Quantmetry
Building AI with pioneers

# SWEET – Electricity Demand 4
# Statistics vs. Machine Learning

*Statistical and machine learning methods combination for improved energy consumption forecasting performance*


**Guillaume Hochard, PhD**
Lead data scientist
Time series Practice Lead
ghochard@quantmetry.com

**Luis Blanche**
Data scientist


Thessaloniki, ISF 2019
June 19th, 2019

# Major challenges for a forecast model

### Architecture
Forecasting at scale
Millions of forecast

### Data quality
Missing data
Outliers

### Weather forecast
Energy forecasts
sensitive to weather

### Models
Performance
vs. Complexity
Features engineering

### Retroaction loop
Human actions based
on the forecasts can
bias them
(eg. Historic correction
in demand forecasting,
over )

### Metrics and KPIs
Statistics vs
business KPIs

### Model lifecycle
Sensitivity to change
in the data
(non-stationarity)

### Explainable AI
Factors that impacts
the forecasts

# Major challenges for a forecast model

**Architecture**
Forecasting at scale
Millions of forecast

**Data quality**
Missing data
Outliers

**Weather forecast**
Energy forecasts
sensitive to weather

**Models**
Performance
vs. Complexity
Features engineering

Caro, E. Impact of meteorological variables in short-term electric load forecasting

**Retroaction loop**
Human actions based
on the forecasts can
bias them
(eg. Historic correction
in demand forecasting,
over )

**Metrics and KPIs**
Statistics vs
business KPIs

**Model lifecycle**
Sensitivity to change
in the data
(non-stationarity)

**Explainable AI**
Factors that impacts
the forecasts

# Major challenges for a forecast model

**Architecture**
Forecasting at scale
Millions of forecast

**Data quality**
Missing data
Outliers

**Weather forecast**
Energy forecasts
sensitive to weather

**Models**
Performance
vs. Complexity
Features engineering

**Retroaction loop**
Human actions based
on the forecasts can
bias them
(eg. Historic correction
in demand forecasting,
over )

**Metrics and KPIs**
Statistics vs
business KPIs

**Model lifecycle**
Sensitivity to change
in the data
(non-stationarity)

**Explainable AI**
Factors that impacts
the forecasts

Wong L., Error metrics for Time Series Forecasting

# Major challenges for a forecast model

### Architecture
Forecasting at scale
Millions of forecast

### Data quality
Missing data
Outliers

### Weather forecast
Energy forecasts
sensitive to weather

### Models
Performance
vs. Complexity
Features engineering

## Our main focus here

### Retroaction loop
Human actions based
on the forecasts can
bias them
(eg. Historic correction
in demand forecasting,
over )

### Metrics and KPIs
Statistics vs
business KPIs

### Model lifecycle
Sensitivity to change
in the data
(non-stationarity)

### Explainable AI
Factors that impacts
the forecasts

# Use case: Challenge EDF x Quantmetry – Energy consumption forecast

## Context

Forecasting the electric consumption on the island of Ushant, France

Challenge: Feb-April 2018

# Use case: Challenge EDF x Quantmetry – Energy consumption forecast

**Context**
Forecasting the electric consumption on the island of Ushant, France
Challenge: Feb-April 2018

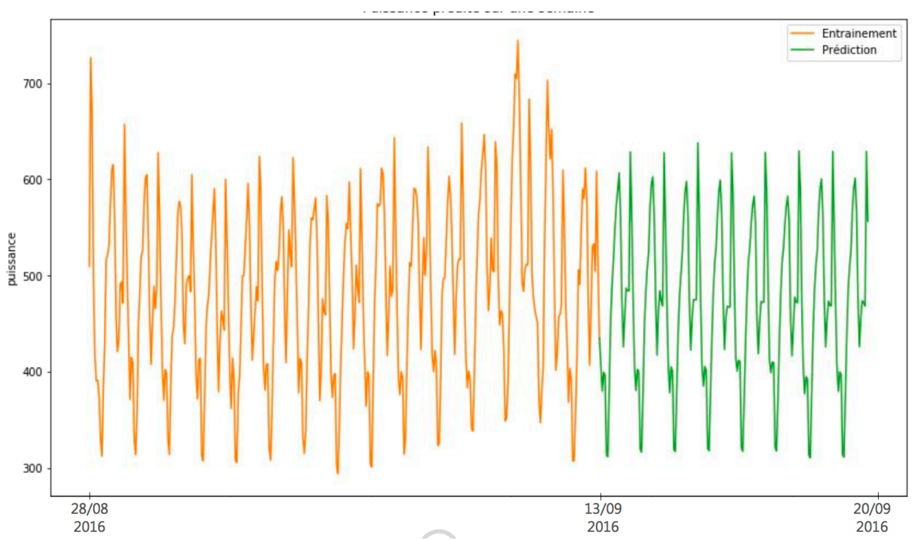
It looks like the Maldives…
but in much more colder ☺

