Analysis and Forecasting of Electricity Prices in Italy

Classical Econometrics vs Automated Procedures

Mirco Balduini

balduini.mirco@gmail.com

Dylan Tartarini[1]

tartarinidylan@gmail.com

May 10, 2021

[1]Special thanks to Irene Bardi for some useful data manipulation tricks

Overview

1. Introduction: why is it important to study electricity prices?
2. Exploratory Data Analysis (EDA)
3. Proposed approaches and models
   - sARIMA
   - Prophet
   - Sliding Windows
4. Conclusions
5. References

Section 1

Introduction: why is it important to study electricity prices?

Introduction 1/2

This is a brief presentation regarding our study on electricity prices from the Italian market using the software R. Our goal is to compare approaches for modelling and forecasting high frequency data in a **semi-automatic way** [2].
We dispose of the hourly time series of the **Unique National Price (PUN)** starting from Jan, 1st, 2008 up to Dec, 31st, 2019. PUN is the reference price detected on Italian Power Exchange (IPEX): data are free to use and can be found on the webpage of the Italian Regulator for the Energy Markets (GME)[3].

[2] Please note that the analysis is not financed by any Institution, we are just learning-by-doing the job of the applied economist.
[3] `https://www.mercatoelettrico.org/it/`.

Introduction 2/2

We first explain why it is important to study the behaviour of electricity prices in a country like Italy; next, we show a brief exploratory data analysis on the univariate time series. Then, we try to model and forecast electricity prices, both in the short and in the long run, using:

- a classical approach from Econometrics Literature, the **Seasonal-ARIMA** model, applied with the helpful *auto.arima*() function;
- an open-source algorythm developed by Facebook's employees, called **Prophet** and freely available in Python and R.

Then, we compare the results of following those two semi-automatic approaches and wrap up our conclusions and considerations on the subject.

Relevance of studying electricity prices

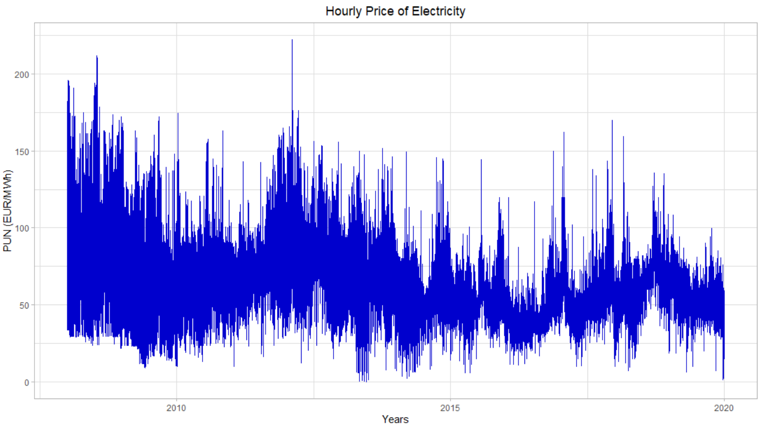
1. **PUN isn't quite a financial time series**:
   Since the *liberalization* [4]of the industry, electricity is exchanged in competitive markets, as occurs with other commodities: it presents, however, some characteristics which make it different, since **it cannot be stored** and **demand needs to be covered immediately**.
2. **The importance of forecasting**:
   These features of electricity price are responsible for its highly volatile behavior and the difficulty of price forecasting. **Modelling** this very peculiar time series is complex but nevertheless necessary, because forecasting has many implications both from the engineering, economic and financial points of view, and affects strategic industries in the economy of any country. In the context of liberalized markets, **short and long run forecasting** are both of great interest, and our analysis aims to contribute to the subject from the Italian perspective.

[4]Decreto legislativo march 16ht, 1999, n. 79 ("Decreto Bersani")

Section 2

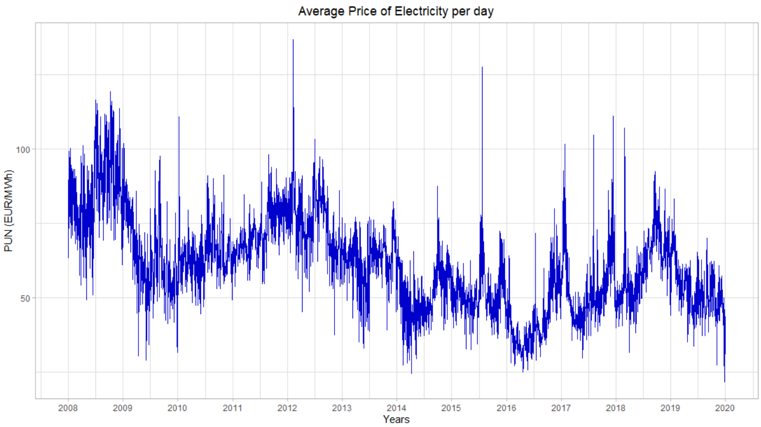
Exploratory Data Analysis (EDA)

EDA: The Dataset at a glance



Our univariate time series includes hourly prices for the price of electricity, recorded by the GME from the MGP Market (Previous Day Market), for 12 years, which means we are dealing with a full dataset of 105.145 observations.

