INSEAD The Business School for the World®

Day-ahead power price prediction
(European Energy Exchange – focus on Belgium)

Data Science for Business

Group 1

May 31st 2022

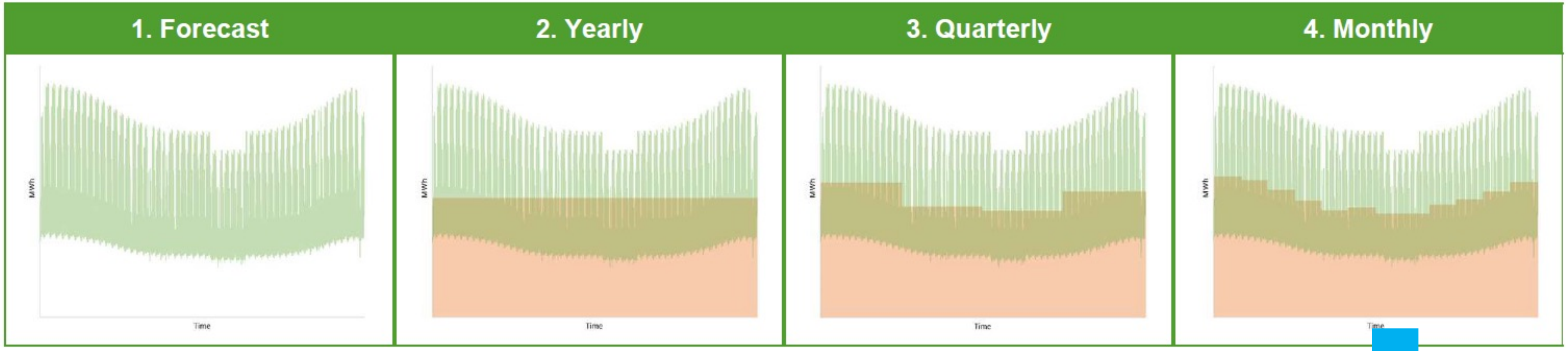
epexspot

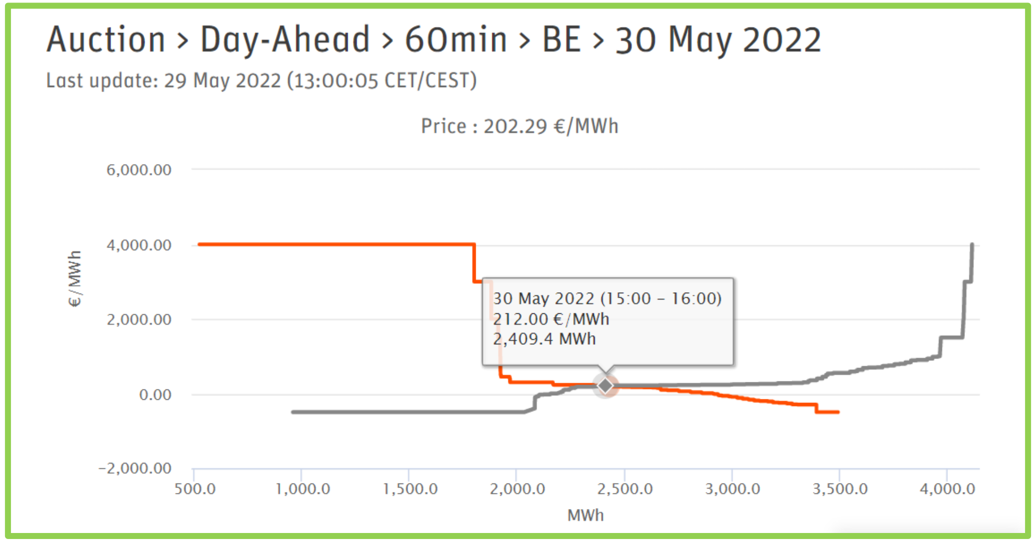
Overview

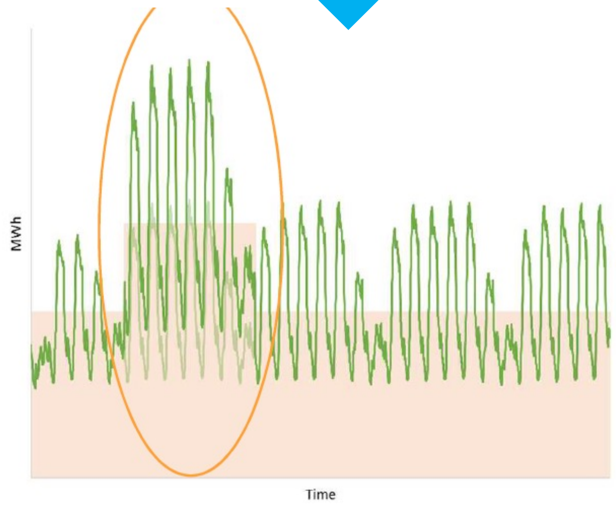
1. Project context and dataset explanation
2. Business case
3. Approach
4. Data Cleaning
5. Timeseries models
6. Supervised learning models
7. Unsupervised learning – data reduction
8. Conclusion

Project context and dataset explanation

Energy players need to balance production and demand at all times - this starts years in advance

