BAYESIAN TIME-SERIES FORECASTING: PROPHET

PROPHET

GB Non-domestic Power Demand

Time-Series Forecasting

Models	Details	Examples
ARIMA-based	Autoregressive (AR) terms {values}, differencing to achieve stationarity (I), and moving average (MA) terms {residuals}.	SARIMA, SARIMAX, GARCH, etc.
Decomposition-based	Deconstruction of time series into components: Trend, Seasonalities, Regressors, Residuals.	Prophet
Neural Networks	Use of Transformers, Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN)	Long Short-Term Memory (LSTM), Temporal Convolutional Networks (TCN)
Ensemble methods	Combines forecasts from multiple models to improve accuracy.	Mix

**Very new: Foundation Models*

Very new: Foundation Models

Why Prophet? *in Power Demand Forecasting*

1. **Open-box:** Clear input components, parameters, outputs, and optimization algorithms. Not a neural network-based model (i.e., less explainable).
   - *Trend, Daily/Weekly/Yearly Seasonality, Holidays, External regressors…*
   - *Weights, values of components are part of the output.*
2. Uses “non-linear” algorithms for seasonality– although it is optimized as a linear model (Fourier Series).
3. Bayesian statistics to harness prior-knowledge: residuals ~ normal distribution.
4. Robustness to missing data and outliers:
   - *Data missing handling + Fitting downweighs outliers + L1 loss function > minimizes absolute deviations, not squared + Bayesian inference by sampling.*

When Prophet?

- <u>ARIMA</u>: ~ *short-term forecasting.*
  - *Evolves in time: "x" past values and errors to predict "y"*
  - *Future values depend on past values in a sequential time-dependent manner.*
  - *Relationship between lags of past observation & their errors.*
  - *So future predictions errors are assumed to be 0 >>* ***increased uncertainty as forecasts extend out.***

- <u>Prophet (Decomposition)</u>: ~ *long-term forecasting.*
  - *Curve-fitting problem: regression data.*
  - *Combination of components (trend, seasonality, holidays, regressors…) by fitting curves to data.*
  - *Weights, values of components are part of the output.*

Prophet's components' coefficients are helpful inputs to short-term forecasts.

PROPHET FORECASTING

Underlying components and logic.
Using Hyperopt to optimize hyperparameter selection

**Bayesian time-series modeling*

Non-Domestic Daily Model

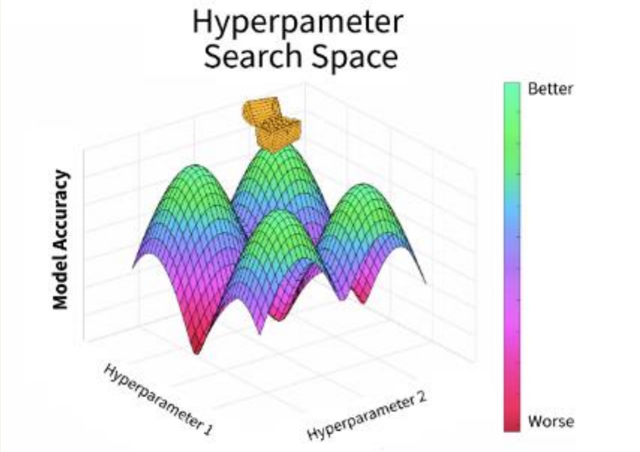
PROPHET

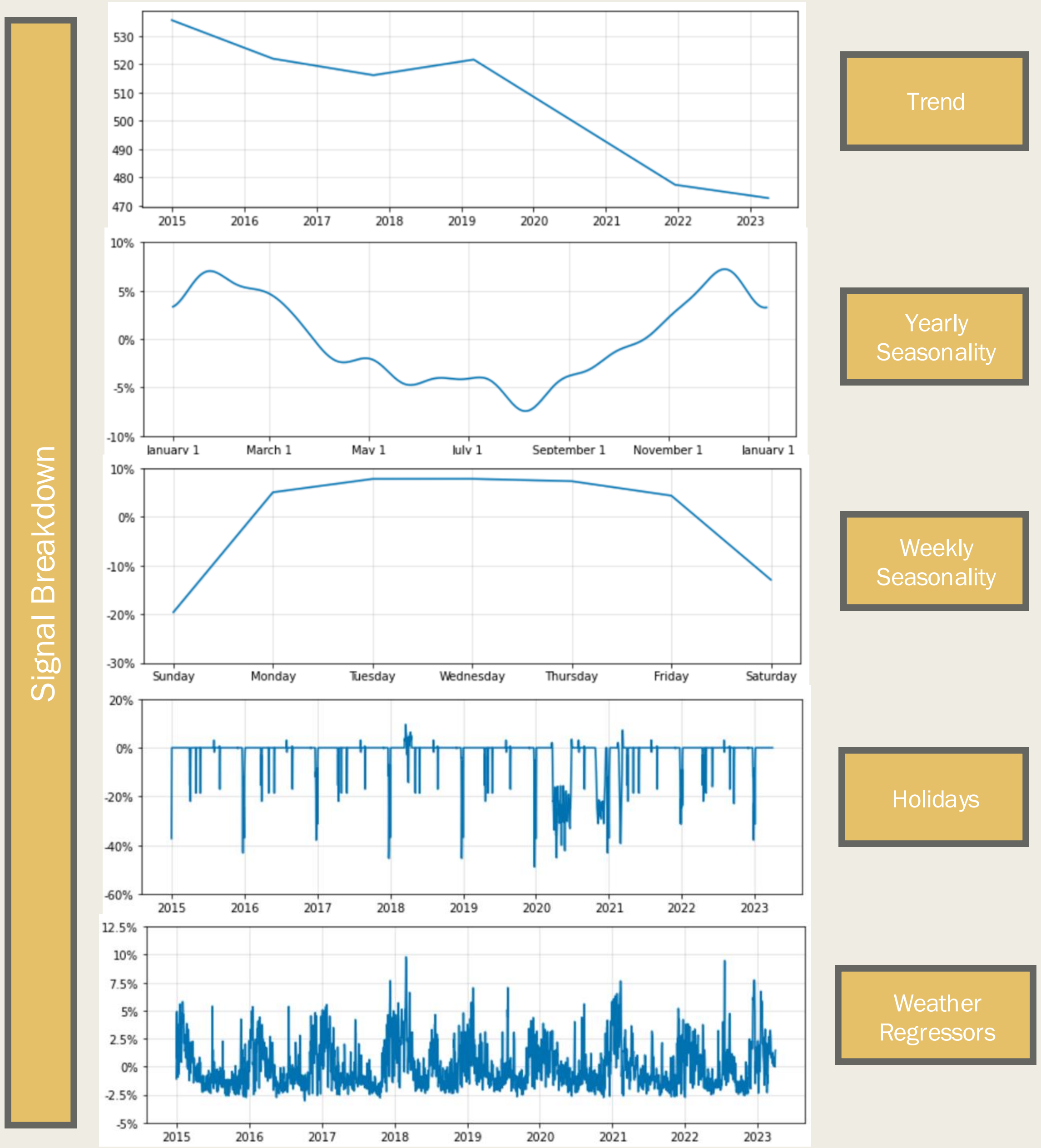
Time Series = Signal + Noise [Error]