EDA: Daily Observations



For long-run modelling, it will be useful to have daily data: we will decrease the frequency in our data and give up on some descriptive power in exchange for improved forecasting capabilities.

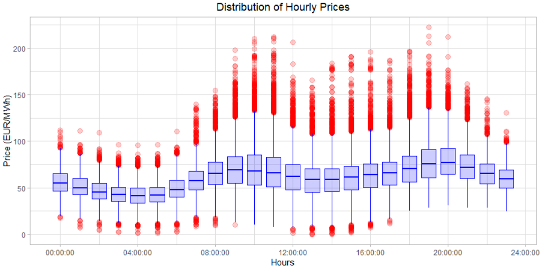
EDA: Complex Seasonality

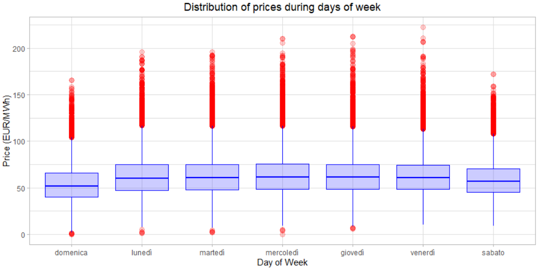
Elctricity prices display complex seasonality because of various factors, including **power generation**, which is influenced by external factors like **weather**, as well as **production cycles** of industries and, of course, the presence of **national holidays** and **weekends**.
We signal the presence of periodicity:

- intra-day (day and night cycle);
- across the same week (prices during weekends are usually lower than during weekdays);
- across months (yearly seasonality).

EDA: Distribution of Prices Across Hours and Days of Week

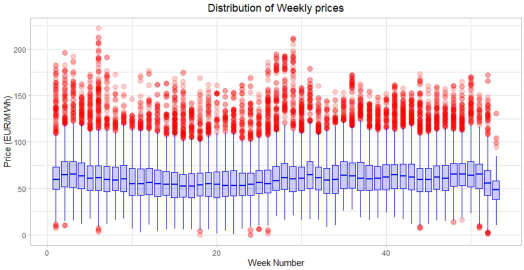
Electricity prices tend to have higher mean and volatility during daytime. They show a similar pattern when comparing weekdays and weekends. This is coherent with the way electricity is allocated in the Italian market.

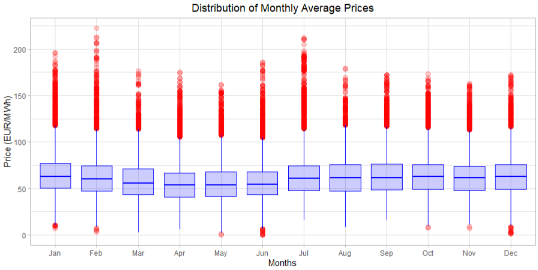




EDA: Distribution of Prices Across the Year

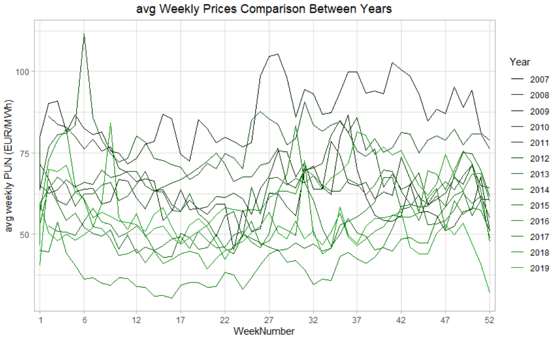
Electricity prices are higher and more volatile in midwinter and summertime, when the demand for electricity is higher, influenced by habits (the use of air-conditioning i.e.) and production cycles.