# Use case: Challenge EDF x Quantmetry – Energy consumption forecast

**Context**
Forecasting the electric consumption on the island of Ushant, France
Challenge: Feb-April 2018

**Data**
Electric consumption: 1 time series on one year of hourly data

Meteorological time series : 11 3-hourly time series over a year



# Use case: Challenge EDF x Quantmetry – Energy consumption forecast

**Context**
Forecasting the electric consumption on the island of Ushant, France
Challenge: Feb-April 2018

**Data**
Electric consumption: 1 time series on one year of hourly data

Meteorological time series : 11 3-hourly time series over a year

Atmospheric Pressure

Windspeed

Wind direction

Temperature

Nebulosity

…

# Use case: Challenge EDF x Quantmetry – Energy consumption forecast

**Context**
Forecasting the electric consumption on the island of Ushant, France
Challenge: Feb-April 2018

**Data**
Electric consumption: 1 time series on one year of hourly data

Meteorological time series : 11 3-hourly time series over a year

**Objective**
Predict the energy consumption of the island at the hourly level for the next week

Univariate, point multi-step forecast horizon (24*7 days = 168 steps)

**Evaluation**

- Mean Absolute Percentage Error : 50%

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

  - $A_t$ = Actual Value
  - $F_t$ = Forecast Value

- Scientific methodology and quality: 20%
- Clarity, presentation : 20%
- Innovation : 10%

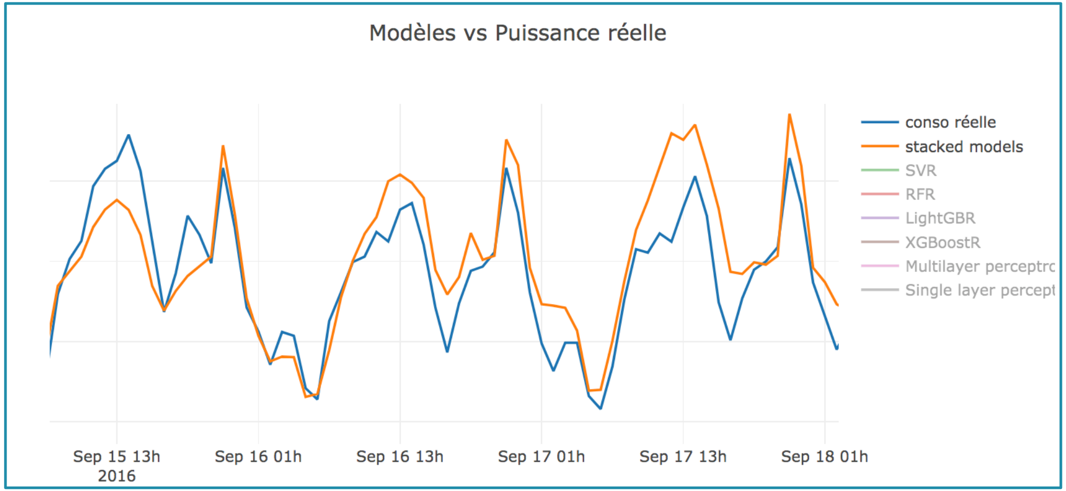
# Our entry benchmark – Machine Learning Driven

## Baseline created with ML standard models

Motivation

- Encourage participants to use  statistical methods to challenge this ML benchmark
- Encourage the combination/stacking of different methods

| Model | MAPE |
| --- | --- |
| Support Vector Regression | 13% |
| Multi Layer Perceptron | 10% |
| LightGBM | 6.5% |
| Random Forest Regressor | 6.3% |
| XGBoost | 6% |
| Stacked with ElasticNet | 5.9% |