Hyper-parameter tuning

HYPEROPT

Hyperparameters: regularization of Objective Function's coefficients.





PROPHET

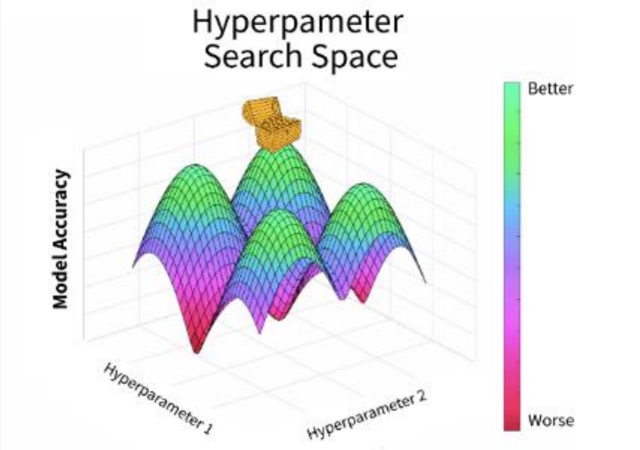
Non-Domestic Daily Model

Time Series = Signal + Noise [Error]

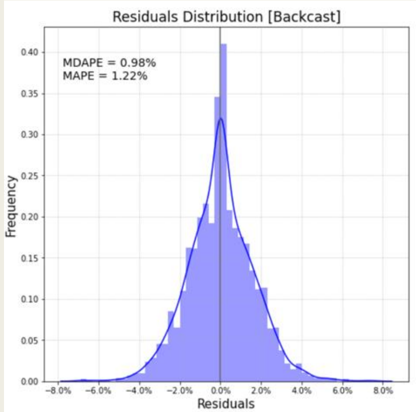
Hyper-parameter tuning

HYPEROPT

<u>Hyperparameters:</u> regularization of Objective Function’s coefficients.

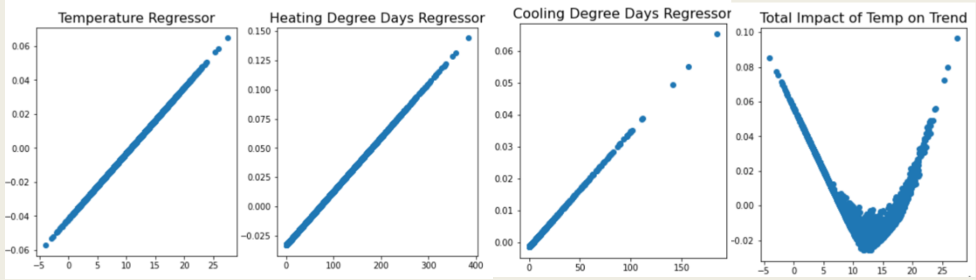


Noise [Error]