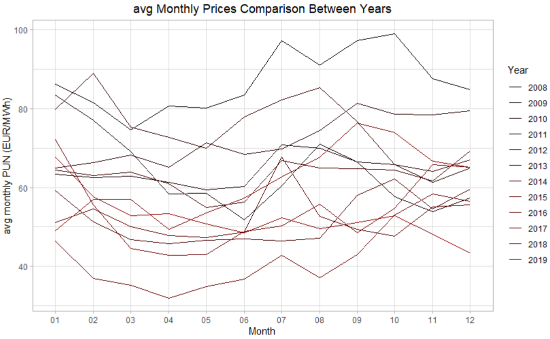




EDA: Visualizing Seasonality Across Years

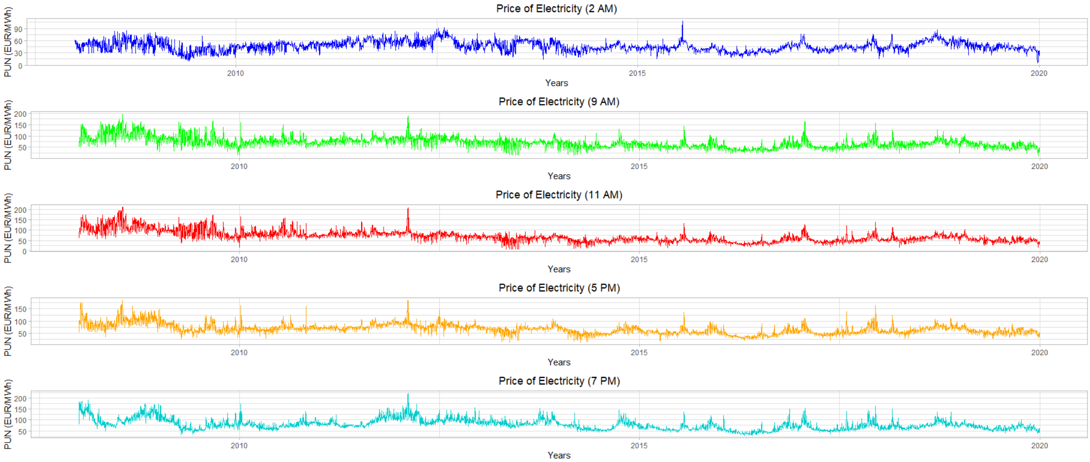
Prices across different years display similar peaks, twist and turns among the same weeks (and of course, the same months), hinting that the various types of seasonality showed with box-plots distributions combine themselves into a complex, year-wide pattern:





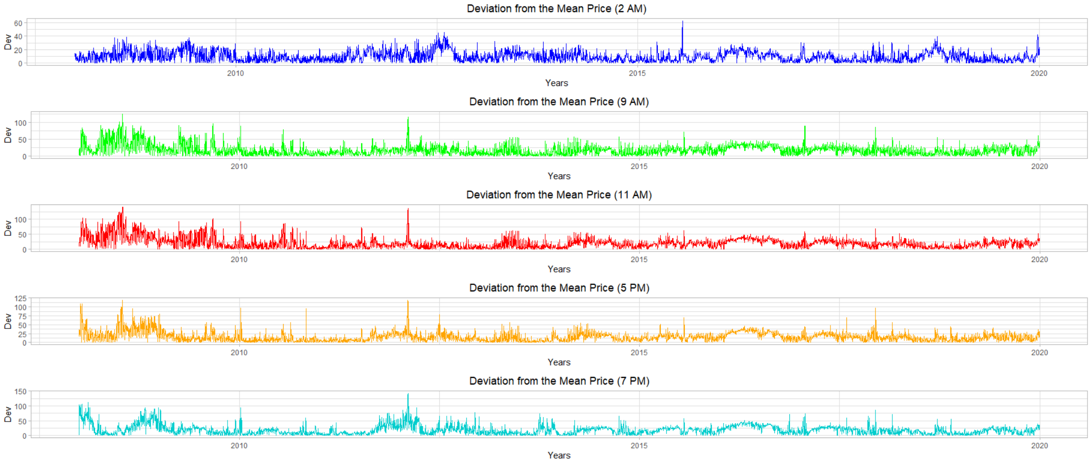
EDA: Conditional Mean Across Hours

We can also observe that mean price along years show the same behaviour across different hours of the day. Here, we present this feature displaying five selected hours:



EDA: Conditional Variance Across Hours

The same conditional structure can also be displayed for the deviation from the mean of hourly pricess along years:



EDA: Stationarity Check (Hourly Data)

We test whether or not we are working with a stationary time series. We run three tests on the hourly time series of electricity prices: *Adjusted Dickey-Fuller test* (ADF) with 47 lags, *Phillips-Perron test* (PP) with short lags and *KPSS test* with short lags:

Test	**H0**	**Stat-Test**	**p-value**	**Outcome**
ADF (47 lags)	Unit Root	-15.141	< 0.01	Stationary
PP (short lags)	Unit Root	-5761.1	< 0.01	Stationary
KPSS (short lags)	No Unit Root	145.3911	< 0.01	Not Stationary

Table: Unit Root tests results on the full hourly dataset

EDA: Stationarity Check (Daily Data)

As previously mentioned, it will come in handy to have the average daily price series to work with, so we submit it to the same trio of unit root tests:

Test	**H0**	**Stat-Test**	**p-value**	**Outcome**
ADF (16 lags)	Unit Root	-5.812	< 0.01	Stationary
PP (short lags)	Unit Root	-867.09	< 0.01	Stationary
KPSS (short lags)	No Unit Root	15.5315	< 0.01	Not Stationary

Table: Unit Root tests results on the daily dataset

EDA: To Sum Up

Concluding our exploratory data analysis, we sum up some important features of our data:

1. We are facing **high frequency data**, which might be difficult to work with;
2. PUN time series displays **complex patterns of seasonality** due to the very nature of the series itself;
3. We have identified graphically a **common structure** both in the **conditional mean** and the **conditional variance** of the series.
4. From ADF and PP test we should establish our data are stationary, both when considering the whole, high frequency, series as well as when we test the average daily price series. KPSS test gives the opposite result, so we are left with a puzzling result.