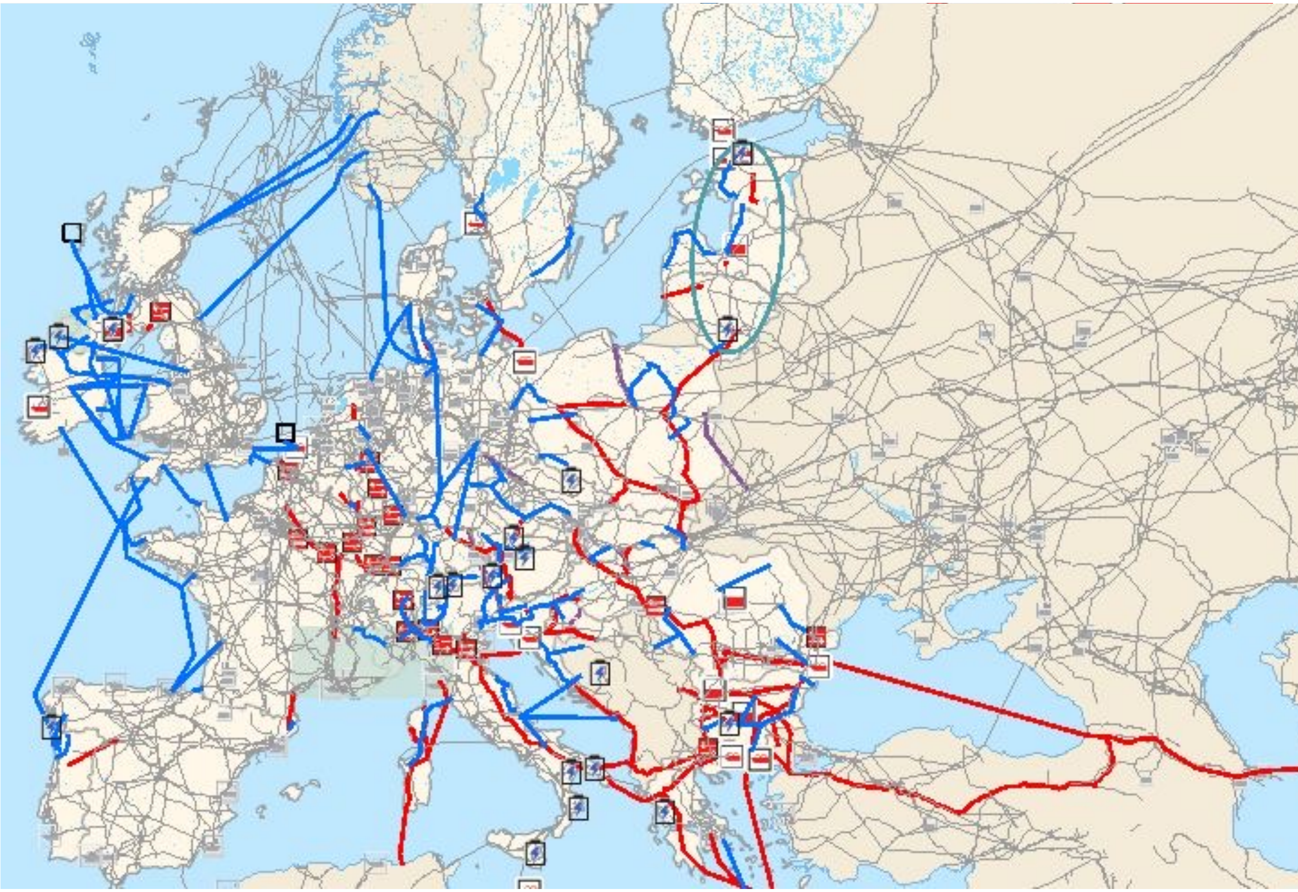






Day-ahead prices are influenced by many different variables

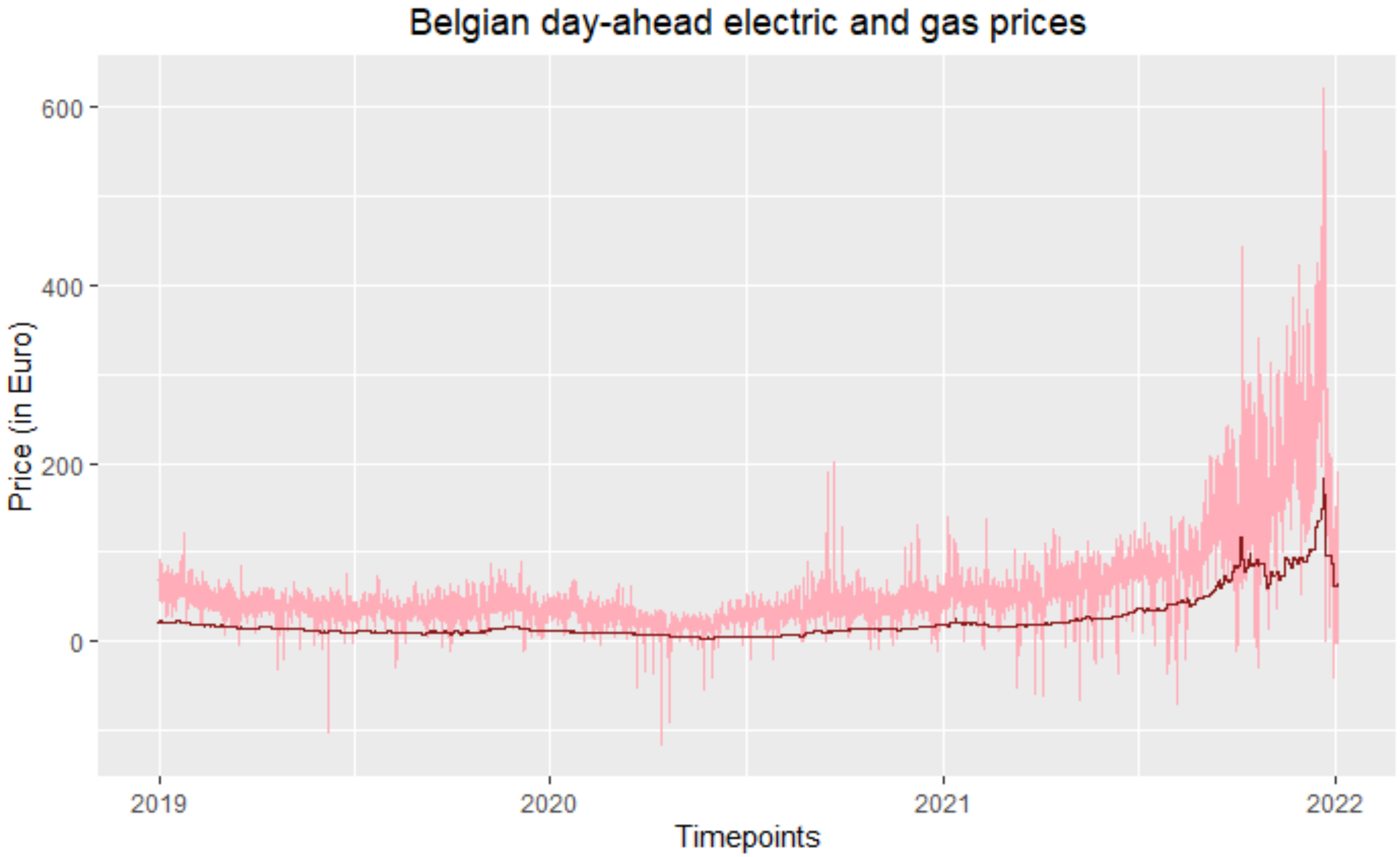
- Season / day of the week / hour of the day
- Electricity demand
- Fuel prices
- Carbon allowance price (EU ETS)
- Renewable energy production
- Availability of power plants
- Cross-border capacity

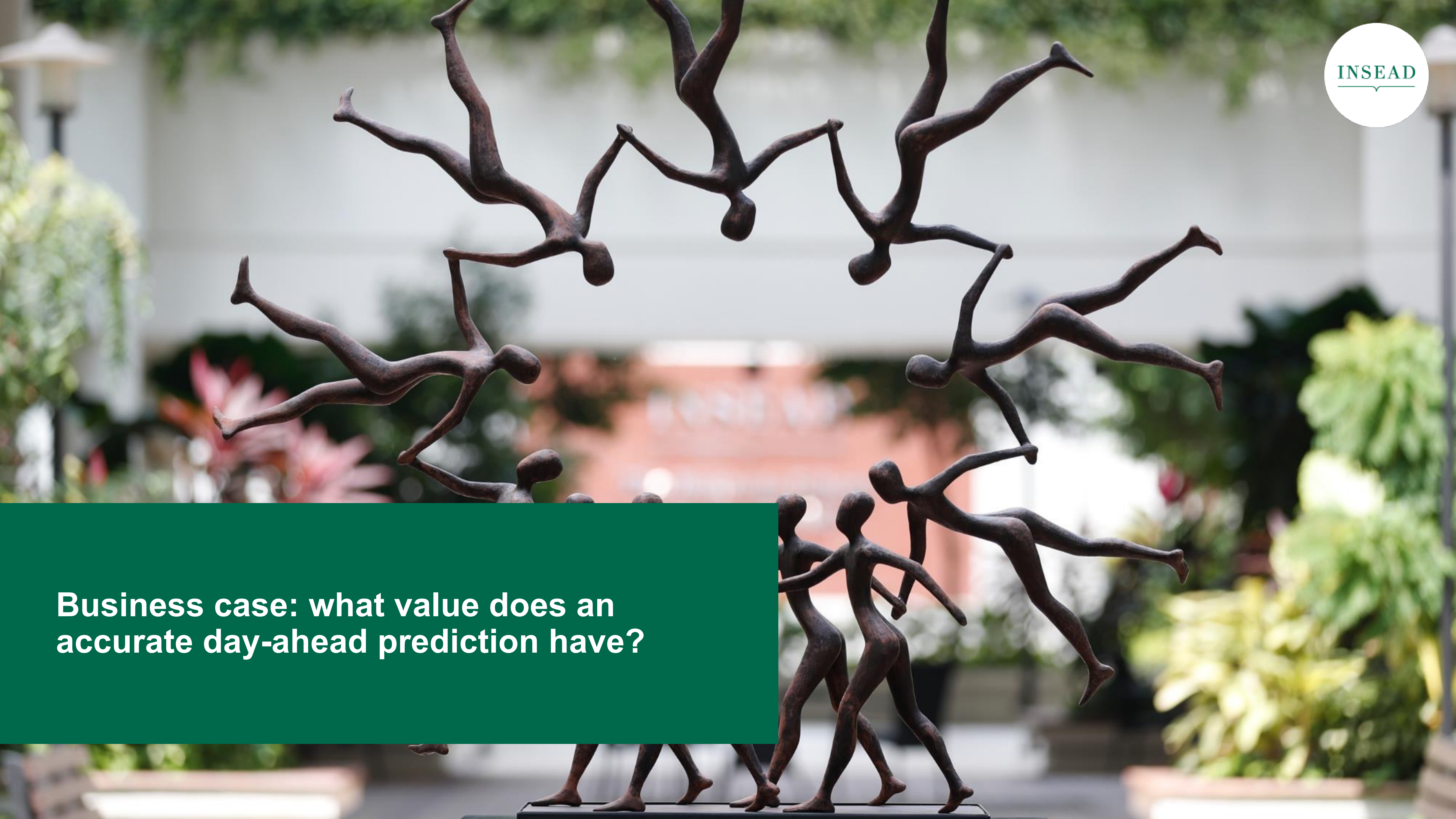


What else has a predictive value?

- Power prices from the day before
- Power prices from the same day a week ago

Power prices are mostly influenced by fuel prices and by geopolitical circumstances





Business case: what value does an accurate day-ahead prediction have?

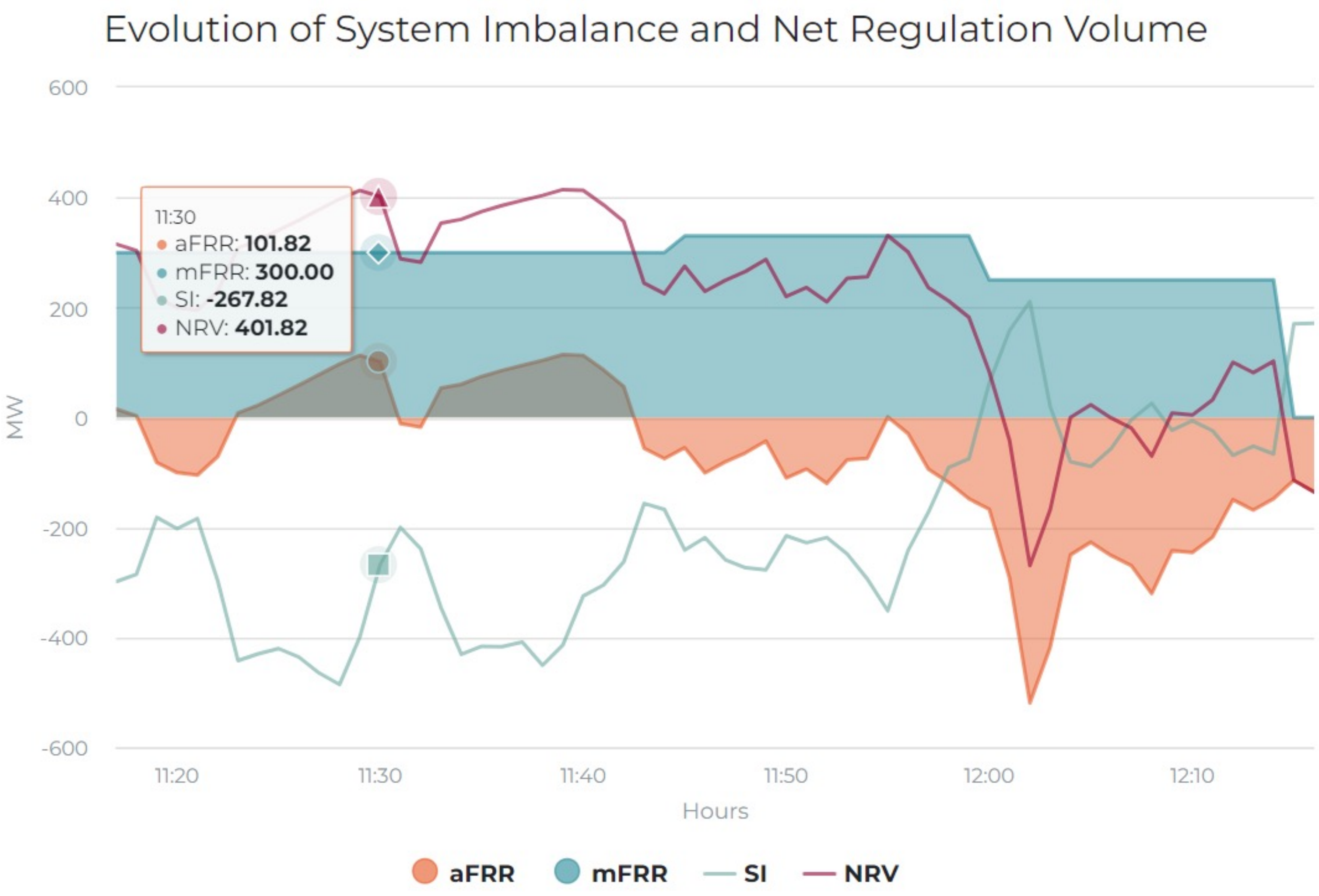
1) Production must equal production also in real-time, but how?