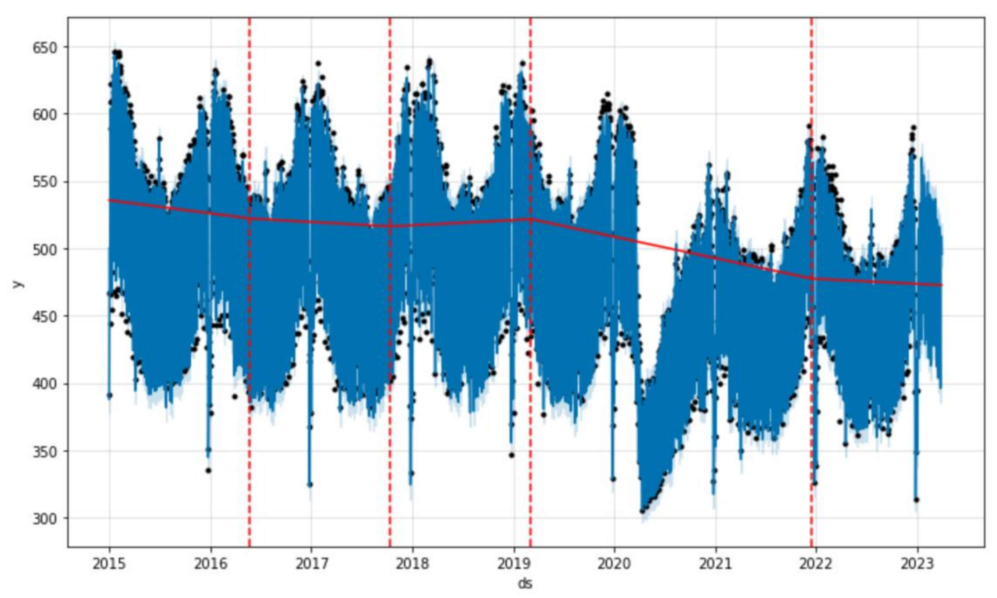


<u>Temperature Regressors</u>

Overcoming linear constraints



Forecast + Trend Changepoints



Prophet Objective Function

Run fourier_series_graphical.py
http://127.0.0.1:8050/

Forecasting – Decomposable time-series model

PROPHET

$$y(t) = g(t) * s(t) * h(t) + \epsilon_t$$

$g(t)$: *trend function [models non-periodic changes in the value of the timeseries]*
$s(t)$: *periodic changes [e.g., weekly and yearly seasonality]*
$h(t)$: *effects of holidays which occur on potentially irregular schedules*
ϵ_t : *error term → any idiosyncratic changes which are not accommodated by the model*

Trend Function

$$g(t) = (k + \boldsymbol{a}(t)^T \boldsymbol{\delta})t + (m + \boldsymbol{a}(t)^T \boldsymbol{\gamma})$$

k: *initial growth rate* [Scalar]
δ: *rate adjustments at changepoints* [Vector]
$a(t)$: *indicator vector for changepoints* [Vector]
m: *offset parameter* [Scalar]
γ_j: *offset adjustment at checkpoints*: $s_j \delta_j$ [Vector]

Adjustments at changepoints j

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l\right)\left(1 - \frac{k + \sum_{l<j} \delta_l}{k + \sum_{l \le j} \delta_l}\right)$$

Seasonality Function

$$s(t) = \sum_{n=1}^{k} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right)\right)$$

k: *number of Fourier terms*
P: *period (e.g., 365.25 for yearly seas.)* [Scalar for each n]
a_n: *fourier coefficients for cosine term* [Scalars for each n]
b_n: *fourier coefficients for sine terms* [Scalars for each n]

Example: Yearly

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{365.25}\right), \dots, \sin\left(\frac{2\pi(10)t}{365.25}\right)\right]$$

$$\boldsymbol{\beta} = [a_1, b_1, \dots, a_N, b_N]^k$$

$$s(t) = X(t)\,\boldsymbol{\beta}$$

Holidays

$$Z(t) = [1(t \in D_1), \dots, 1(t \in D_L)]$$

$$h(t) = \sum_{i=1}^{L} \kappa_i Z_i(t)$$

$$h(t) = z(t)\boldsymbol{\kappa}$$

L: *number of holidays*
$\boldsymbol{\kappa_i}$: *holiday effect coefficients* ***[Scalars for each i]***
$Z_i(t)$: *indicator function for holiday i at time t* ***[Scalars for each i]***

Same concept for the included regressors.

Prophet Objective Function

Forecasting – Decomposable time-series model

PROPHET

$$y(t) = g(t) * s(t) * h(t) + \epsilon_t$$

$g(t)$: $trend\ function\ [models\ non-periodic\ changes\ in\ the\ value\ of\ the\ timeseries]$
$s(t)$: $periodic\ changes\ [e.g., weekly\ and\ yearly\ seasonality]$
$h(t)$: $effects\ of\ holidays\ which\ occur\ on\ potentially\ irregular\ schedules$
ϵ_t : $error\ term \rightarrow any\ idiosyncratic\ changes\ which\ are\ not\ accommodated\ by\ the\ model$

Regularization + Objective Function in classic ML…

- The objective is to minimize the sum of the errors (residuals) plus the regularization penalties:

$$Objective = \sum_{t=1}^{T} |y_t - \hat{y}_t| + \lambda_\delta\ ||\delta||_2{}^2 + \lambda_a\ ||a||_2{}^2 + \lambda_b\ ||b||_2{}^2 + \lambda_\kappa\ ||\kappa||_2{}^2$$

where:
$\hat{y}(t) = g(t) * s(t) * h(t)$ *is the fitted value at time t.*

The penalties on δ, a, b, and κ discourage large coefficients, resulting in a smoother trend and seasonal components that better generalize to new data.

<u>Equivalent:</u>

- L1 loss function for more robustness to outliers because it penalizes them linearly.
- L2 (Ridge Regression) penalizes the hyperparameters quadratically.