# Our entry benchmark – Machine Learning Driven

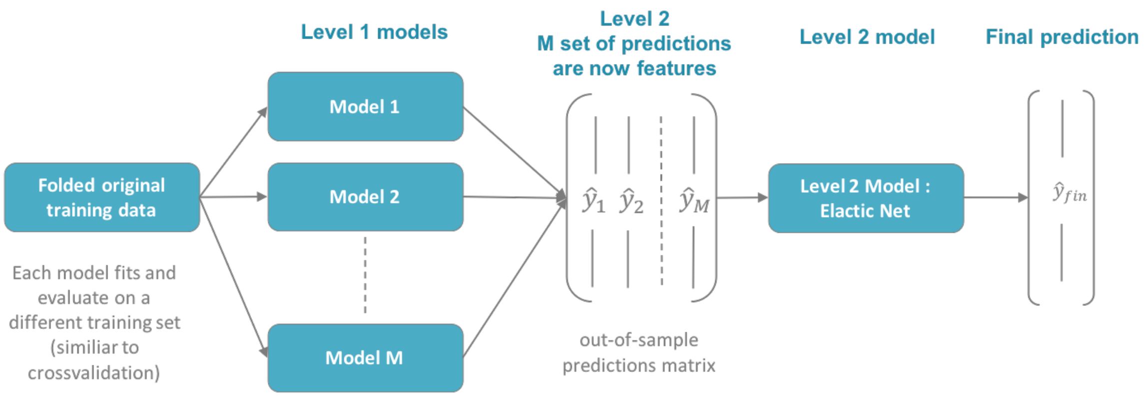
## Baseline created with ML standard models

Motivation

- Encourage participants to use  statistical methods to challenge this ML benchmark
- Encourage the combination/stacking of different methods

| Model | MAPE |
|---|---|
| Support Vector Regression | 13% |
| Multi Layer Perceptron | 10% |
| LightGBM | 6.5% |
| Random Forest Regressor | 6.3% |
| XGBoost | 6% |
| Stacked with ElasticNet | 5.9% |

## Elastic net (LASSO + Ridge regression) stacking



# Leaderboard of the competition

| Team / Method | Rank | MAPE |
|---|---|---|
| Challenge Baseline | | 5.9% |
| ARMA + GBM | 1 | 5.3% |
| Profile clustering | 3 | 3.8% |
| ExtraTreeRegressor (ARIMA for missing values) | 6 | 4 % |
| 2 GBM (ensemble) | 13 | 9.5% |
| Stacking of 6 GBM + 20 ExtraTreesRegressor | 18 | 19.6% |
| LightGBM | 20 | 22.4% |

## First insights

Performance vs rank : remember the evaluation grid (performance 50%)

Statistical methods and ML were combined and show interesting results

# Leaderboard of the competition

| Team / Method | Rank | MAPE |
|---|---|---|
| Challenge Baseline | | 5.9% |
| ARMA + GBM | 1 | 5.3% |
| Profile clustering | 3 | 3.8% |
| ExtraTreeRegressor (ARIMA for missing values) | 6 | 4 % |
| 2 GBM (ensemble) | 13 | 9.5% |
| Stacking of 6 GBM + 20 ExtraTreesRegressor | 18 | 19.6% |
| LightGBM | 20 | 22.4% |

## Performance analysis

What makes a team perform better than another?

What are the choices that helped to improve performance?

ML based methods perform worse but not always => why?

## Identify the best pratices on:

- Data cleaning
- Feature engineering
- Models

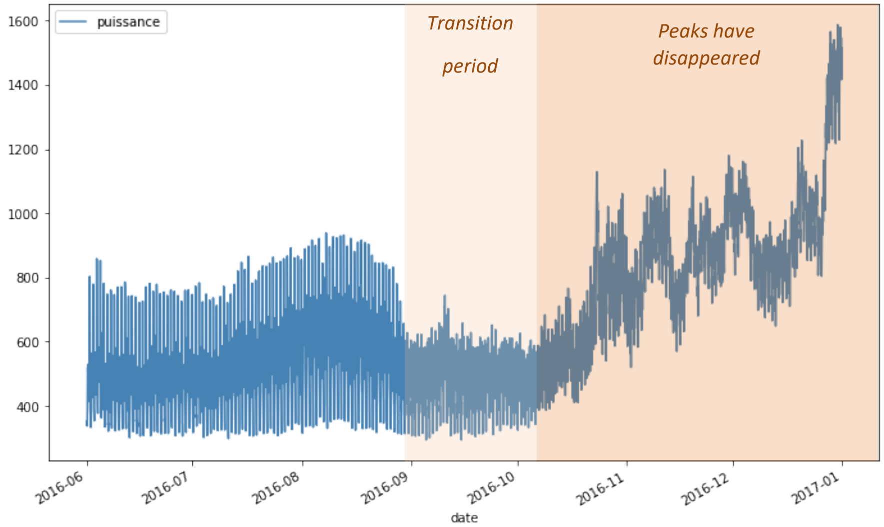
Unlike M4, we do not compare models performance but global approaches performance