Section 3

Proposed approaches and models

Subsection 1

sARIMA

sARIMA Modelling

Autoregressive Integrated Moving Average model is the workhorse for economists when it comes to forecasting univariate time series, being able to deal with time-dependency in the process originating the data (AR), as well as the possibility of the process being non-stationary (I) and of showing some kind of structure in the mean (MA). **Seasonal-ARIMA** is the extension needed when data show periodic patterns, which is the case for electricity prices. We can write down the general model $sARIMA(p, d, q)x(P, D, Q)_s$ as:

$$\Phi_P(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \delta + \Theta_Q(B^S)\theta(B)w_t$$

where w_t is a Gaussian white noise process. The ordinary autoregressive and moving average components are represented by polynomials $\phi(B)$ and $\theta(B)$ of orders p and q respectively. Seasonal autoregressive and moving average components are written as $\Phi_P(B^s)$ and $\Theta_Q(B^S)$ of orders P and Q. Ordinary and seasonal differences components are given by ∇^d and ∇_s^D. Finally, S represents the seasonal period.

Long-Casting and Short-Casting Electricity Prices

We decided to use two different subsets of our data depending on the forecasting horizon we wanted to test:

1. **Long-Run Forecasting**: forecasting the exact price during hours of the day can be tricky with a forecasting horizon longer than a few days, and it might also be redundant for businesses who only need a general idea of what that price will be; therefore, we employ **average daily prices** going from Jan 1, 2008 to Dec 31, 2018 for long-run forecasting.
2. **Short-Run Forecasting**: using only a subset of one month (in our case, going from Dec 1, 2019 to Dec 28, 2019) for fitting the model, it is possible to obtain hourly predictions for two-three days ahead.

The subsetting approach is followed for practical purposes, since it is not possible to obtain any good prediction with a sARIMA model challenged with more than 100.000 observations. Both methods of subsetting have their ups and downs, mainly due to the presence of a trade-off between forecasting horizon and ability to employ one kind of seasonality instead of the other.

sARIMA: Modelling Average Daily Price of Electricity (1/2)

First, we take the average daily price of electricity. The reshaping operation leaves us with 4018 observations going from Jan 1, 2008 to Dec, 31, 2018.
Since we are now dealing with daily data, we lose the possibility of accounting for the day-night kind of seasonality: our decision is then to procede with $S = 7$, to account for weekly seasonality. Through grid-searching and the helpful R's function $auto.arima()$ (which helps selecting the model minimizing AIC and BIC), we identify the model to use as:

$$sARIMA(5, 1, 1)x(1, 0, 1)_7$$

IC	**Value**
AIC	6.58478
BIC	6.59889

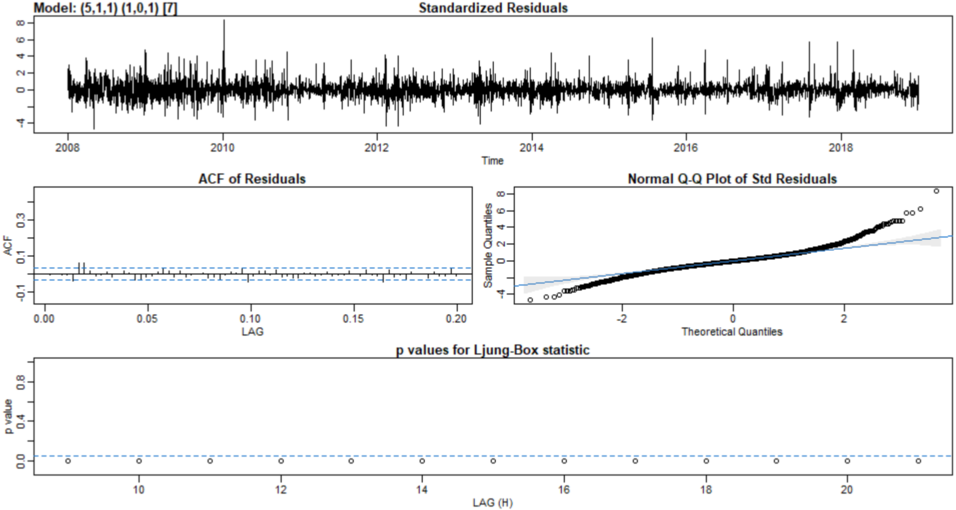
Table: Information Criterions of proposed model