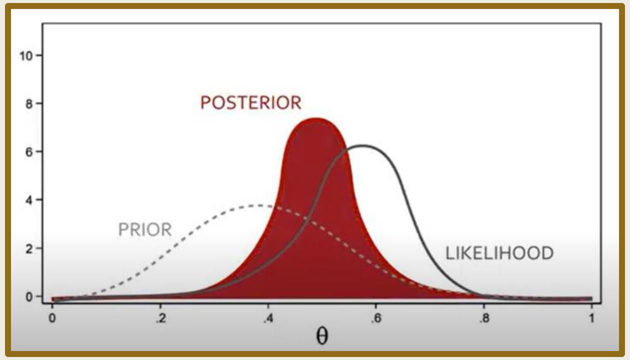
Bayesian probabilities Refreshing memory

- Represents a degree of belief or uncertainty about an event, **updated as new evidence becomes available.**
  - *“Probabilities” (beliefs) change as you learn more.*
- You need to input “prior” beliefs and data (residuals/errors follow a normal distribution).
- Make sense in time-series forecasting because:
  - *They allow for updating predictions as new data arrives.*
  - *Bayesian models provide uncertainty estimates for future values.*
  - *Handle non-stationarity.*
  - *Incorporate prior knowledge (such as past patterns or domain expertise).*

$$P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)}$$

Model Fitting, Posterior Distribution

- Most calculations use Bayesian inference.
  - *Note: We clearly want the residuals to follow a normal distribution → We might as well input and use that information.*
- Posterior Log-likelihood is used in:
  - *Optimization for MAP Estimation.*
  - *MCMC Sampling for Posterior Distribution.*
- The errors/residuals of y are normally distributed. That's our first assumption ("prior distribution").



Posterior log-likelihood

$$\log P(\theta|D) = \log P(D|\theta) + \log P(\theta) - \log P(D)$$

- θ: Model parameters (e.g., trend coefficients, seasonal effects).
- D: Observed data.
- $P(D|\theta)$: **Likelihood** of the data given the parameters.
- $P(\theta)$: **Prior** probability of the parameters.
- $P(D)$: **Marginal likelihood** of the data (a normalizing constant).

Formulating a linear regression for Bayesian Inference

$$y_n = \alpha + \beta x_n + \epsilon_n \quad \text{where} \quad \epsilon_n \sim \text{normal}(0, \sigma).$$