# Data preparation / cleaning best practices

Data cleaning | Feature engineering and selection | Models and combinations

## 1. Graphing and data exploration

Power consumption time series and unexplained peak values




### Problem
Some peaks are observed and then disappear

### Exploration
No feature to explain the peaks +
No feature to explain the transition/no peak periods

### Why?
Network reconfiguration ?
Measurement protocol?

### Solution
Restrict the dataset to the last period
Transform the series to make it stationary

# Data preparation / cleaning best practices

Data cleaning → Feature engineering and selection → Models and combinations

1. Graphing and data exploration
2. Temporal realignement of time series

### Problem

Power: hourly time series
Meteorological : 3-hours TS

### Numerical features

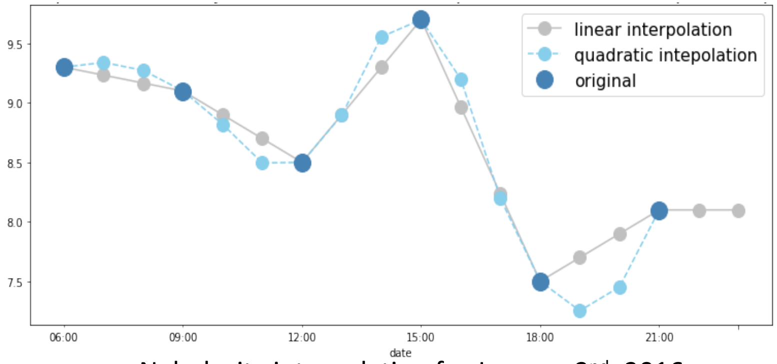
Lagrange interpolation

$$L(X) = \sum_{j=0}^{n} y_j \left( \prod_{i=0, i \neq j}^{n} \frac{X - x_i}{x_j - x_i} \right)$$

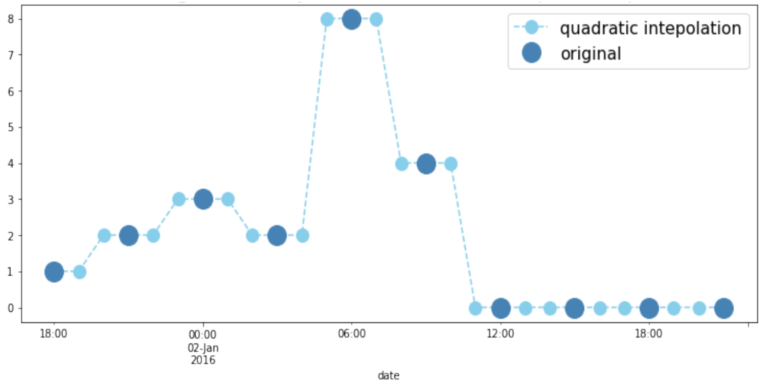
### Categorical features

K-NN interpolation

Temperatures interpolation for January 2nd, 2016




Nebulosity interpolation for January 2nd, 2016



# Data preparation / cleaning best practices

Data cleaning → Feature engineering and selection → Models and combinations

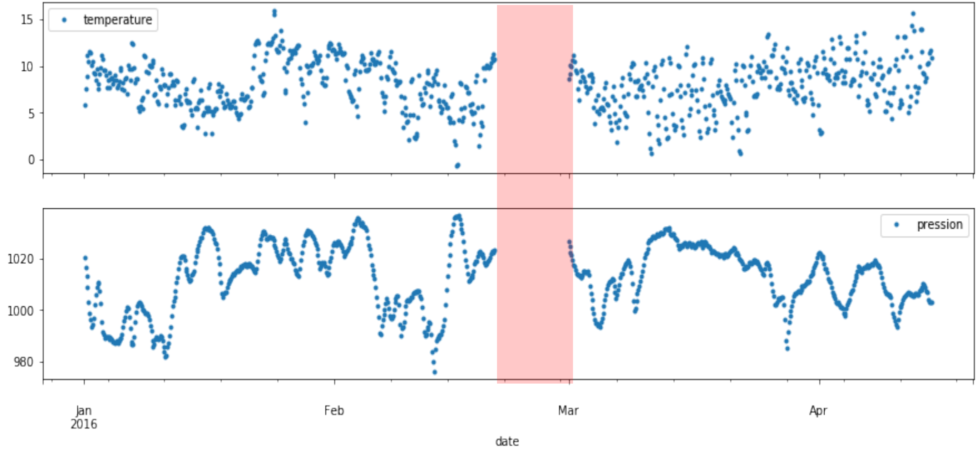
1. Graphing and data exploration
2. Temporal realignement of time series
3. Handling missing data

### Imputation

- Multiple Imputation by Chained Equations (MICE)
- MissForest

- ARIMA to forecast missing values (method ranked #6)