PM	**Value**
MAE	2.13017
MAPE	7.44287
RMSE	3.07399

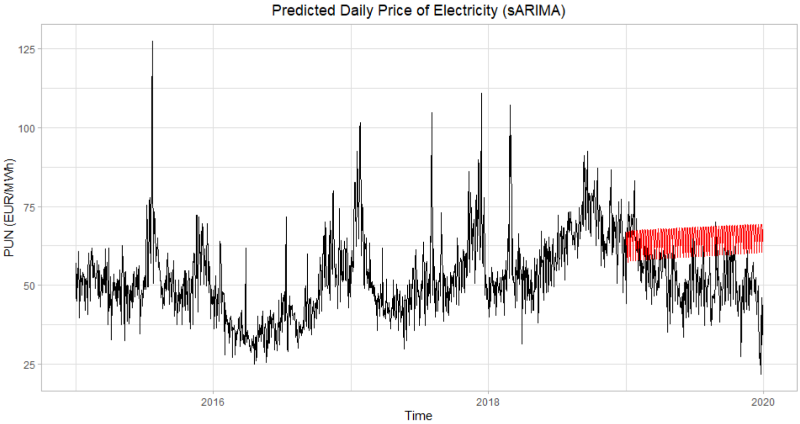
Table: Performance metrics of proposed model

sARIMA: Modelling Average Daily Price of Electricity (2/2)

Model fitting and residual analysis for $sARIMA(5,1,1)x(1,0,1)_7$ on daily prices



sARIMA: Forecasting Average Daily PUN (**year-ahead**)



Unfortunately, sARIMA fails in forecasting daily PUN, probably because of the misreading of the last trends in the mean of 2018 prices. We are unable to say our performance is satisfactory here.

sARIMA: Modelling Hourly Price of Electricity (1/2)

If the purpose is to forecast hourly prices instead of daily prices, our suggestion is to utilize only the last month of observations to fit the model, and then forecasting the next 24/48/72 hours. We utilize here a subset of hourly prices going from Dec 1, 2019 to Dec 28, 2019: that is to say, we are trying to forecast the last three days of our dataset.
Clearly, the relevant seasonality here is of the intra-day kind, so we decide on $S = 24$ and then, according to *auto.arima*(), we set foot on the following model:

$$sARIMA(5,1,2)x(1,0,1)_24$$

IC	Value
AIC	5.18284
BIC	5.25019

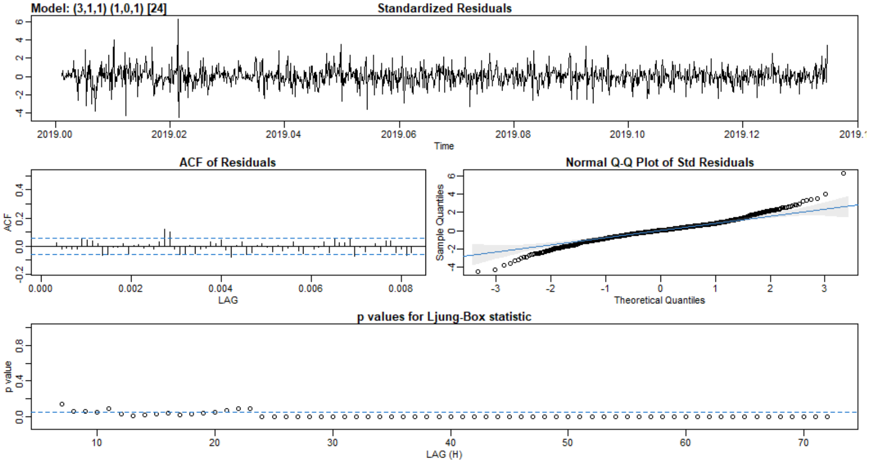
Table: Information Criterions of proposed model

PM	Value
MAE	4.52678
MAPE	7.51827
RMSE	6.35266

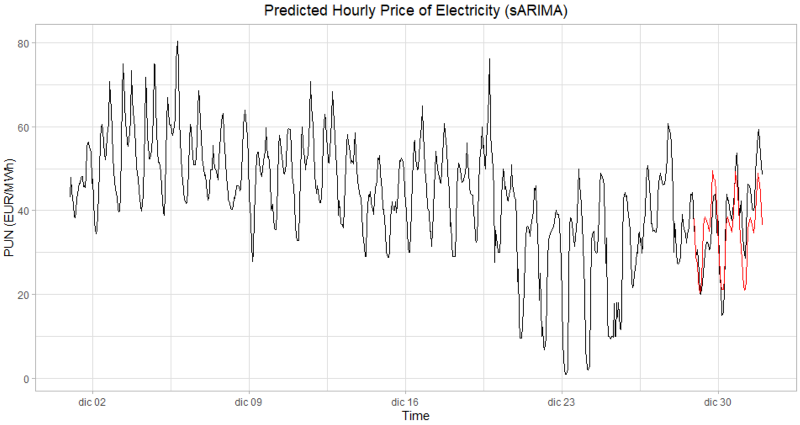
Table: Performance metrics of proposed model

sARIMA: Modelling Hourly Price of Electricity (2/2)

Model fitting and residual analysis for $sARIMA(5,1,2)x(1,0,1)_24$ on hourly prices



sARIMA: Forecasting Hourly PUN (**day-ahead**)



Surprisingly, sARIMA modelling performs way better with a higher frequency dataset and a shorter horizon than it does when trying to predict averages in the long run. To us, model's performance on the three days-ahead test set is very satisfactory.

sARIMA Modelling: Conclusions

Seasonal-Arima does well when we try to forecast hourly prices day-ahead, but its performance much worsens when trying to predict average daily prices year-ahead, no matter the subset size. Generally speaking, sARIMA modelling is subject to all its **limits** here:

- Forecasting performance tends to deteriorate very rapidly according to the number of step ahead to forecast;
- As we previously showed, electricity prices show at least three seasonal patterns (daily, weekly, monthly): sARIMA is able to model only one periodicity at a time;
- sARIMA models, by their very own nature, are not able to deal with structure in the conditional variance, which electricity prices do show.