This is equivalent to the following sampling involving the residual,

$$y_n - (\alpha + \beta X_n) \sim \text{normal}(0, \sigma),$$

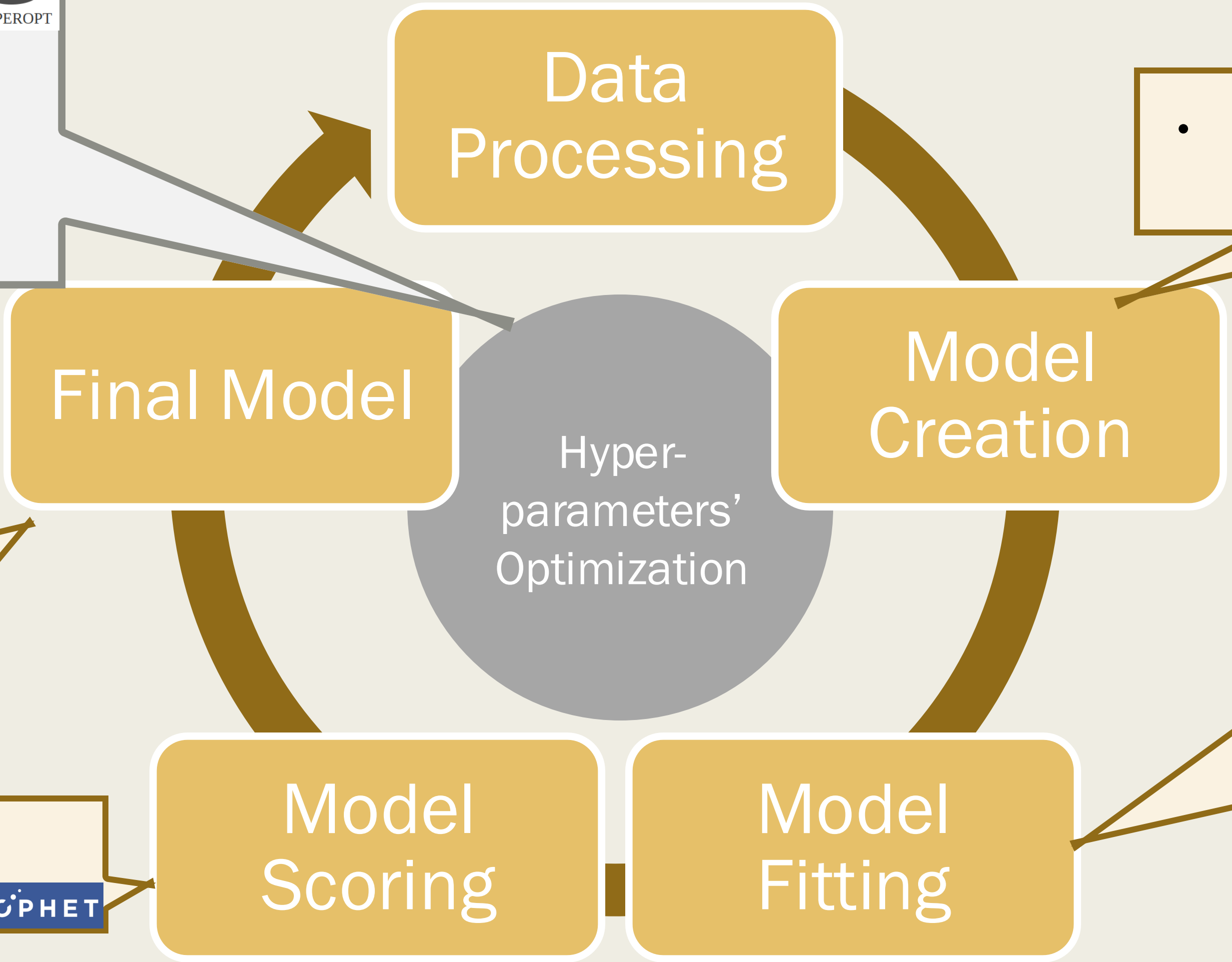
and reducing still further, to

$$y_n \sim \text{normal}(\alpha + \beta X_n, \sigma).$$

Graph of the steps taken by Prophet + HP…

Hyper-parameters tuning:
- Inputs:
  - Search space.
  - Distribution.
  - Max iterations.
- Tree-based probabilistic optimization.

- Specify all parameters/variables of the model.



- MCMC/MAP: Bayesian Inference | Optimization | Sampling (w/ regularization).
- Output: Model with parameters' coefficients, etc.

- Iterative historic forecast testing.

- Including all parameters from the best fitted model.
- When forecasting: Uncertainty samples for forecast's intervals.

PROPHET

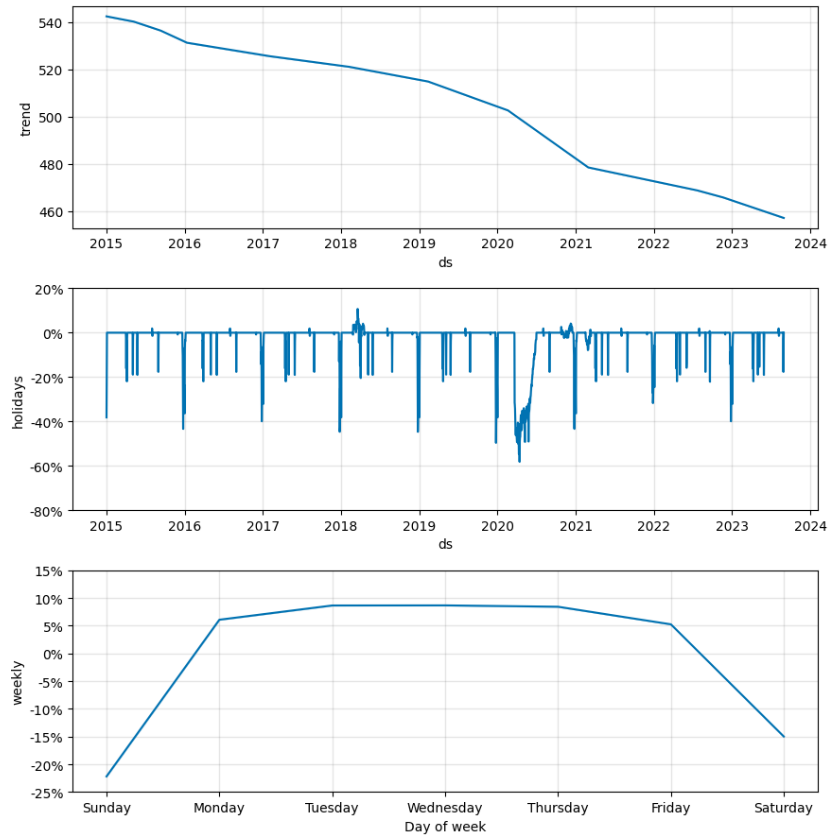
GB DAILY MODEL

Example

HYPEROPT

Main Components

Trend, Holidays, Weekly Seasonality



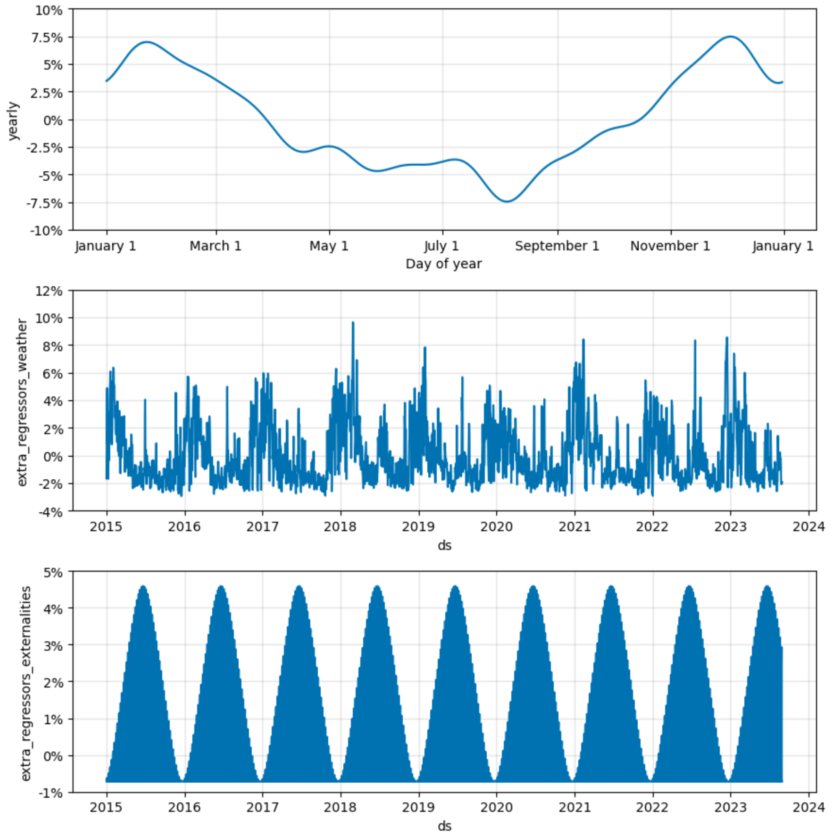
Ad-hoc implementations:

- Holidays including day-of-week sensitivity.
- Weekdays yearly-changes (by daylight seconds).
- Lockdowns' impact using "holidays".

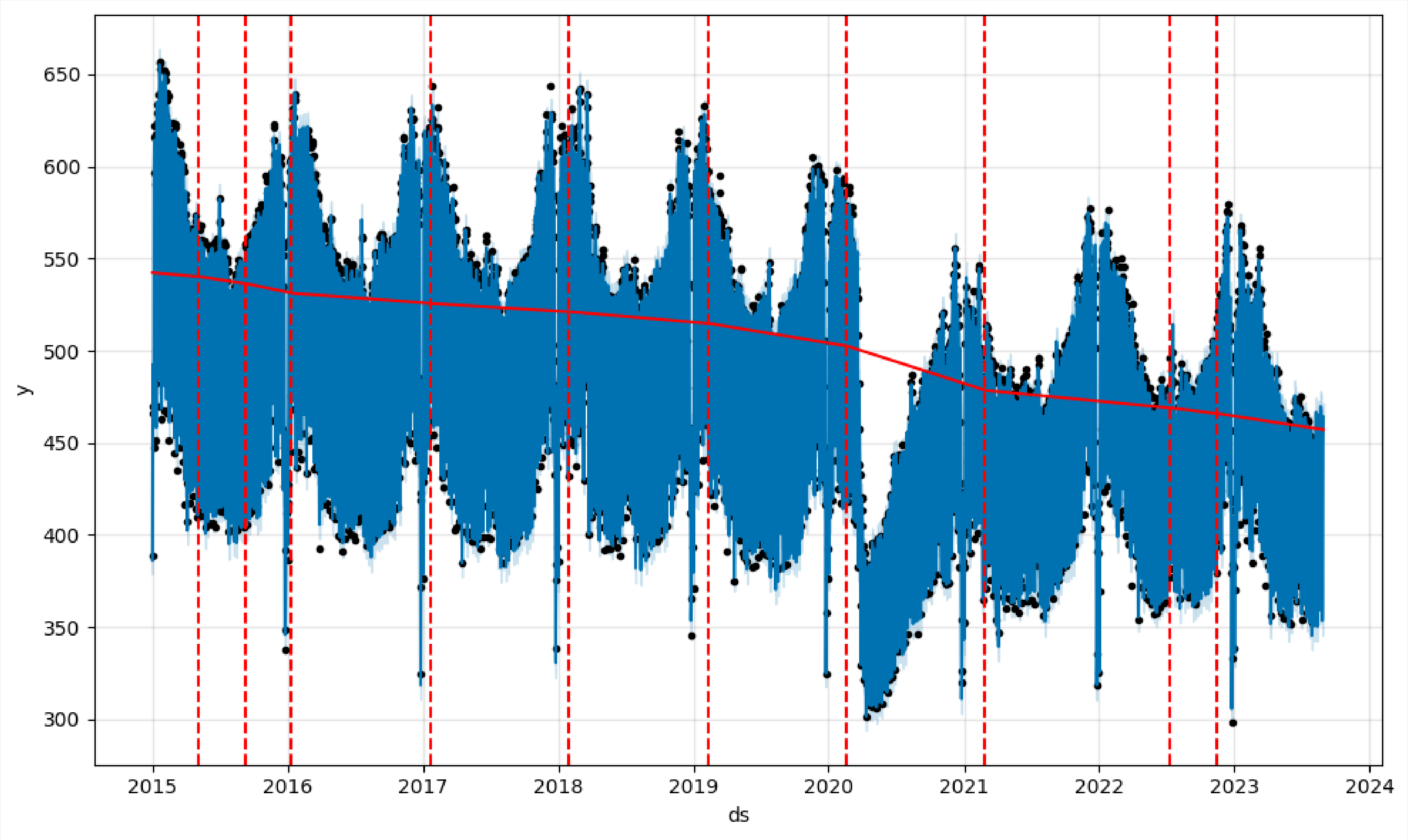
Notes:

The lag-dependencies from an ARIMA model are "captured" by components: trend, yearly Fourier series.

Yearly Seasonality, Weather reg., Weekends reg.