→ Balancing capacity!

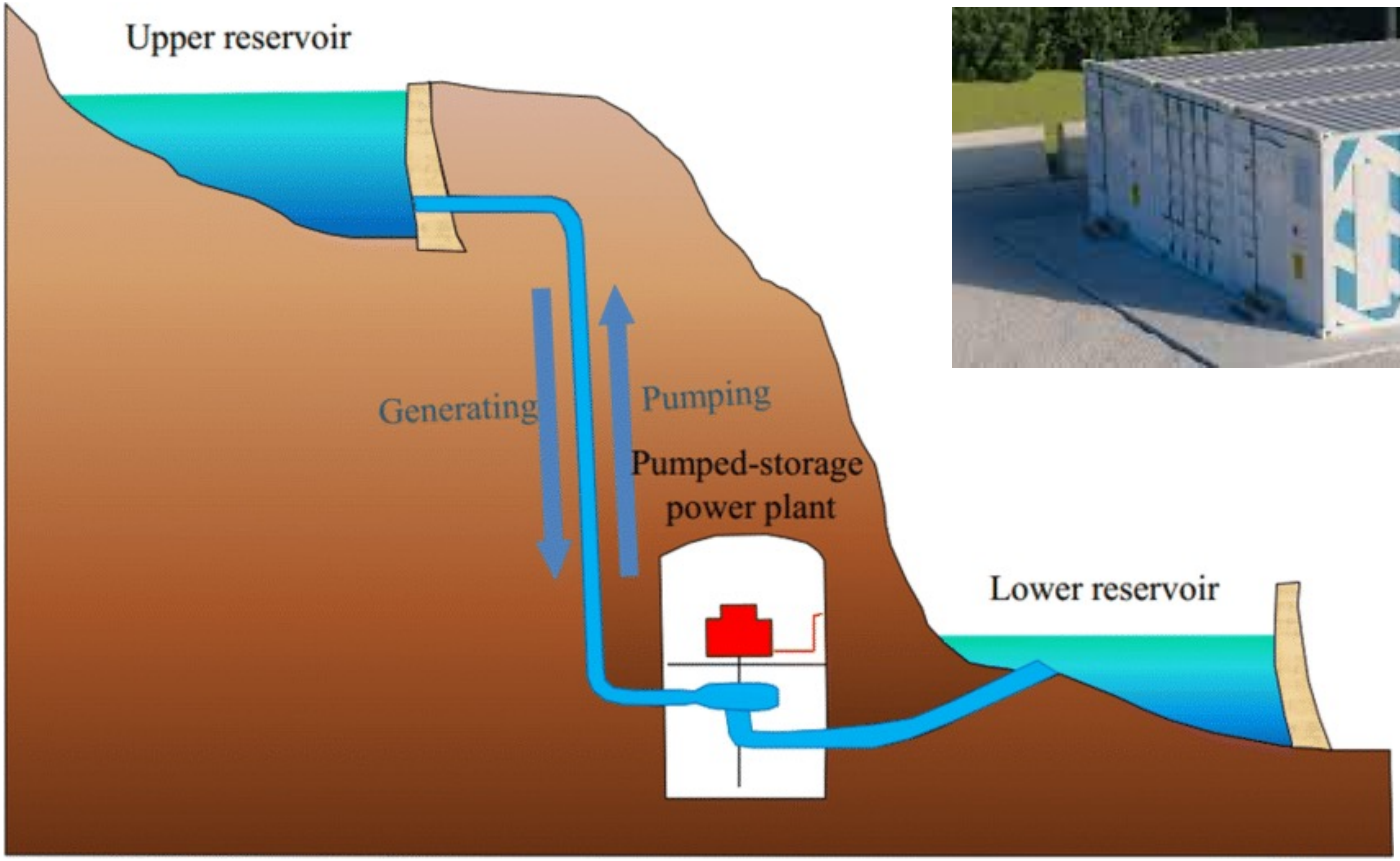
Energy companies are remunerated to guarantee balancing capacity to TSOs

BUT

is providing balancing capacity the most profitable way to use their assets?



2) When best to (de)charge a battery? When best to fill/empty a pumped hydro power plant?



3) Optimal block building for day-ahead auction

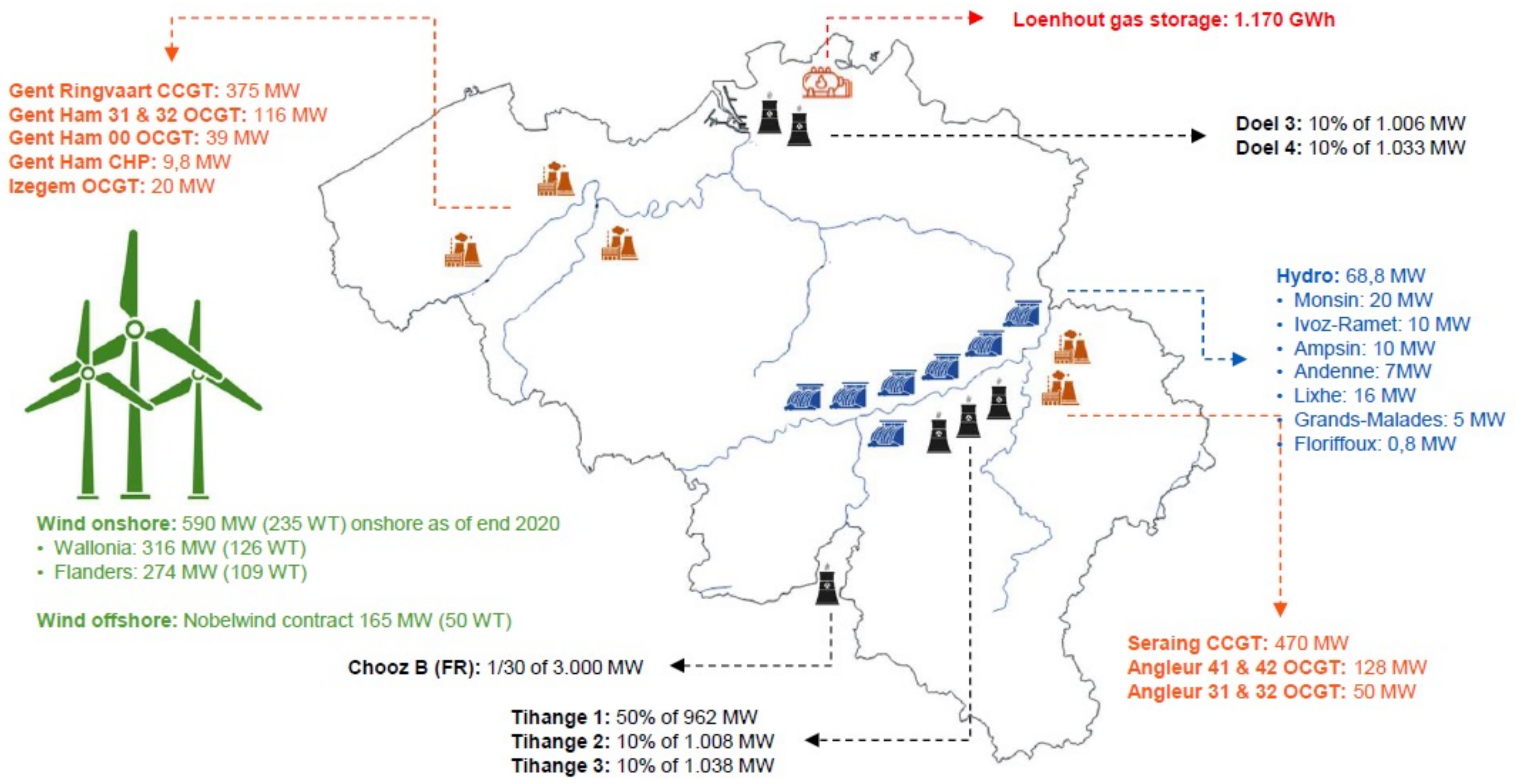
Bidding options:

- Single hours
- Blocks
  - Linked blocks
  - Exclusive blocks
  - Big blocks
  - Loop blocks