Subsection 2

Prophet

What is Prophet?

Prophet is a forecasting procedure developed by two Data Scientist of **Facebook**'s team, implemented in R and Python, available for download[5] on CRAN and PyPI.

- It allows to forecast time series data based on additive model where **non-linear trends** are fitted with yearly, weekly and daily seasonality (including holiday effects);
- It's **fully automatic**: Prophet can obtain reasonable forecasts with no manual effort, and it's robust to outliers, missing data and structural breaks;
- It's **fast**: the trade-off between learning how to implement it and its predictive power is outstandingly convenient;
- Prophet allows **tuning** thanks to human-readable parameters, letting the forecaster improve its performances by adding his/her domain knowledge.

[5] link to Prophet's Github repository: `https://facebook.github.io/prophet/`

Prophet and the PUN

According to its developers, Prophet works best with time series that have strong seasonal effects and several seasons of historical data, a description that seems to fit well with PUN. Therefore, we decided to employ Prophet both for long and short run forecasting of electricity prices, using the same subsets utilized for sARIMA modelling:

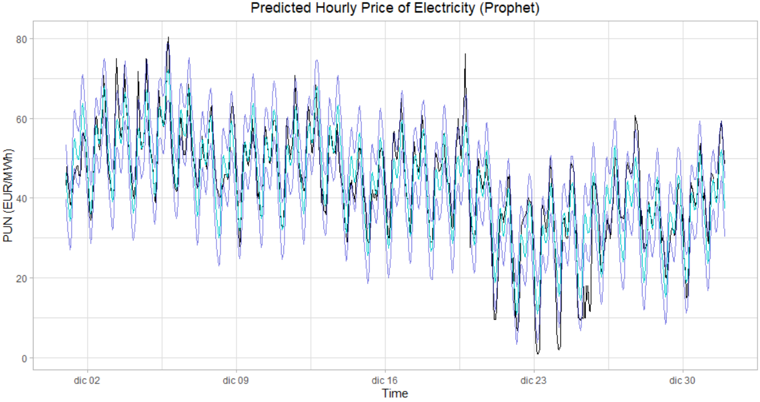
- A **hourly subset** going from Dec 1, 2019 to Dec 28, 2019 to prepare Prophet for **short-run forecasting**;
- The whole **daily average price series** going from Jan 1, 2008 to Dec 31, 2018 for year-ahead, **long-run forecasting**.

Prophet's Cross Validation and Evaluation Metrics

The command *cross_validation()* computes forecasts of the given model from historical cutoff points. Beginning from (end - horizon), it works backwards making cutoffs with a spacing of period until initial is reached.
From those forecasts, it is possible to output a table with various evaluation metrics using *performance_metrics()*:

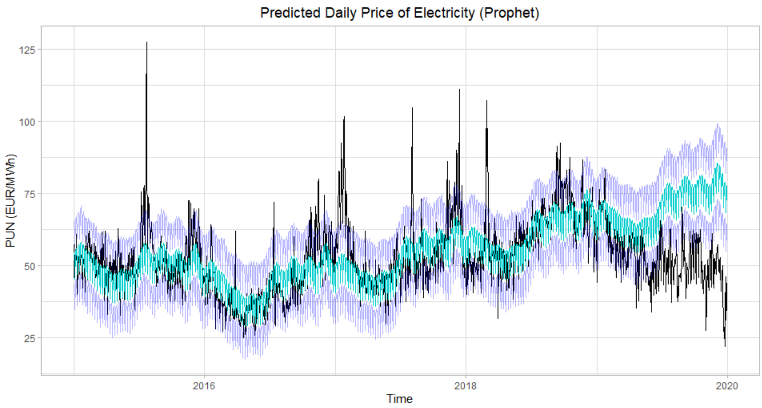
Evaluation Metric	**Hourly Data**	**Daily Data**
MAE	143.45283	17.42714
MAPE	2.88789	0.28711
RMSE	444.80080	27.33718

Prophet's Short-Run Forecast



Prophet outcomes acceptable results when trying to forecast in the short run: here we can observe the hourly model fit for December 2019, including the prediction of the last three days compared to the actual value. Confidence intervals are given in a darker color.

Prophet's Long-Run Forecast



Prophet should also be able to mimic seasonal year-wide patterns and to outcome long-run forecasts for average daily price of electricity. Here, we can see how the model fits along the years (only data since 2015 are shown). Just like Seasonal-ARIMA, Prophet fails in correctly forecasting 2019's averages, probably for the same practical issue.