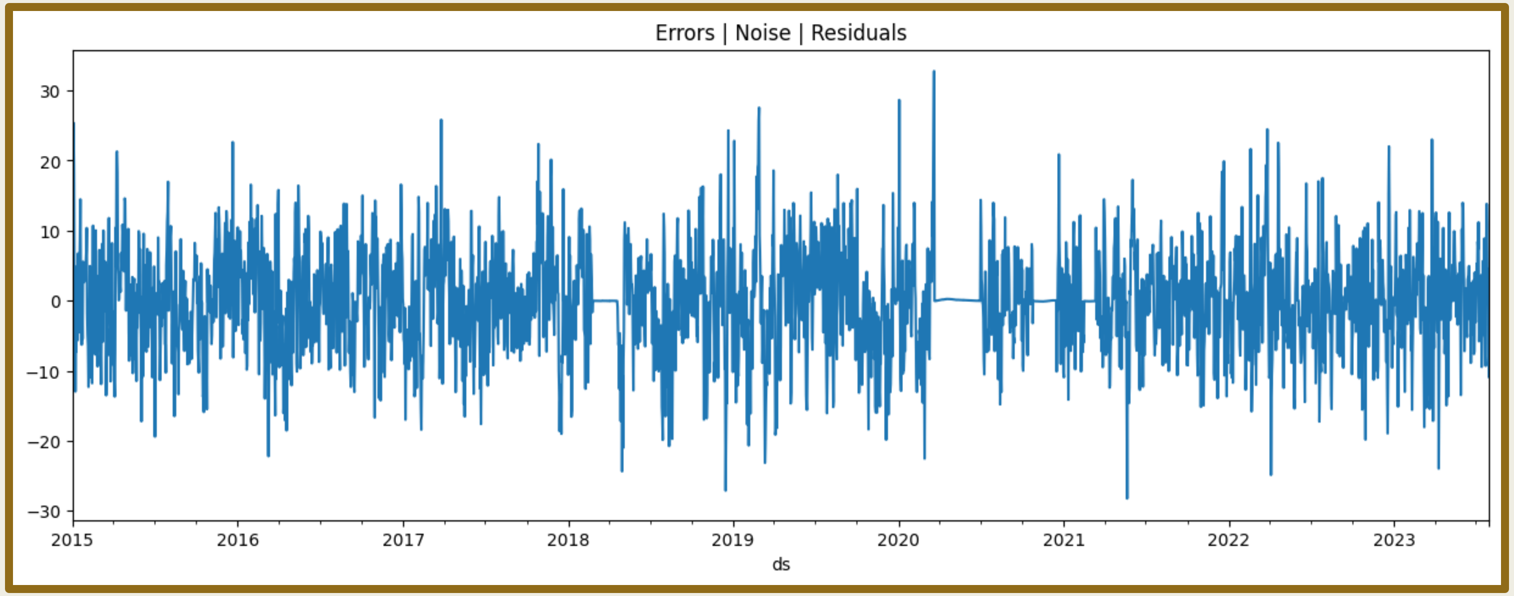


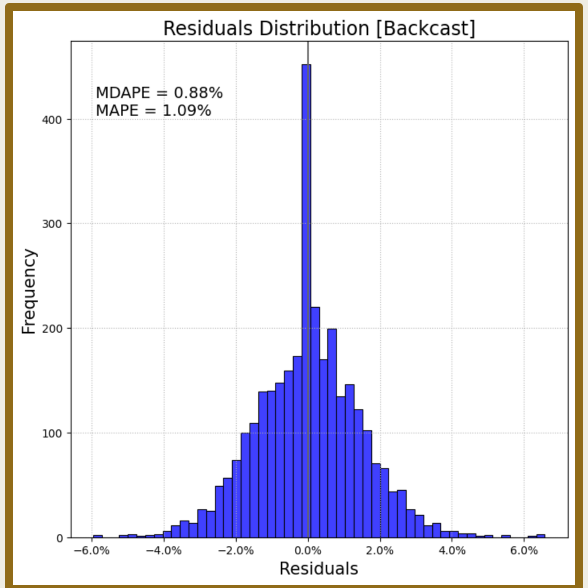
Back-cast and Forecast



Noise, Error, Residuals from back-casting

- By looking at the overall model's residuals it seems quite stationary → What about by component?
- Optimization starts by setting the residual as a normally distributed around 0.