Missing data in temperature and pressure



# Feature engineering best practices

0. Data cleaning
Feature engineering and selection
Models and combinations

1. Timestamp decompostion (hour/day/month)

# Feature engineering best practices

Data cleaning → Feature engineering and selection → Models and combinations

1. Timestamp decompostion (hour/day/month)
2. Encoding

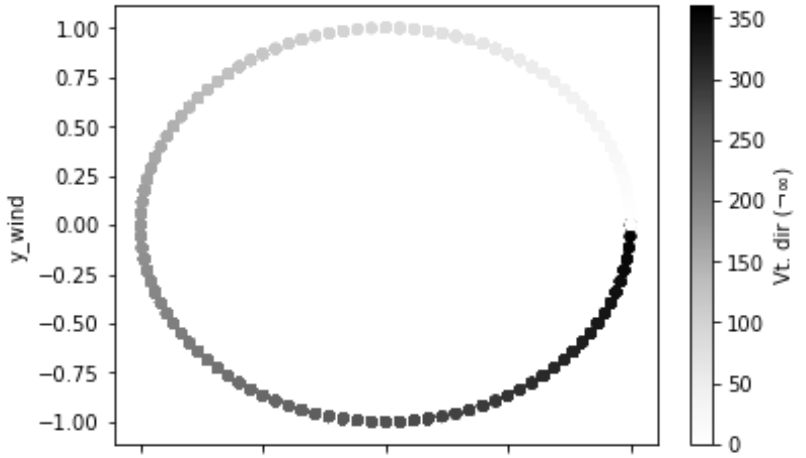
Circular features: cosine/sine transform

- Hour : 11pm is close to 1am
- Wind direction : 359° is close to 1°

Categorical features with high dimensionality

Target encoding

Cosine/sine encoding of wind direction



# Feature engineering best practices

0. Data cleaning → Feature engineering and selection → Models and combinations

1. Timestamp decompostion (hour/day/month)
2. Encoding
3. Seasonality and lags

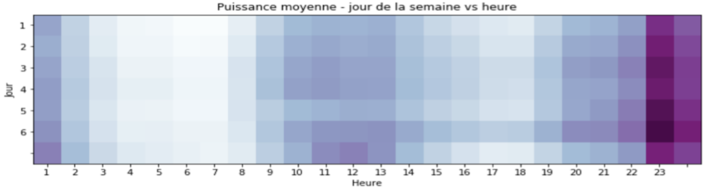
Power

24, 48, 72 hours lags
1, 2, 3, 4 weeks lags

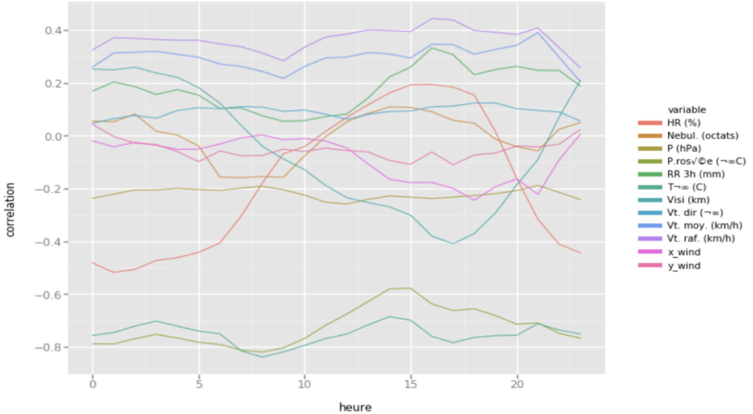
Temperature

3, 6, 12, 24 hours lag

Seasonality study




Correlations and search for the best-lag



# Feature engineering best practices

0. Data cleaning → Feature engineering and selection → Models and combinations

1. Timestamp decompostion (hour/day/month)
2. Encoding
3. Seasonality and lags
4. Calendar events and lags

- Tourism
- Week-end
- Day before holiday
- Holiday
- Day after holiday
- Summer holiday zones
- …

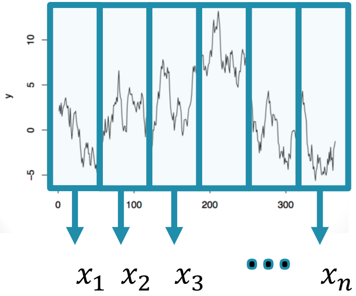
# Feature engineering best practices

0. Data cleaning → **Feature engineering and selection** → Models and combinations

1. Timestamp decompostion (hour/day/month)
2. Encoding
3. Seasonality and lags
4. Calendar events and lags
5. Time series Features

**Sliding window statistics**

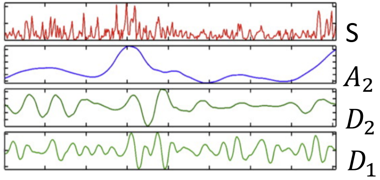
Moments
Interquartile range
Zero-Crossing rate ….