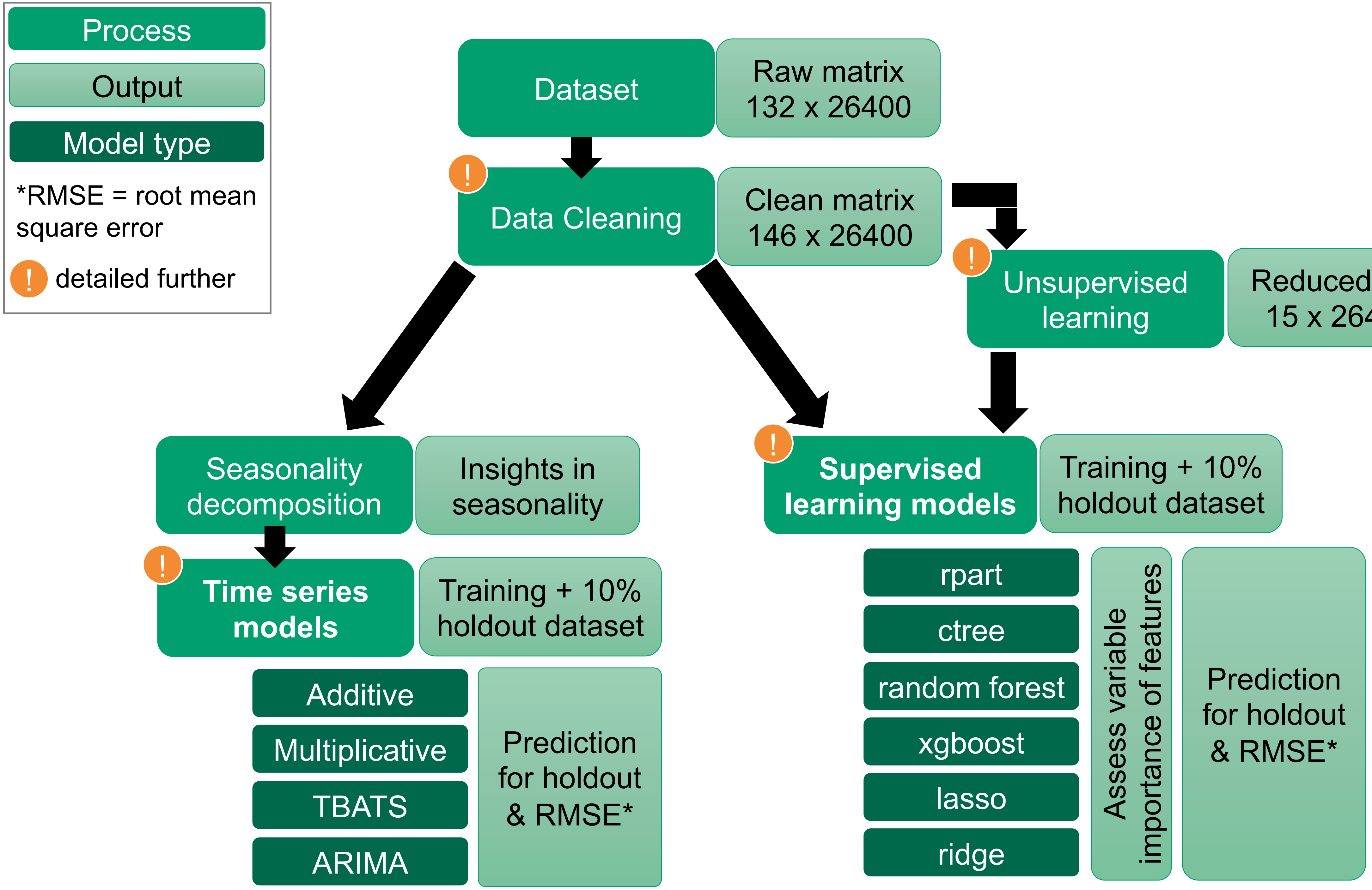
Aim:

- Be “on” (more flexibility)
- Avoid many starts/stops (this deteriorates assets)

4) Determine best timing for short-term maintenance interventions



Approach



Process
Output
Model type
*RMSE = root mean square error
! detailed further
Dataset
Raw matrix 132 x 26400
Data Cleaning
Clean matrix 146 x 26400
Unsupervised learning
Reduced mtx 15 x 26400
Seasonality decomposition
Insights in seasonality
Time series models
Training + 10% holdout dataset
Additive
Multiplicative
TBATS
ARIMA
Prediction for holdout & RMSE*
Supervised learning models
Training + 10% holdout dataset
rpart
ctree
random forest
xgboost
lasso
ridge
Assess variable importance of features
Prediction for holdout & RMSE*

Data Cleaning

Dataset cleaning steps

INSEAD

Timeseries models	Supervised learning models

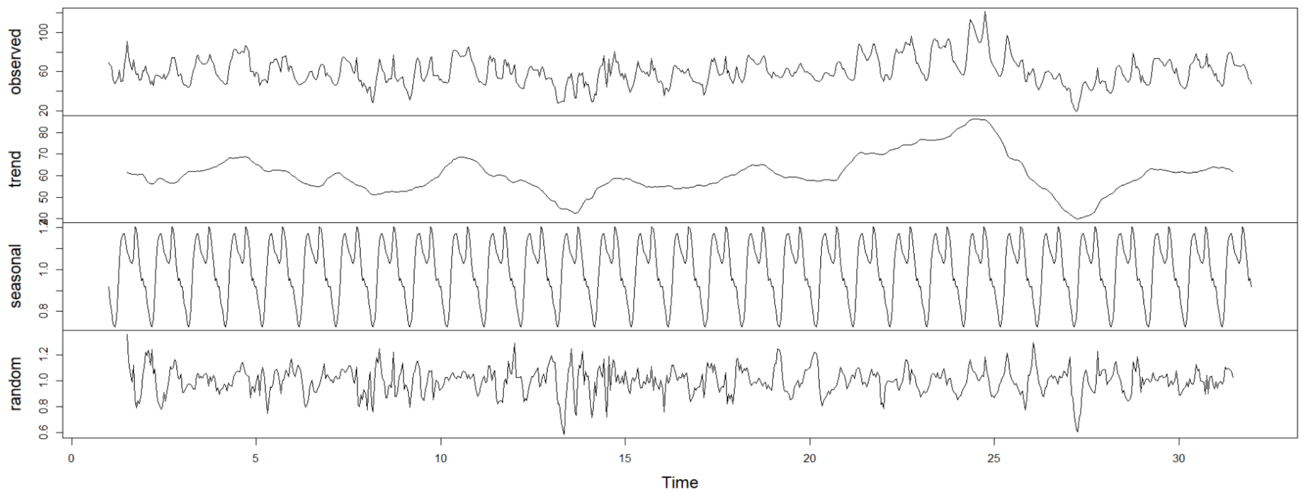
- ✓ Change column classification of variables, e.g. hour & weekday
- ✓ Remove outliers, e.g. error in load flow calculation on 6/8/’19
- ✓ Fill missing data with FixNAs function, this adds extra columns

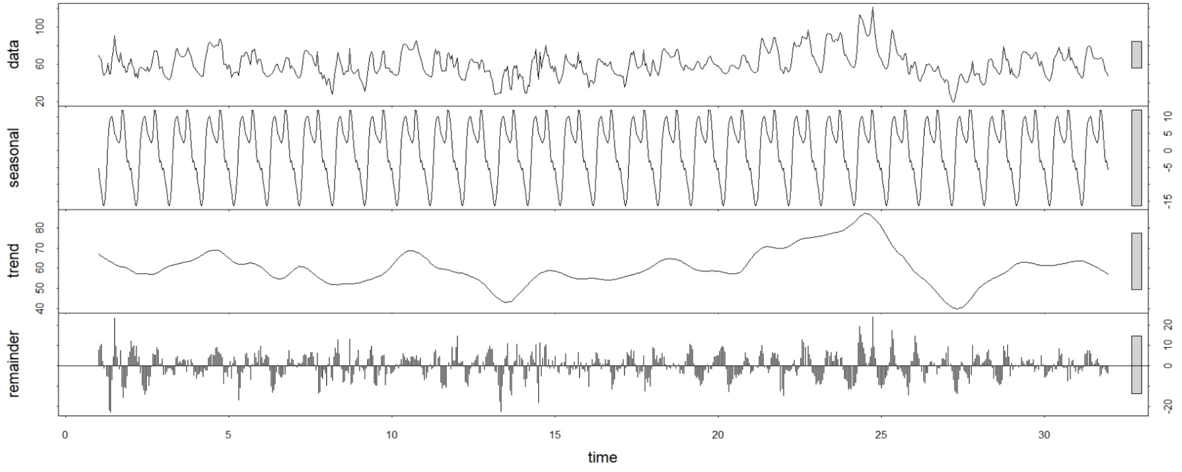
Timeseries models	Supervised learning models
✓ Extract date and time with "POSIXct" for time series models	✓ Remove date/time columns