Subsection 3

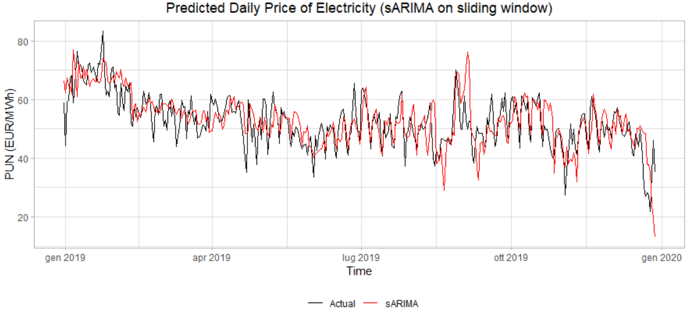
Sliding Windows

Solving the Long-Casting Task

As we noticed employing basic, semi-automatic, procedures, both sARIMA and Prophet outputs somewhat believable results when it comes to forecasting in the short-run, but fail to do so in the long-run, probably because of the very features of the data we are working with.
We decided to get actively involved in the last part of the analysis, and wrote two **for loops** with the goal of running both models on a **sliding window of 31 daily observations**, such as both models would forecast three days ahead, drop the first three observations, add the predictions to the set and then repeat the task.
The first 31 observations given are the average daily prices for December, 2018, and the results obtained following this procedure are quite encouraging.

sARIMA on a Sliding Window of Observations

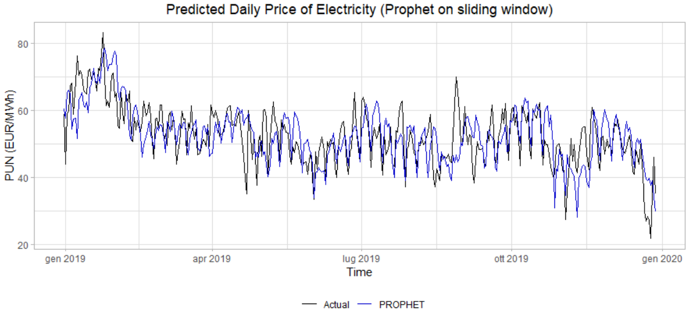
Inside the for loop, we are running a $sARIMA(1,1,1)x(1,0,1)_7$ on a sliding window of 31 observations. We lose a certain degree of automation here, because the model choice is performed via manual grid search of parameters on the first 31 observations, instead of using $auto.arima()$. Nevertheless, results seem believable, according to both the graph and the performance metrics.



PM	Value
MAE	5.151847
MAPE	10.56214
RMSE	6.888072

Prophet on a Sliding Window of Observations

Running Prophet on the sliding window inside the for loop we are able to (almost) maintain the same degree of automation of the procedure, obtaining far more convincing results than running Prophet on the whole average daily series.



PM	Value
MAE	5.450871
MAPE	10.8462
RMSE	6.974609

Section 4

Conclusions

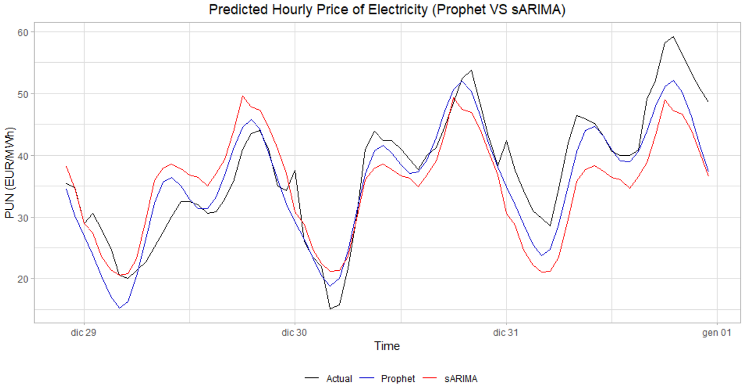
Automate Forecasting of High-Frequency Datasets? (1/2)

In the EDA section, we presented the reader with the PUN time series, a high frequency dataset which presents structure both in the conditional mean and conditional variance. Usually, one would need some heavy time series schooling (say: GARCH models and/or ARMAX processes) to properly model and getting predictions out of this set, as well as some domain knowledge of the field the data comes out of.
Domain knowledge brings a higher level of complexity to the table: indeed, for the very way electricity is exchanged in liberalized markets, one has an interest in predicting its price tomorrow and in having an idea of its possible price the next month-semester-year.
For the vast majority, enterprises do not have on their dependencies someone with that skill level, so the question is whether or not one could reach acceptable results with only univariate regressions like ARIMA models, or to even make the forecasting procedure semi-automatic.