**Waveform decomposition**

Fourier/Wavelet Transform

Keep only the highest coefficients



# Feature engineering best practices

0. Data cleaning → Feature engineering and selection → Models and combinations

1. Timestamp decompostion (hour/day/month)
2. Encoding
3. Seasonality and lags
4. Calendar events and lags
5. Time series Features
6. Feature creation

- Felt cold

$$T_{felt} = (A - T) * \sqrt{W}$$

- Ratios between features

(works well with GBM/tree methods in particular)

# Feature engineering best practices

0. Data cleaning → Feature engineering and selection → Models and combinations

1. Timestamp decompostion (hour/day/month)
2. Encoding
3. Seasonality and lags
4. Calendar events and lags
5. Time series Features
6. Feature creation
7. Open data integration

- Solar Irradiance

http://www.soda-pro.com/web-services/radiation/cams-radiation-service

- Sea tide levels

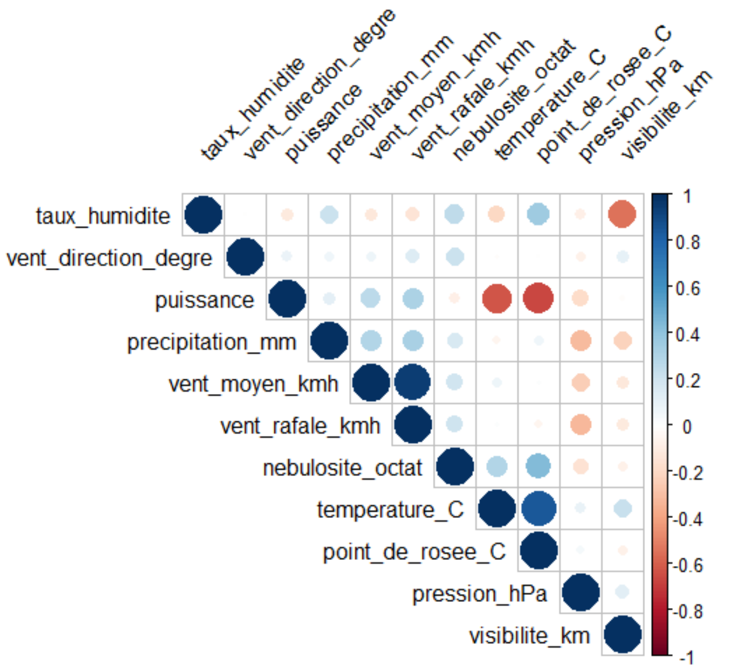
https://uhslc.soest.hawaii.edu/data/?rq

# Feature engineering best practices

0. Data cleaning → Feature engineering and selection → Models and combinations

1. Timestamp decompostion (hour/day/month)
2. Encoding
3. Seasonality and lags
4. Calendar events and lags
5. Time series Features
6. Feature creation
7. Open data integration
8. Feature selection

- Corelation Matrix
- Lasso regression
- Stepwise regression
- Train a random Forest and check for feature importances
- Try and iterate manual process



# Feature engineering best practices

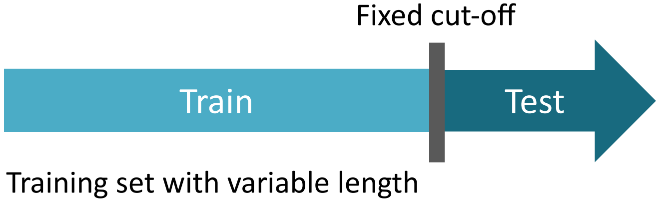
0. Data cleaning
Feature engineering and selection
Models and combinations