Timeseries models

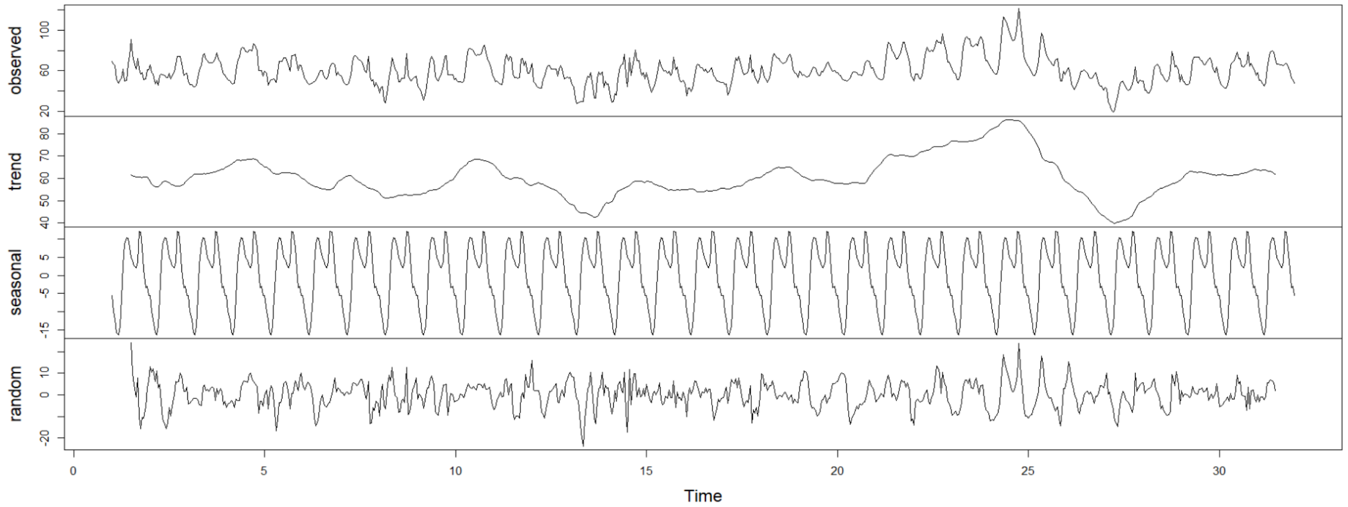
Seasonality decomposition with additive, multiplicative and STL all show daily seasonality

Decomposition of multiplicative time series



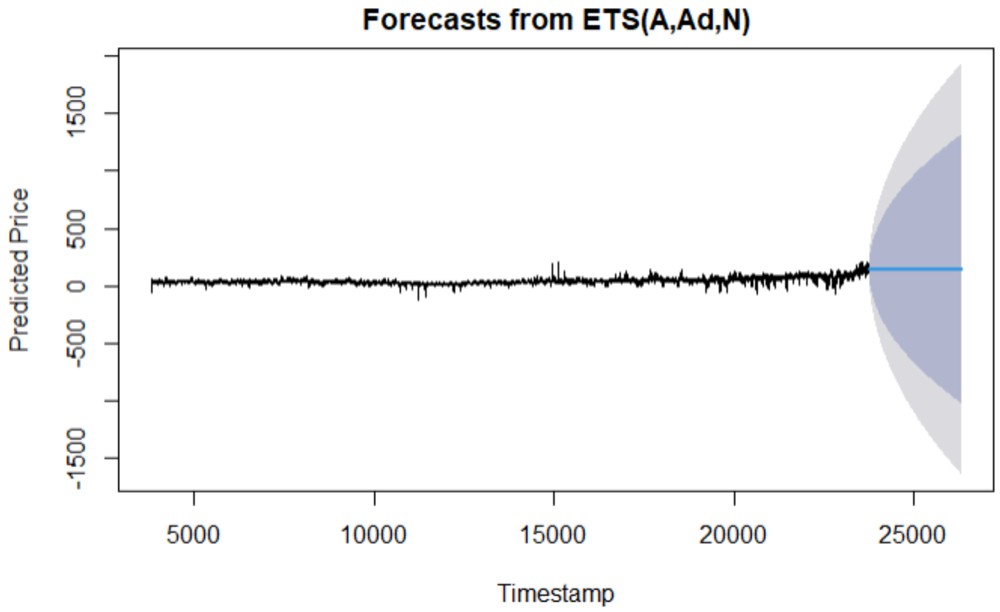


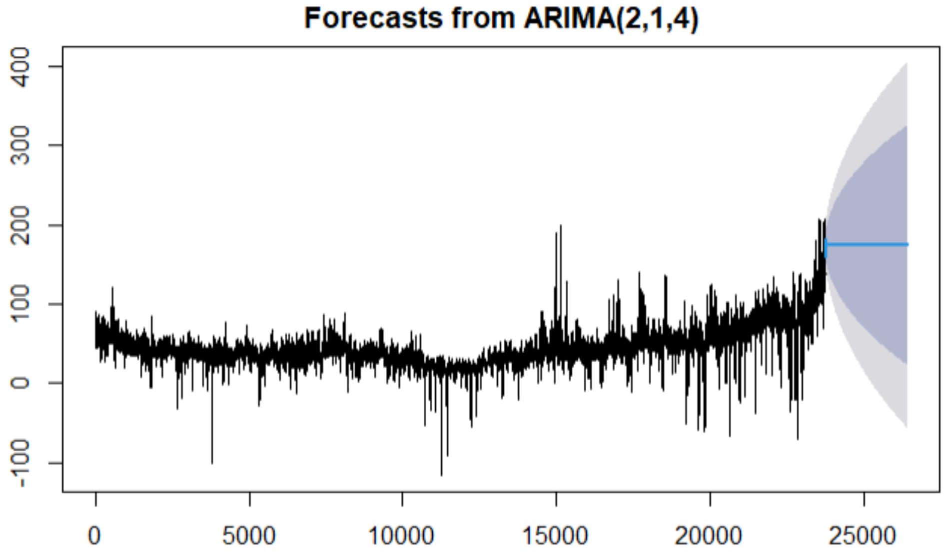
Decomposition of additive time series



Results of timeseries models: different challenges occur when using these techniques

Model	RMSE	Main issue
Additive	Inf	Underfitting, poor predictive power
TBATS	Inf	Very slow to run
ARIMA	Inf	Underfitting, poor predictive power





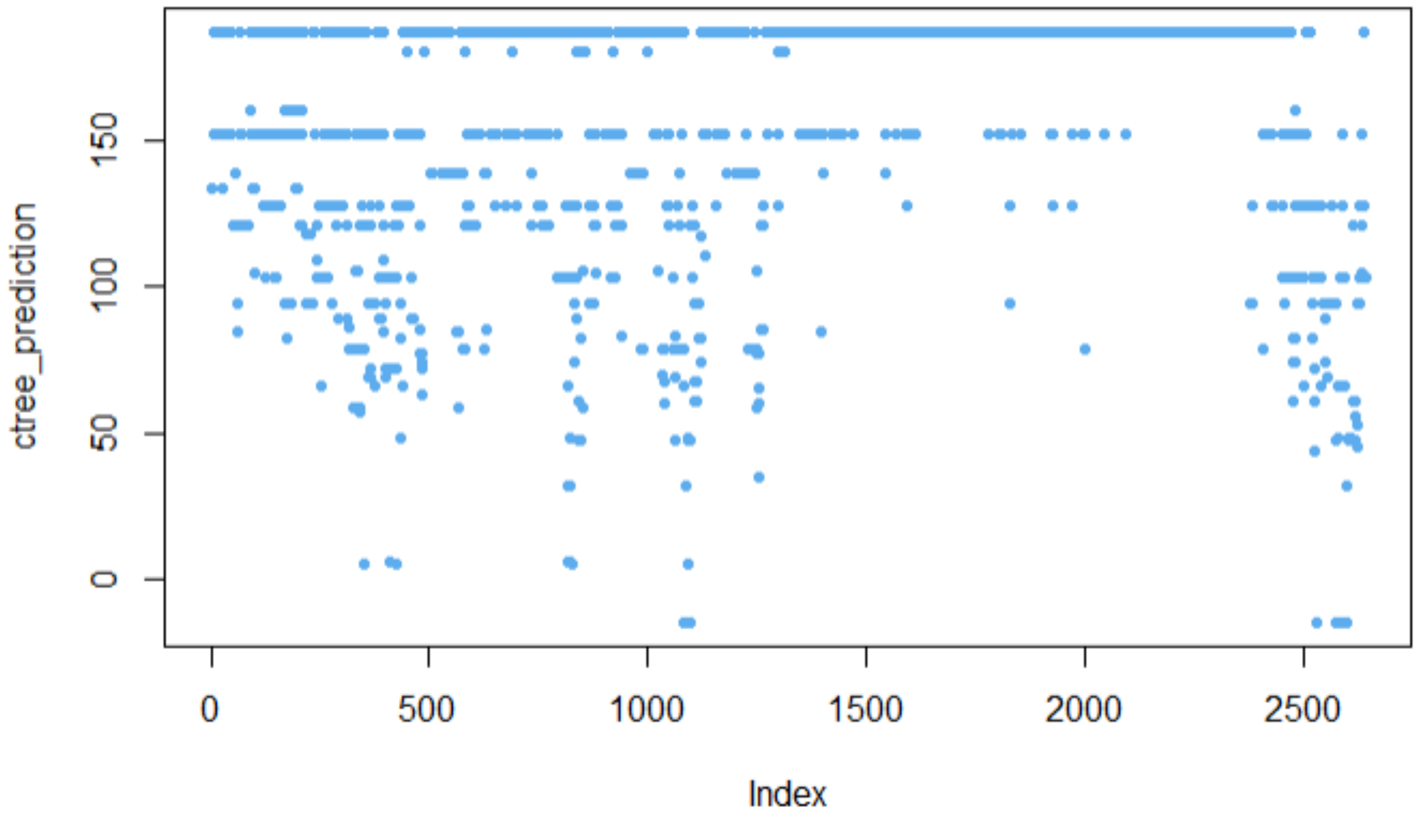
Supervised learning models

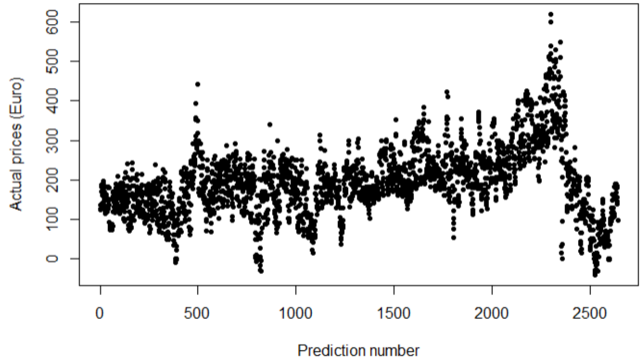
Results of supervised learning models: higher predictive value than timeseries