Automate Forecasting of High-Frequency Datasets? (2/2)

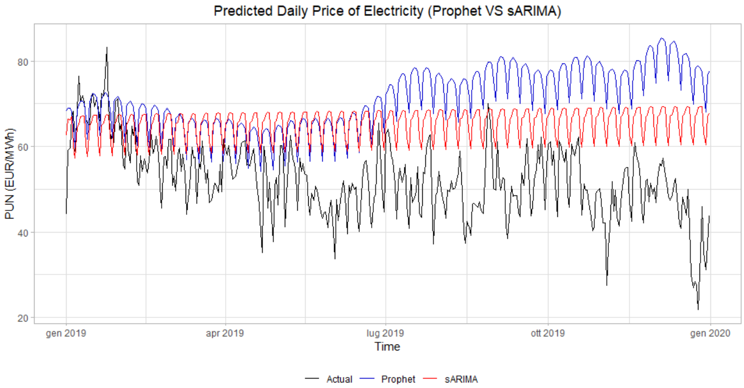
Our resolution is to compare Seasonal-ARIMA modelling, obtained with little time series knowledge and the help of R's *auto.arima*() function, with the automatic procedure developed by Facebook: Prophet.
We employed two training sets: a first, made out of average daily prices collected for several years and utilized for long-run forecasting, and the second one made of only one month of hourly observations and used for immediate forecasting. While results yielded by the short-run procedure were satisfying, the same couldn't be said for our first attempt at long-run forecasting. We then decided to try a different approach and run both models inside for loops with a sliding windows of observations, and obtained very encouraging results.
The next three slides propose a comparison of the predictions obtained via those methods.

PROPHET VS sARIMA: Shot-Run Forecasting



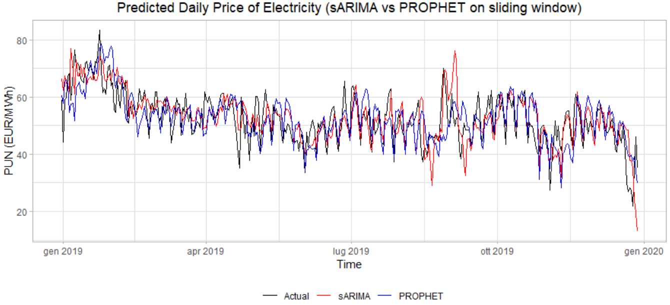
As we can see, both models are able to outcome acceptable predictions for hourly PUN one, two and three days ahead. Prophet however, outperforms sARIMA modelling with this particular subset, as can be seen comparing the *Mean Absolute Percentage Error* of the two models: Prophet's MAPE is equal to 2.89, while sARIMA's amounts to 7.52.

PROPHET VS sARIMA: Long-Run Forecasting



Once again, to correctly compare the performances of two models, we have to look for the MAPE: for year-ahead Prophet's MAPE is equal to 0.29, while sARIMA's amounts to 7.44. This time however, none of the two models is able to yield satisfactory results.

PROPHET VS sARIMA: Long-Run Forecasting with Sliding Windows



Following the sliding window procedure, both sARIMA and Prophet are now able to obtain satisfying predictions in the long-run (one year); interestingly, they produce opposite predictions when the actual data shows high volatility. Metrics comparison suggest that with this data, sARIMA performs slightly better than Prophet, with a MAPE of 10.56 against one of 10.85.

Conclusions. . .

- Both Prophet and sARIMA are able to produce acceptable day-ahead predictions for high-frequency data. In this particular subset of electricity prices, Prophet outperforms sARIMA in the task of immediate, day-ahead forecasting;
- Electricity prices pose a challenge both to sARIMA modelling and Prophet when it comes to forecast in the long-run. We work around this difficulty with a sliding window procedure and obtain good results for both models. Here, sARIMA is slightly better than Prophet, according to performance metrics;
- One with a little of time series schooling could just use Prophet for forecasting specific high-frequency data in the short-run or try out a sliding window procedure for address the long run-forecasting task;
- Having some time series knowledge at disposal, one could employ both approaches, comparing them to one another to see which one performs better in that particular case.

. . . and Next Steps

We feel like the PUN time series pose a serious challenge to practitioners and we'd like to do some more research on it in the near future. A couple of our ideas stepping away from automatic procedures are:

- Following *Alonso et al (2008)* in the development of econometric models able to **combine seasonality with structure in the conditional mean and variance simultaneously**;
- Employing Machine Learning approaches like **Principal Components Analysis**;
- Including **external predictors** in our analysis of electricity prices to strengthen forecasting ability.

Section 5

References

References

Alonso A., García-Martos C., Rodríguez J., Sánchez M. (2008)
Forecasting electricity prices and their volatilities using Unobserved Components
Universidad Carlos III de Madrid Working Papers Working Paper 08-14, Statistic and Econometric Series 06.

García-Martos C., Rodríguez J., Sánchez M. (2011)
Forecasting electricity prices and their volatilities using Unobserved Components
Energy Economics (33) pp. 1227-1239.

Feuerriegel S., Naumzik C. (2020)
Forecasting electricity prices with machine learning: Predictor sensitivity
*s*ubmitted for publication: available **here**.

Letham B., Taylor, S. (2017)
Forecasting at Scale
*a*vailable **here**.

Shumway R., Stoffer D. (2016)
Time Series Analysis and its Applications
Springer fourth edition.

The End