature over time, and noticed a yearly seasonality (Figure 5). Then, I did the same for the temperature, which also seemed to have a yearly seasonal component (Figure 6). There were some unusual spikes in the temperature that deserved my attention.

I have added the following features, to see more clearly how does date and time correlate with temperature and energy production data:

- day (1 to 31)
- month (1 to 12)
- dayofweek (1 to 7, Monday being 1)
- is_weekend (True or False, True for Saturday and Sunday)

After exploring those correlations, I noticed that there is some strong bias for the temperature data in the days 1-12 each month (Figure 7). I fixed it by taking a mean for the correct-looking part (days 13-31). Then, I subtracted that mean from the mean for each of the flawed days (1-12), which gave me some kind of "error". And finally, I subtracted this error from each temperature data for the days 1-12, which brought the mean temperature of each day to around the same level (Figure 8).

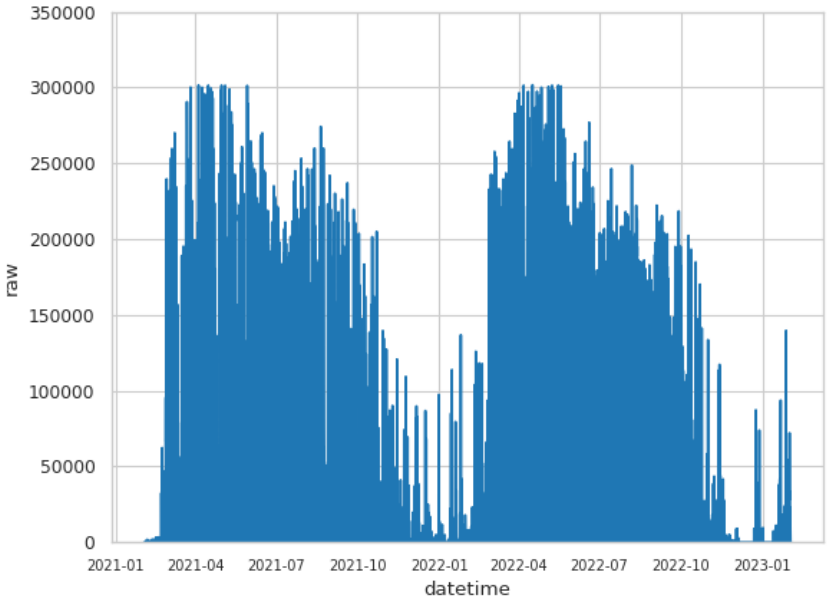

Figure 5: The raw feature as a time series

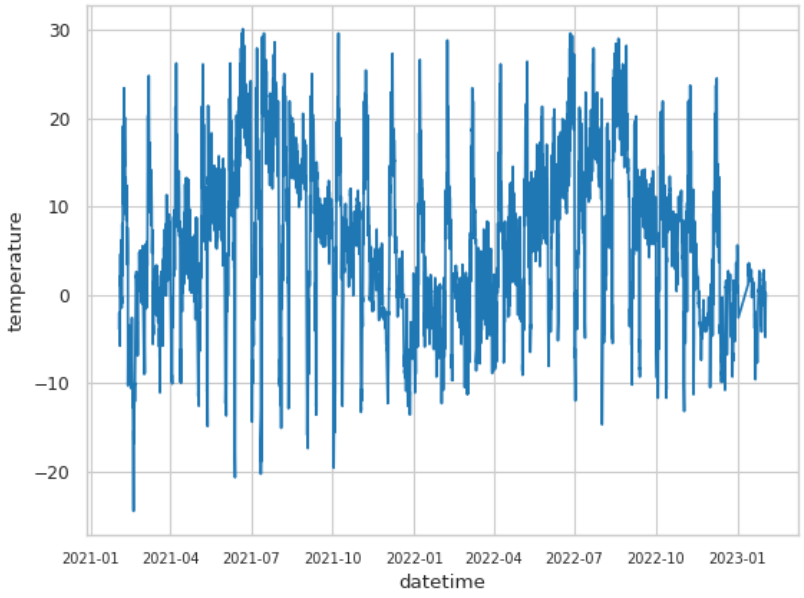

Figure 6: The temperature feature as a time series

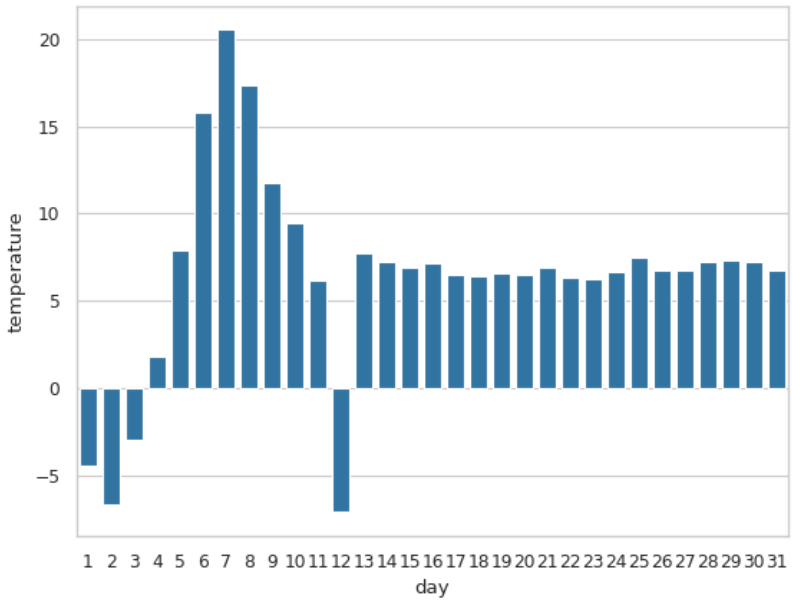

Figure 7: Mean temperature for each day of the month - before transformations

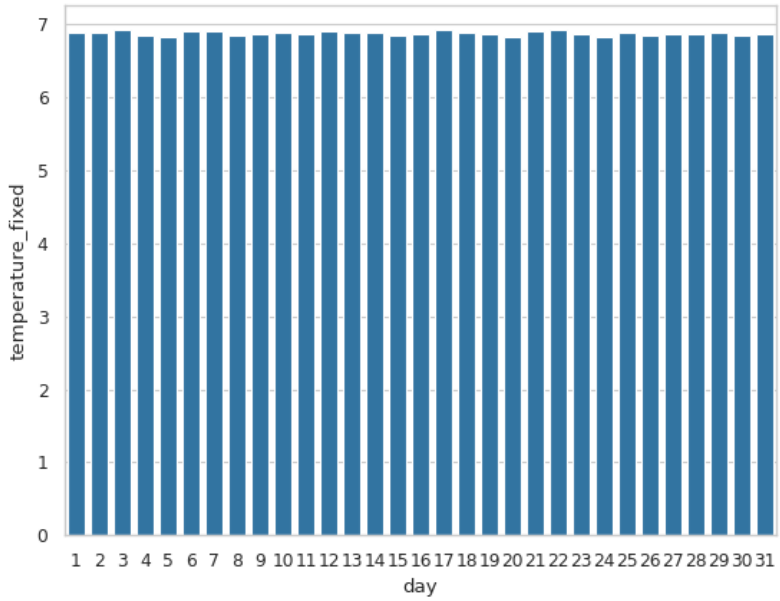

Figure 8: Mean temperature for each day of the month - after transformations

5 Data distribution and normalization

The raw feature follows a right-skewed exponential distribution, with most of the values being equal to 0 (Figure 9). This is why it is hard to change this distribution to mimic the bell curve. I took the square root of the raw data and standardized it afterwards (Figure 10).

The temperature distribution, on the other hand, closely resembled the bell curve, from the start (Figure 11). I have performed only standardization (12).