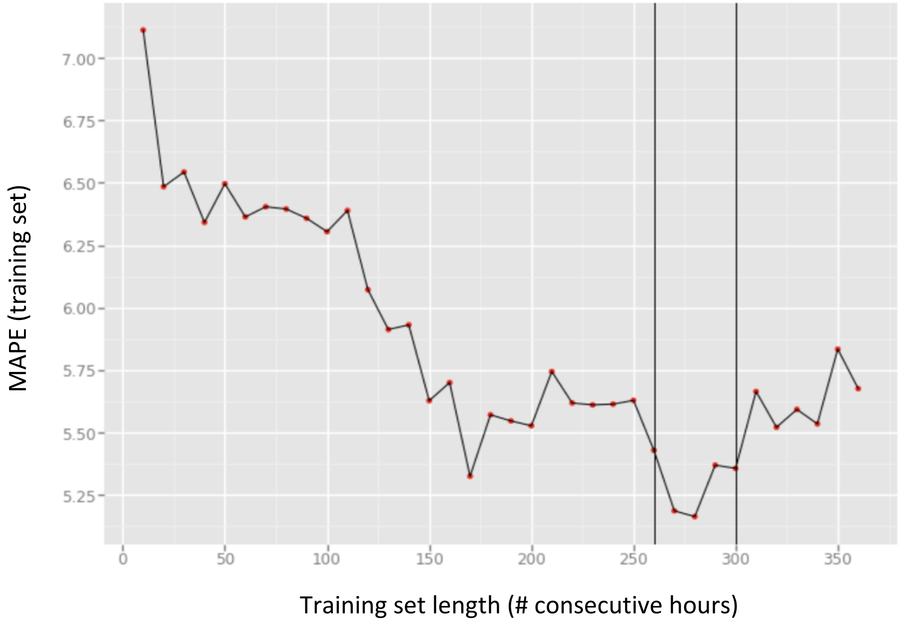
1. Timestamp decompostion (hour/day/month)
2. Encoding
3. Seasonality and lags
4. Calendar events and lags
5. Time series Features
6. Feature creation
7. Open data integration
8. Feature selection
9. Log/ Box-Cox transform of the target

# Model training: shall I take all the data available?

Data cleaning → Feature engineering and selection → **Models and combinations**

## Optimal training set size look out




### Depends on the forecast horizon

Short term or long term patterns to model

### Tradeoff for performance optimization

More representative/less noisy training data vs the number of samples to train on

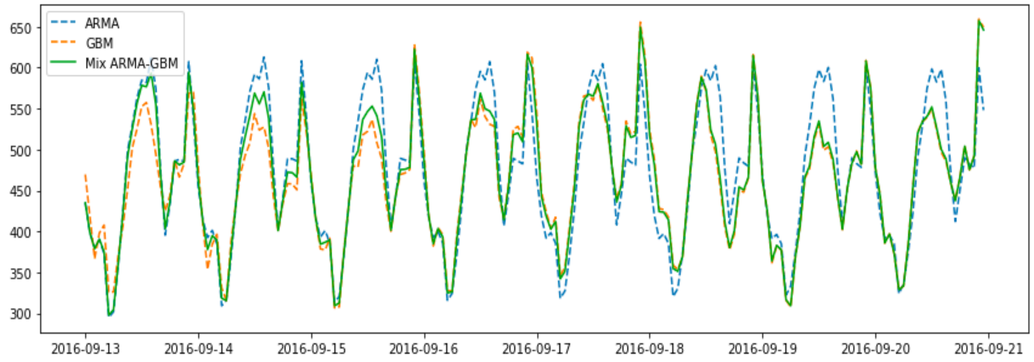
# Model of the winning method

Data cleaning → Feature engineering and selection → **Models and combinations**

## Statistics and ML together

- Hybrid combination  between ARMA and a Gradient Boosting model
- MAPE: 5.3%

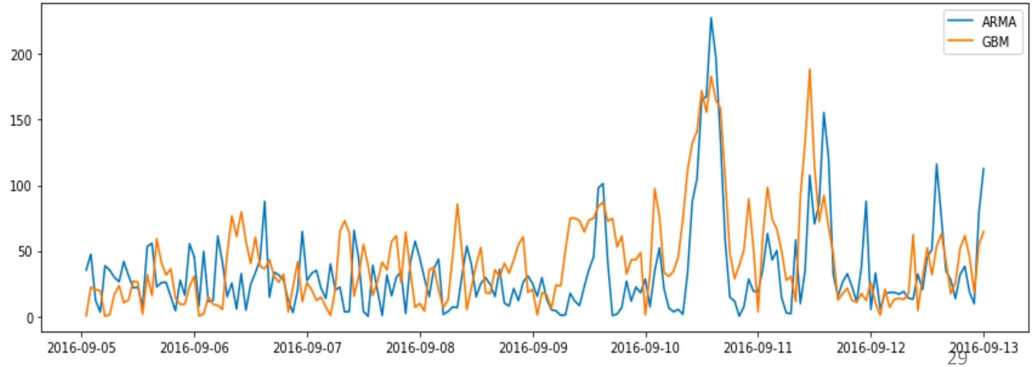
Intuition : ARMA is better at the begining of the forecasting horizon