Model	RMSE[1]	MASE[2]	Comments
rpart	NA	NA	Extremely slow to run even with cp =0.1
ctree	78.7	2.8	Poor predictive power of all three, tuning the models did not help; likely overfitting in xgboost
random forest	110.8	4.4	
xgboost	92.0	3.4	
lasso	51.2	1.9	Lower errors compared to other models, capture trends better but low predictive power
Ridge	59.6	2.3	
Lasso with reduced data	92.3	3.7	Deteriorates performance of the lasso model with full dataset

1. RMSE = root mean square error, i.e. the quadratic mean of the differences between predicted values and observed values
2. MASE = mean absolute scaled error, i.e. mean absolute error of the forecast values, divided by the mean absolute error of the in-sample one-step naive forecast

Ctree



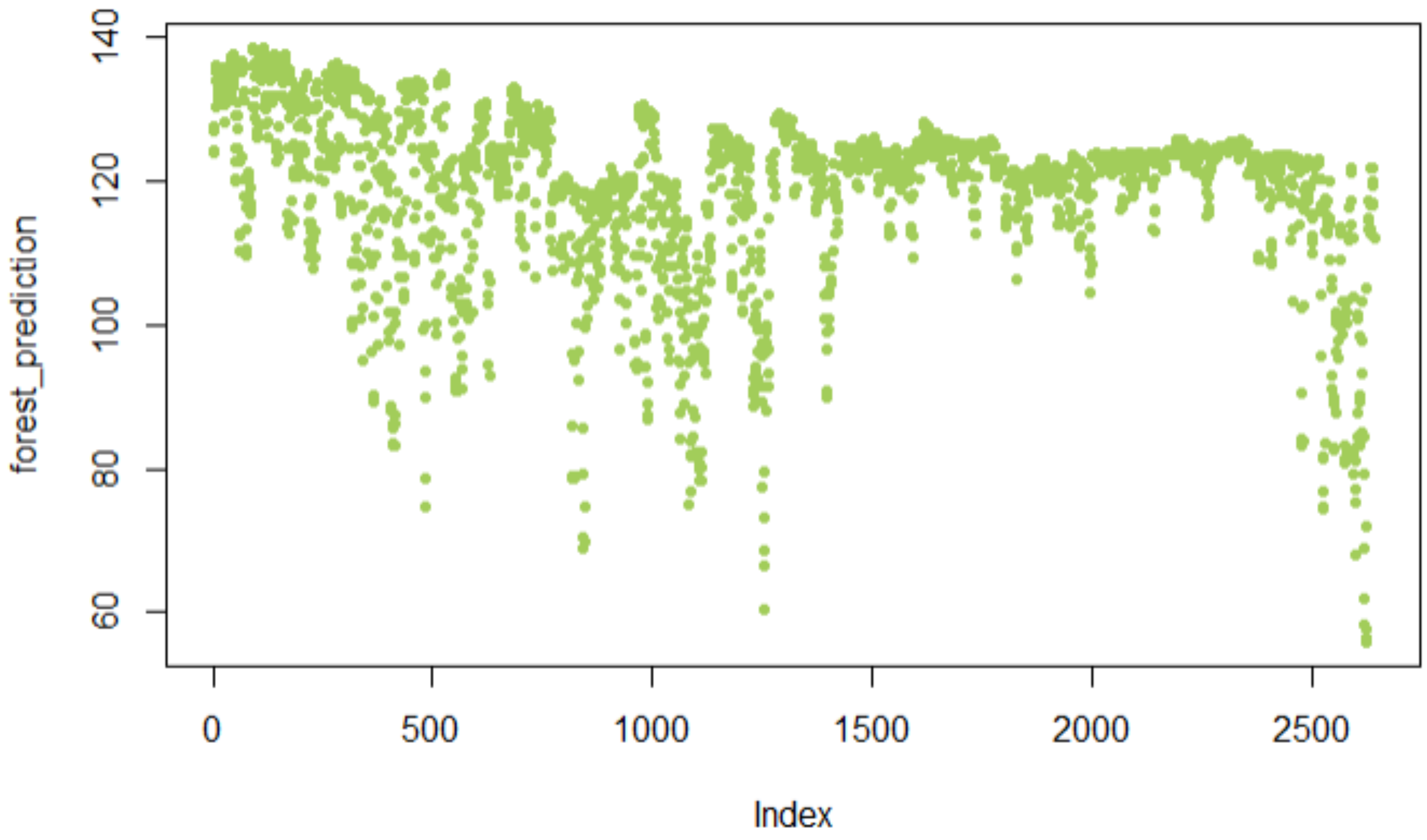


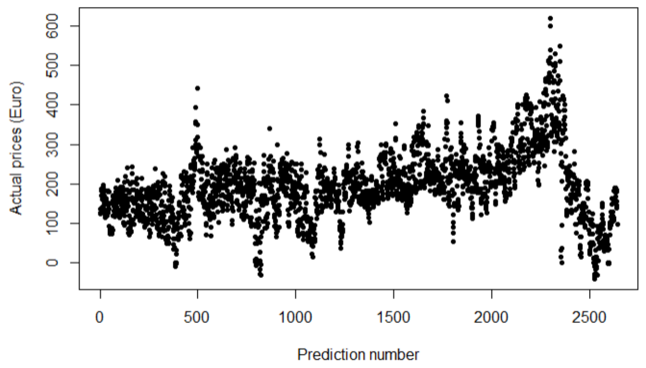
Results	
RMSE	78.7
MASE	2.8

Learnings The decision making process of the model

Challenges Takes a very long time to run and plot

Random forest





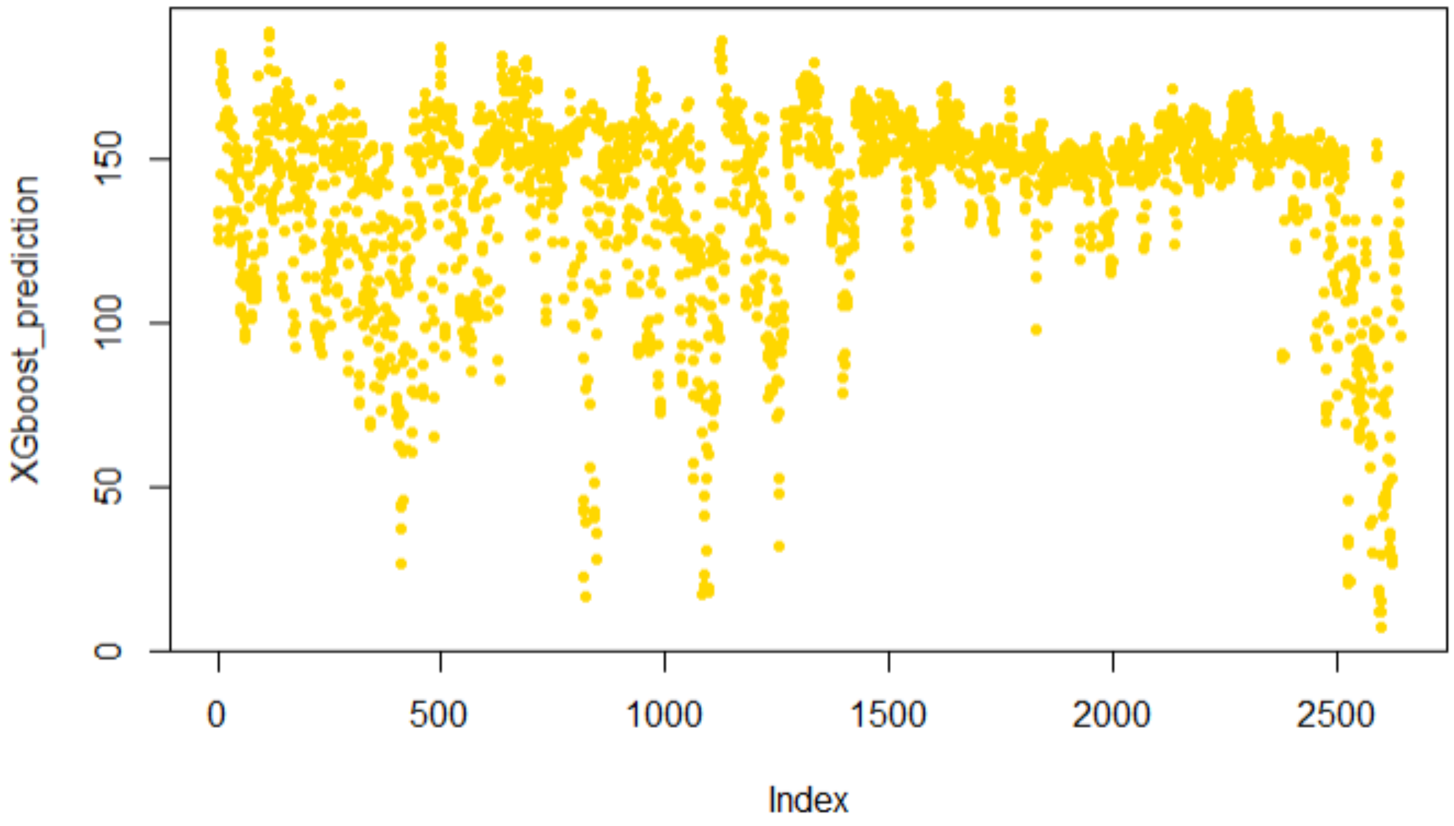
Parameters used
ntree = 500
mtry = 10
nodesize = 10
maxnodes = 10
cutoff = c(0.5, 0.5)