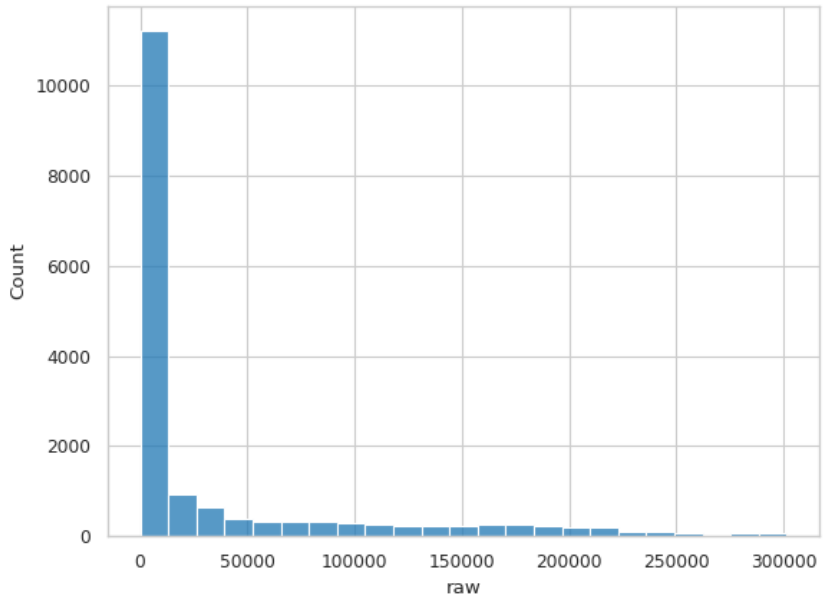

Figure 9: Raw histogram - before transformations

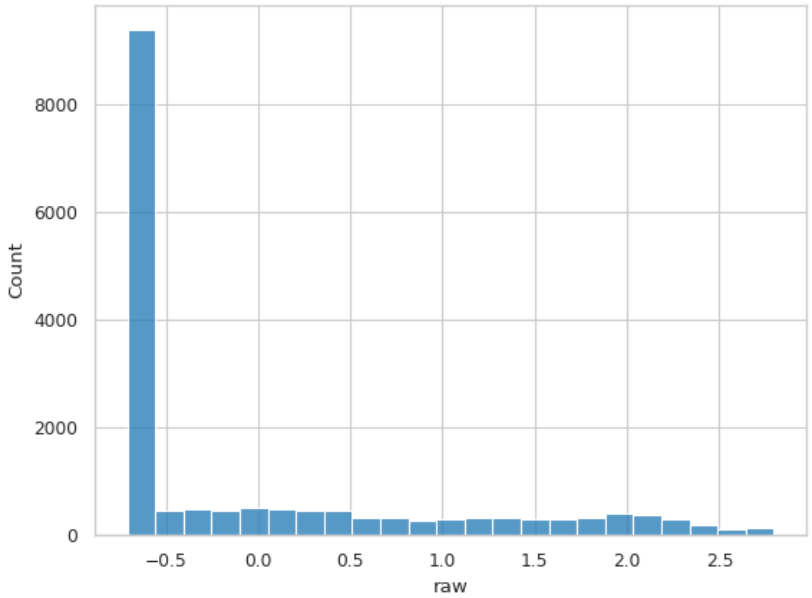

Figure 10: Raw histogram - after transformations

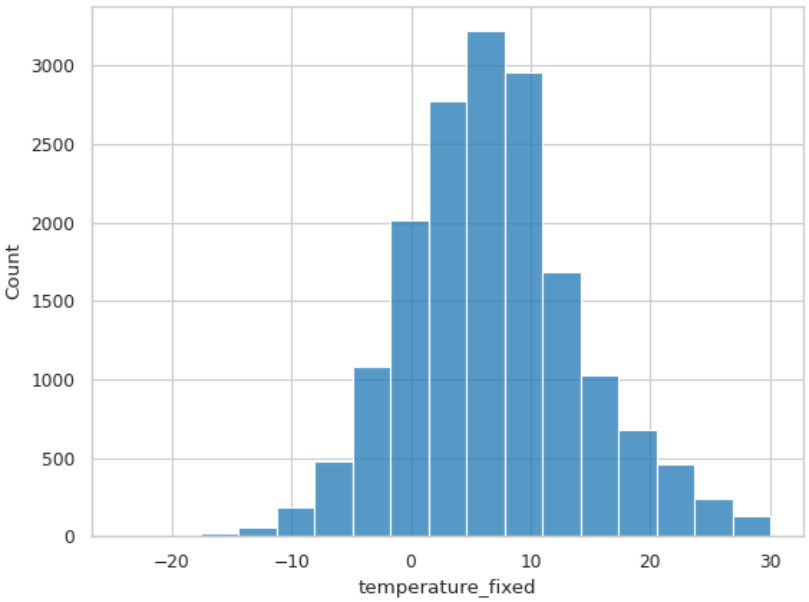

Figure 11: Temperature histogram - before transformations

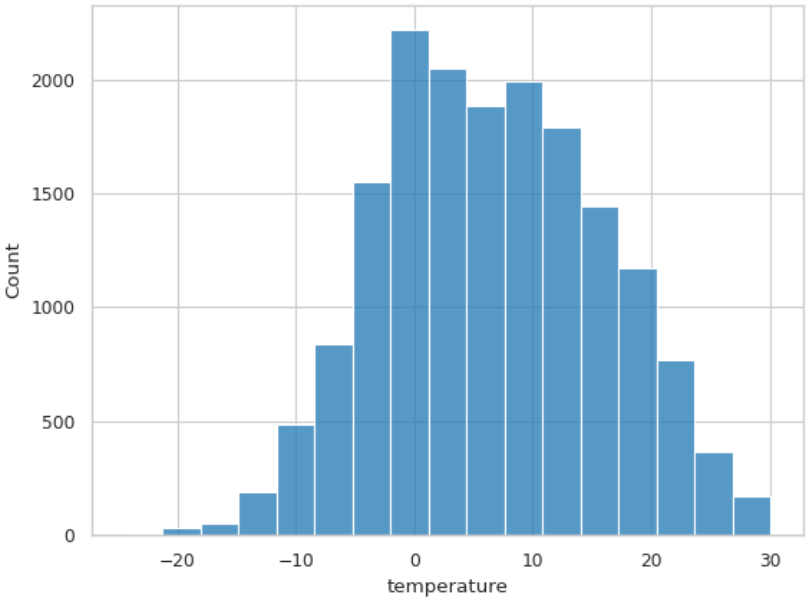

Figure 12: Temperature histogram - after transformations

6 Additive classical decomposition

Classical decomposition is about decomposing a timeseries into the trend, seasonal and residual components. Additive classical decomposition assumes the following:

$$y_t = T_t + S_t + R_t$$

where:
y_t - original series
T_t - trend component
S_t - seasonal component
R_t - remainder

For the data I was working with, there seemed to be two seasonalities:

- yearly (production is higher during summer)
- daily (production is higher during sunlight)

The trend, if it could be called so, seemed to be flat (no relevant change from one year to the other).

First, I created a centered, yearly moving average to see the yearly trend, which proved to be flat, as expected (Figure 13). Then, I subtracted this trend from the original raw data, to get data without the trend, consisting of yearly and daily seasonality (Figure 14). After that, I created a centered, daily moving average, to remove the daily seasonality. The daily moving average resembled the yearly seasonality (Figure 15). I subtracted the daily moving average from the yearly-de-trended data and was left with what was supposed to be only the daily seasonality. It, however, still looked like it had a yearly seasonal component (Figure 16).

My guess is that it might be connected to the amplitude, and a multiplicative classical decomposition might have worked better in this case. This is the last point where I came doing the work for this report, as I didn't have a clear idea on how to proceed further.