## Combination details

- $y_{pred} = \alpha(t) y_{pred,ARMA} + (1 - \alpha(t)) y_{pred,GB}$
- $\alpha(t) = e^{-\frac{t}{\lambda}}$

$\lambda$ is solved by optimization



# Model of the winning method

Data cleaning → Feature engineering and selection → **Models and combinations**

### Statistics and ML together

- Hybrid combination  between ARMA and a Gradient Boosting model

- MAPE: 5.3%

Intuition : ARMA is better at the begining of the forecasting horizon

### Combination details

- $y_{pred} = \alpha(t) y_{pred,ARMA} + \big(1 - \alpha(t)\big) y_{pred,GB}$
- $\alpha(t) = e^{-\frac{t}{\lambda}}$

$\lambda$ is solved by optimization

```python
from scipy.optimize import minimize

def mix_pred(param, y_arma, y_gb, linear):
    if linear:
        alpha = param
    else:
        x = np.linspace(0, 1, len(y_arma))
        alpha = np.exp(-x / param)
    y_mix = (alpha * y_arma + (1-alpha)*y_gb)
    return y_mix

def find_optimal_mix(param, y_true, y_arma, y_gb, linear=True):
    y_mix = mix_pred(param, y_arma, y_gb, linear)
    mape = mean_absolute_percentage_error(y_true, y_mix)
    return mape

is_linear = False
result = minimize(find_optimal_mix, 0.5,
                  args=(local_test['y'], local_test['y_hat_arma'], y_pred, is_linear))
slope = result.x

y_mix = mix_pred(slope, local_test['y_hat_arma'], y_pred, is_linear)
print('Best param: %.2f' % (slope))
print('MAPE: %.2f%%' % (result.fun))
```

# More ideas on ML and statistical models combination

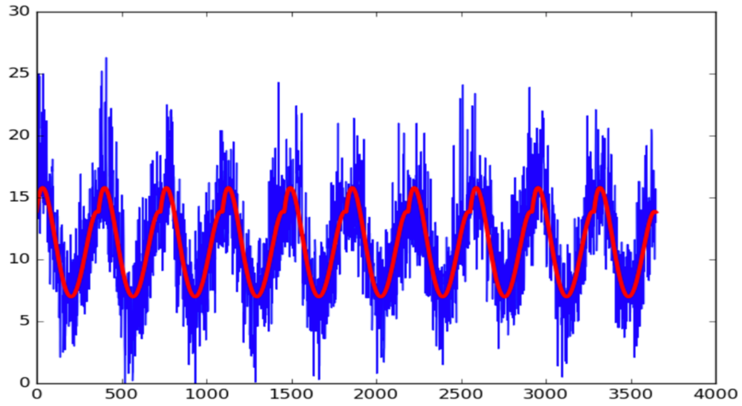
## Statistics and ML together: more ideas

## Additive model

$$P = Ps + Pm + N$$

- Seasonality part: $Ps$
- Part sensitive to external regressors: $Pm$
- Noise: $N$

**Statistical model to predict $Ps$** (seasonality component)




**Machine learning model**

- **Prediction of the residual component $Pm$** (time series without $Ps$).
- Impacted by meteorological, calendar (holidays, events) features
- ML models (RF, GB, XGBoost, etc.) ou DL (MLP, LSTM, CNN)

# More ideas on ML and statistical models combination

## Statistics and ML together: more ideas

### Additive model

$$P = Ps + Pm + N$$

- Seasonality part: $Ps$
- Part sensitive to external regressors: $Pm$
- Noise: $N$

### Parametrization of statistical models

Neural network to estimate ARMA parameters

Callot L., On the paramatrization of simple autogressive models with neural networks

# Key learnings of this competition

## Data Preparation & Feature Engineering

- Diagnostic by visualisation: know your data
- Missing values: different strategies w.r.t. data types and the data itself
- Feature engineering and open data help !!! (don't rush to the model too soon)

## Models and methods

- Our pure ML benchmark was beaten by hybrid but also pure ML methods
- Hybrid methods makes better predictions (as in M4 results)

- Statistical and ML methods are complementary
  Profi M., ISF 2019: Combinations of ML and traditional approches to increase accuracy in forecasting

- Model is only one part of the performance improvement journey
- Good performance = clean data, rich data, smart feature engineering and selection, model combination

## Make your own challenge

- Interesting for getting new ideas
- Even with 20-30 teams (not necessary to go to Kaggle for challenges)

Quantmetry
Building AI with pionneers

# We're hiring!

ghochard@quantmetry.com

Guillaume Hochard