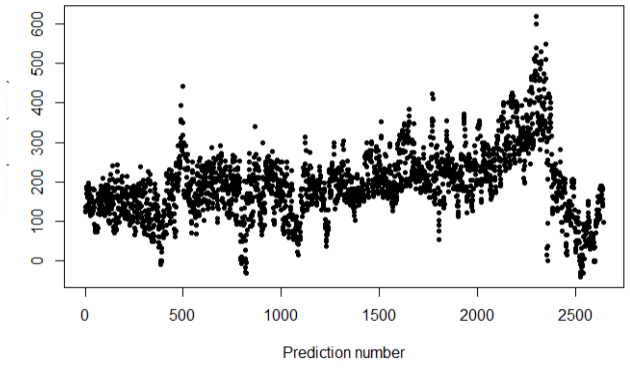
Results	
RMSE	110.8
MASE	4.4

Learnings The top 10 most important features are lagged prices (day -1 and day -7), gas price on different hubs and predicted load

Challenges Overfitting

xgboost





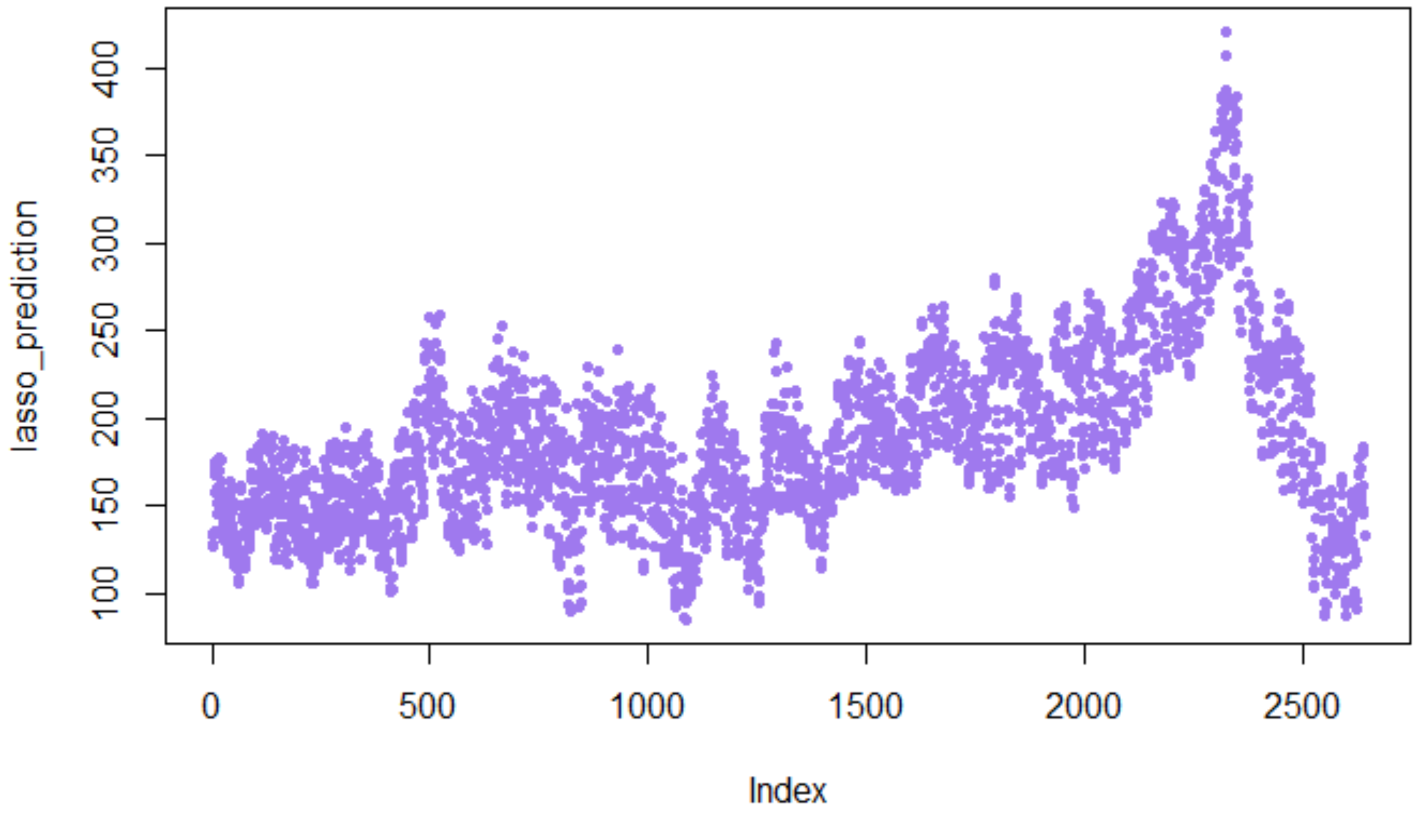
Parameters used
Eta=0.1
Max_depth = 20
Nround = 60

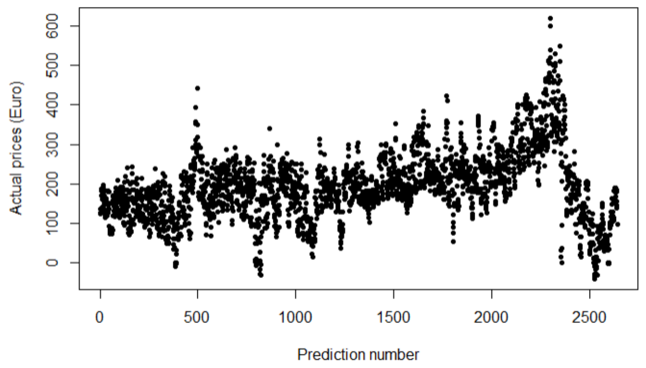
Results	
RMSE	92.0
MASE	3.4

Learnings Most important features are lagged day-ahead prices and renewable energy production forecast

Challenges Overfitting

Lasso





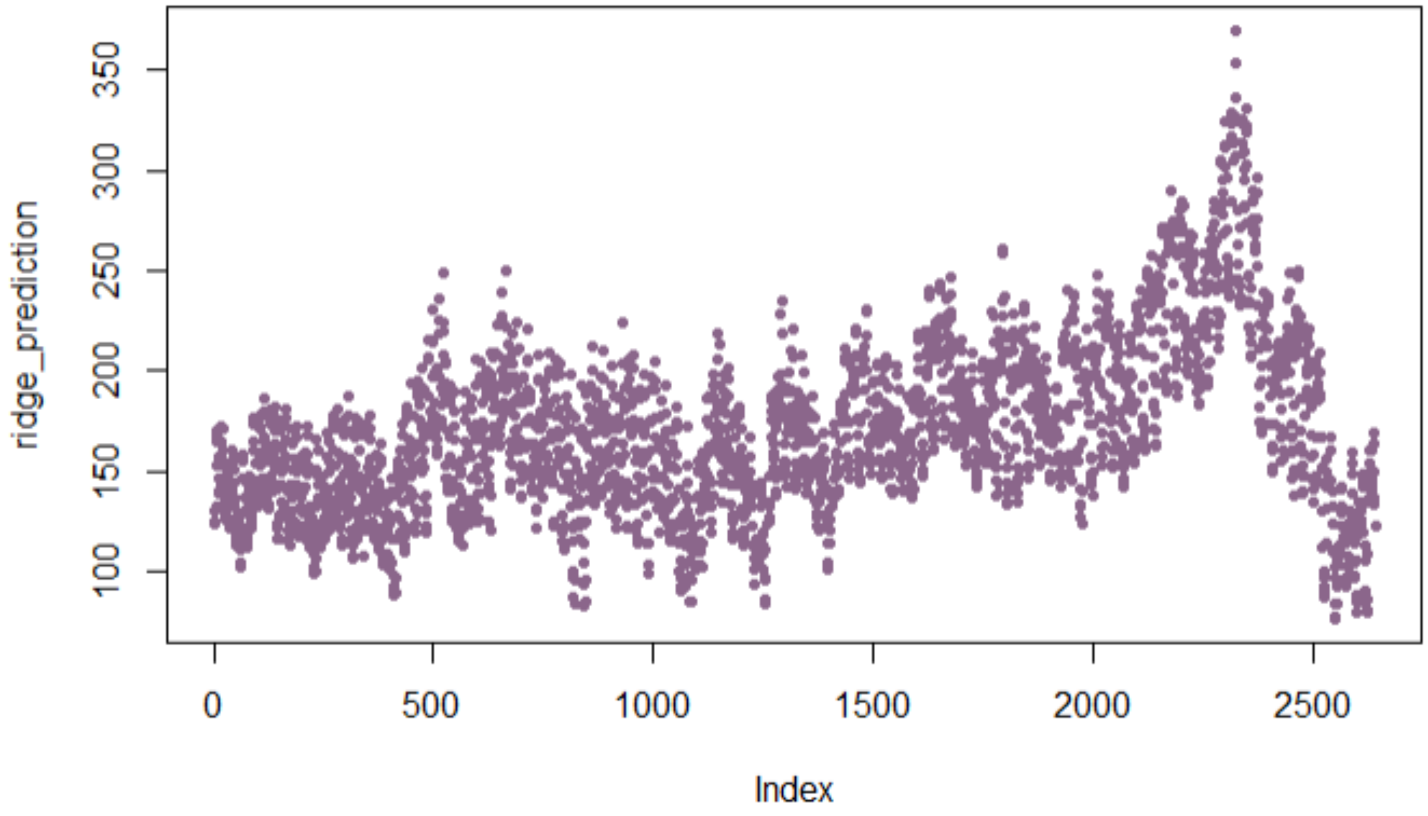
Parameters used
alpha = 1
lamda = 1.0412