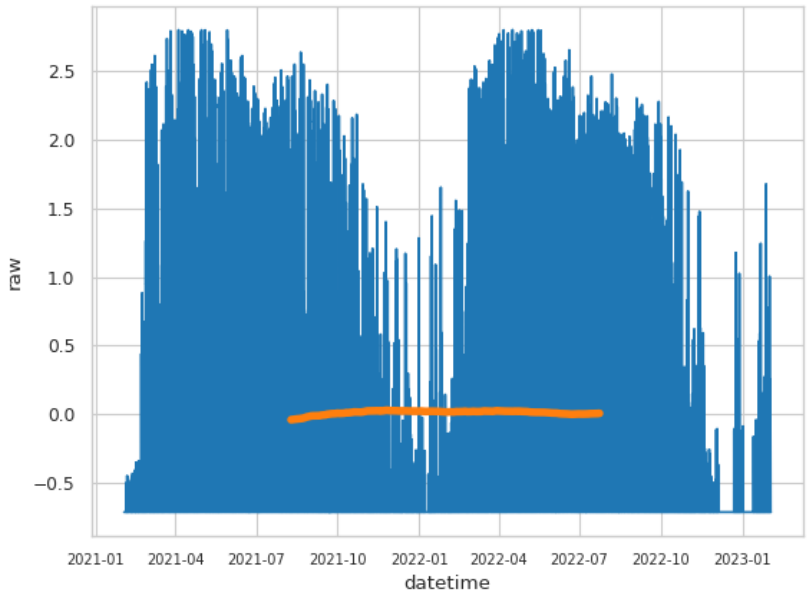

Figure 13: Yearly centered moving average on raw energy production data

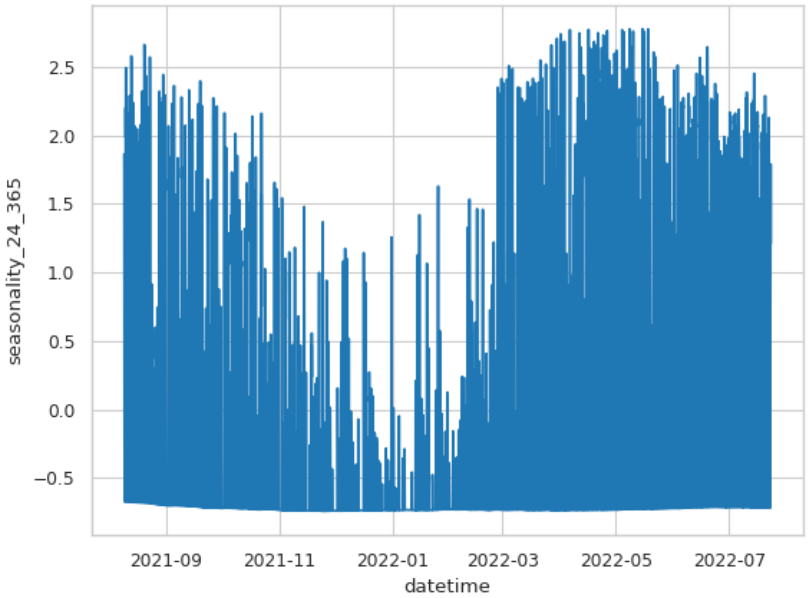

Figure 14: De-trended energy production data

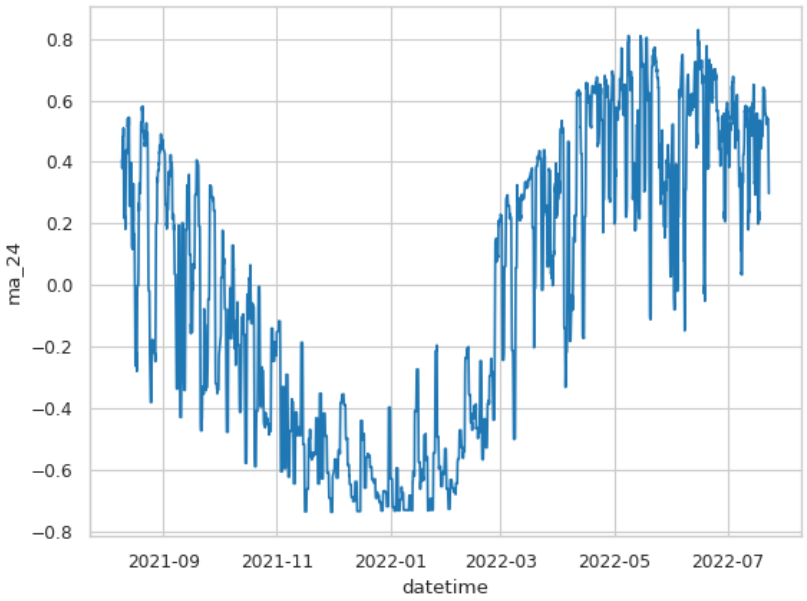

Figure 15: Daily centered moving average

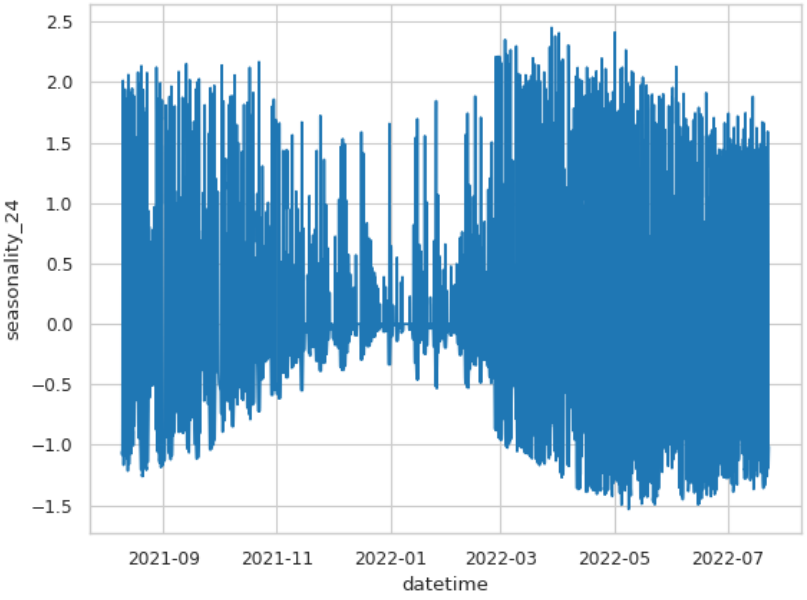

Figure 16: Daily seasonality component (?)

7 Appendix

I presented only selected charts from the work I did. The whole analysis process with the code is available at github.com/kacperfin/energy-data-science.