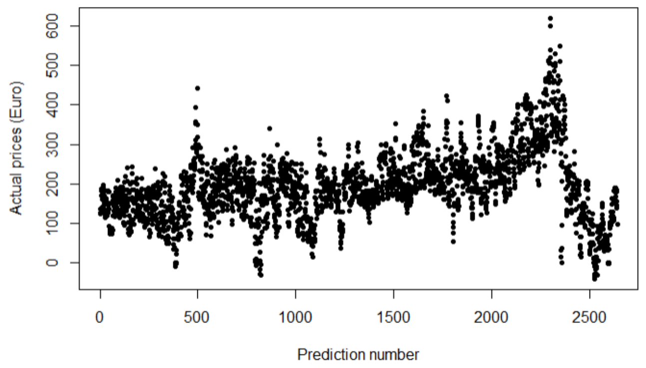
Results	
RMSE	51.2
MASE	1.9

Learnings Optimal log(lambda) is -1.2 with approx. 30 variables remaining

Challenges Getting the model more accurate

Ridge





Parameters used
alpha = 0
lambda = 2.988

Results	
RMSE	59.6
MASE	2.3

Learnings Optimal log(lambda) is around 3

Challenges It is harder to see how many variables remain

Unsupervised learning

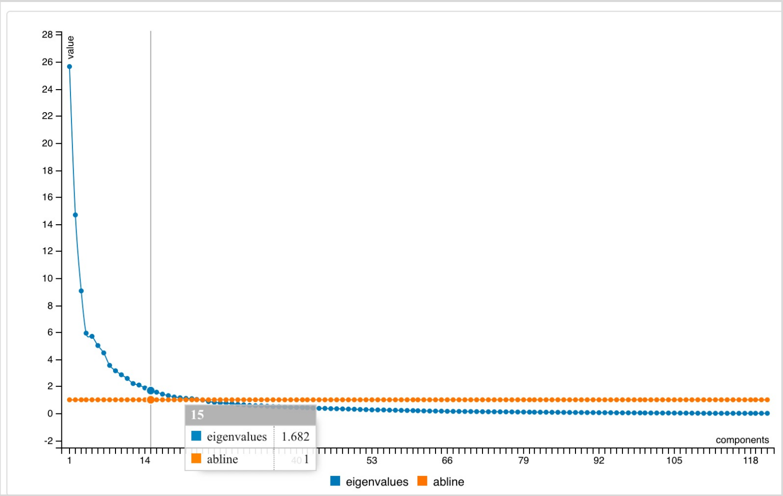
Large amount of data variables can be reduced by unsupervised learning methods

Approach: Identification of key data descriptors though dimensionality reduction

Number of derived factors: 15 factors (with highest eigenvalues) were chosen to achieve 75% of cumulative variance explanation across all variables

Variance_Explained_Table

	Eigenvalue	Pct of explained variance	Cumulative pct of explained variance
Component 1	25.65	21.2	21.2
Component 2	14.67	12.13	33.32
Component 3	9.05	7.48	40.81
Component 4	5.93	4.9	45.71
Component 5	5.69	4.71	50.41
Component 6	5.01	4.14	54.55
Component 7	4.47	3.69	58.25
Component 8	3.54	2.93	61.18
Component 9	3.15	2.6	63.78
Component 10	2.85	2.36	66.13
Component 11	2.58	2.13	68.26
Component 12	2.2	1.82	70.08
Component 13	2.09	1.73	71.81
Component 14	1.88	1.55	73.36
Component 15	1.68	1.39	74.75

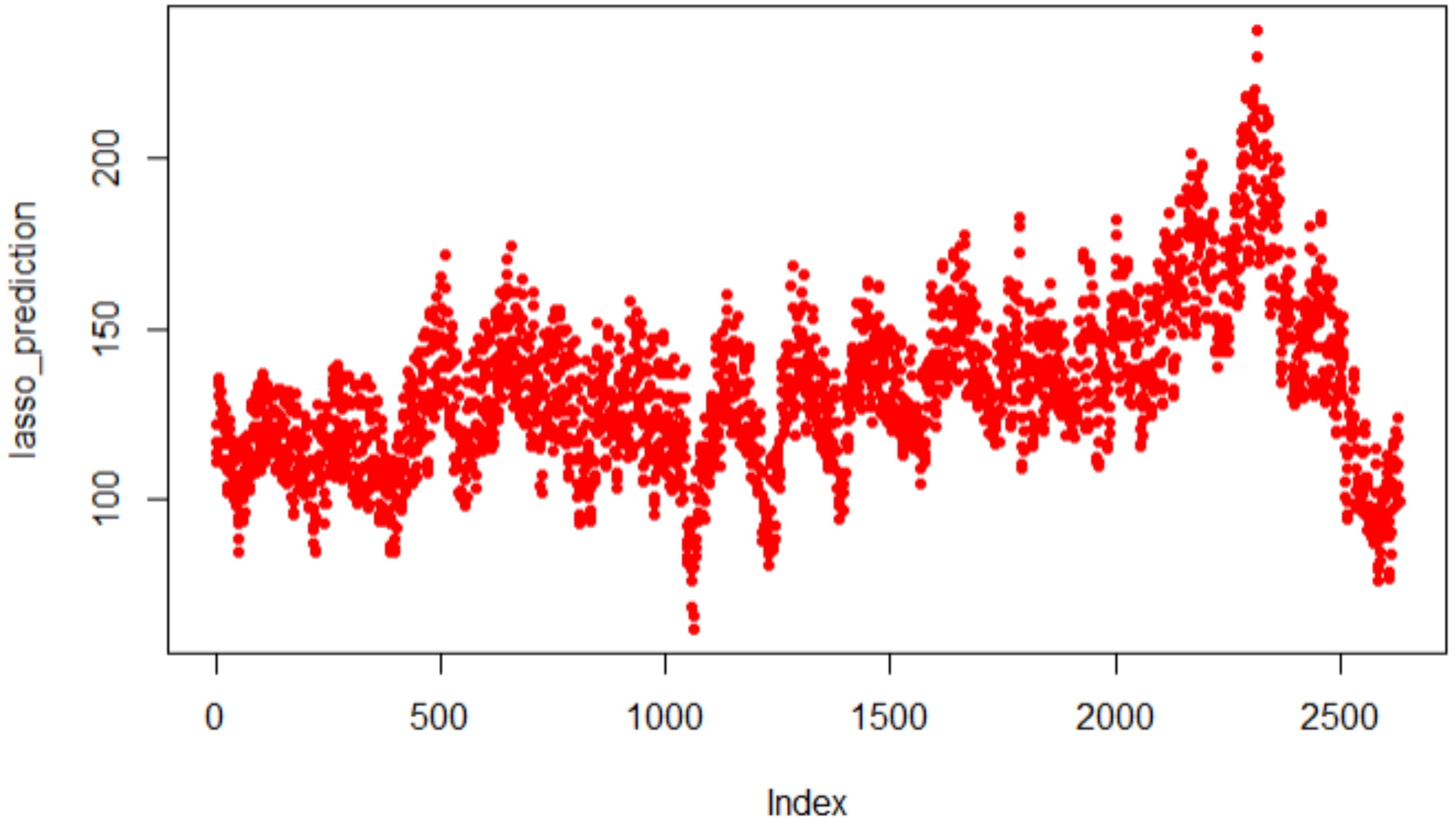


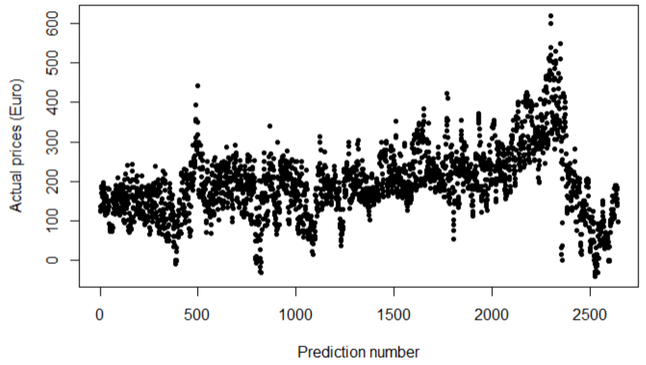
Example of factors description		
#	**Factor name**	**Key included variables**
1	Lag energy prices	Lag (1/7) electricity prices in Belgium and other countries
2	Energy consumption and fossils power	Forecast on energy consumption (ECMWF) and available fossils power n Belgium and other countries
3	Gas and oil prices	Futures gas and oil prices in eur and usd

Results: The usage of the derived factors in supervised learning model (Lasso) gave a decrease in forecast precision (RMSE) by almost 2 times (from 51 to 92)

Potential explanation: Reduction in information outweighed the simplification of the model

Lasso using factors from unsupervised learning





Results	
RMSE	92.3
MASE	3.7

Learnings Data reduction is not always better, sometimes you lose too much information

Challenges Finding a balance between many variables and simplicity is challenging

Conclusion

Conclusion: there is a gap between school examples and real-life examples

- The models we have learned to use are too simplistic to predict something as complex as electricity prices
- EDF data scientist uses a combination of different xgboost models and trains the model every day
- Factors that cannot be quantified have a huge impact on the electricity prices, e.g. pandemic, war in Ukraine, …
- More time needed to get better results

Thank you!

Questions?