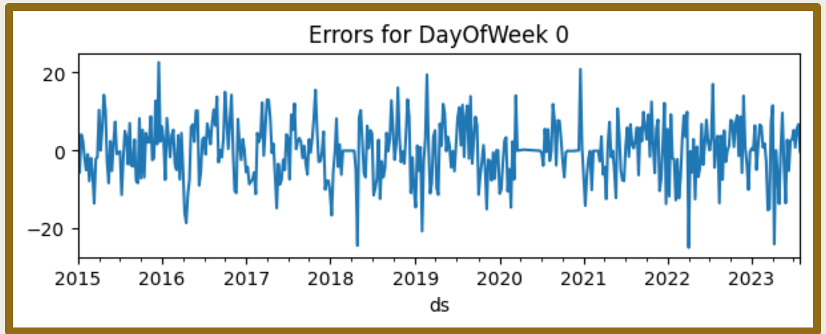


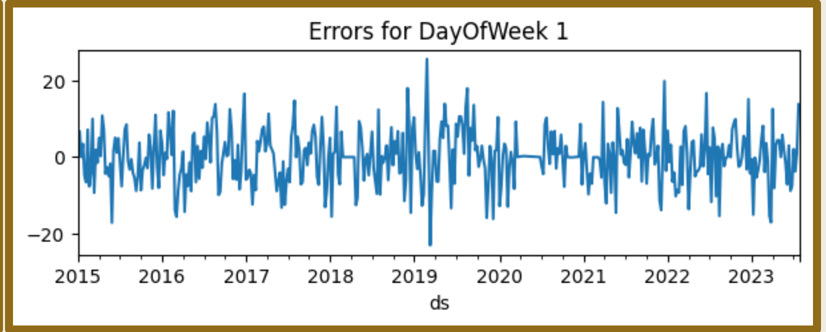


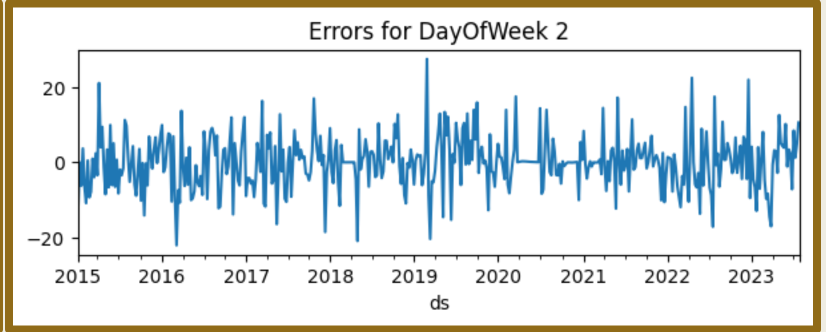
Noise by Day of Week

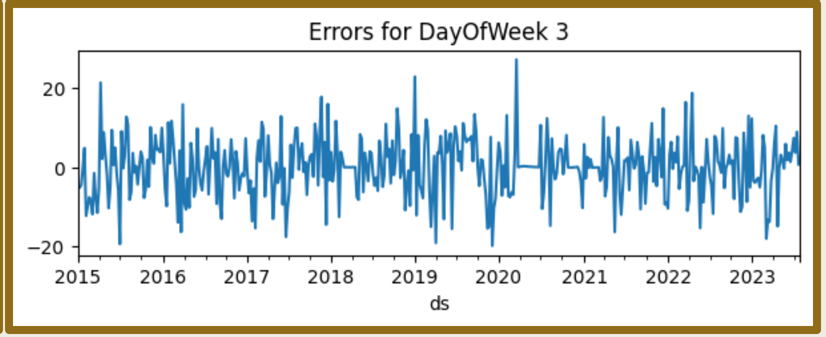
Example by graphs of how explainable signals not captured by the model can be mathematically captured by refining the model:

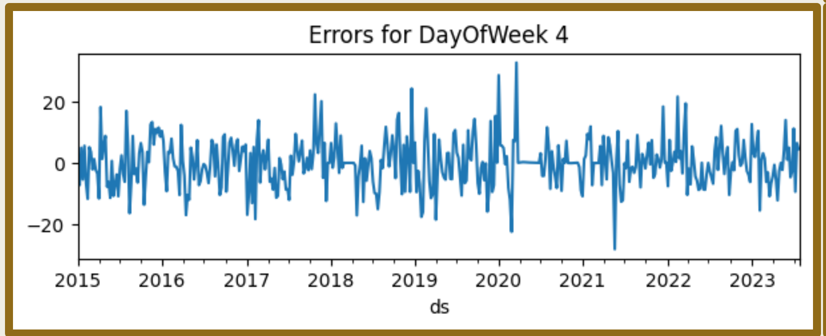
- We can see some days of the week are less stationary.
- Before adding the “daylight_hours” regressor for the weekends → noise was even more seasonal.

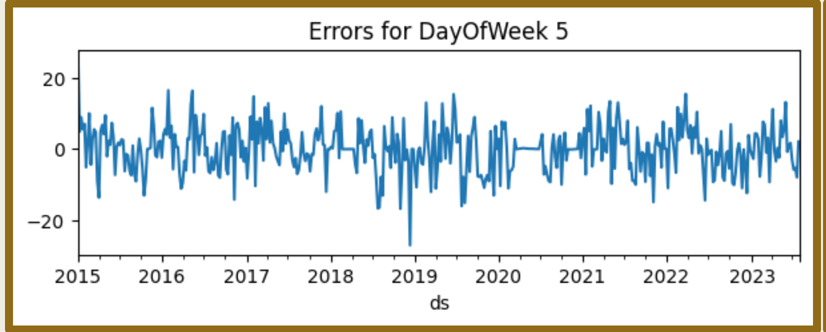


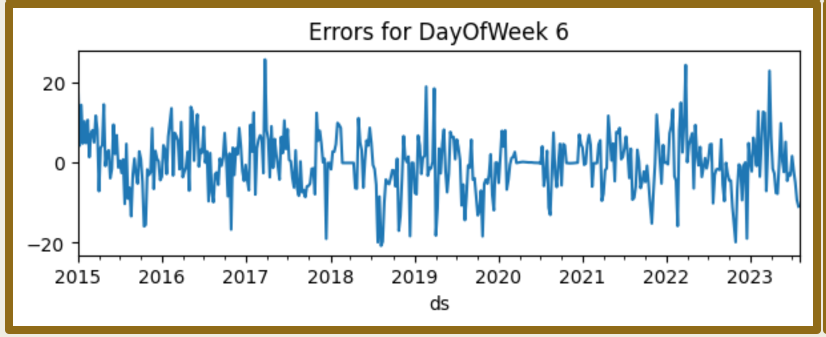










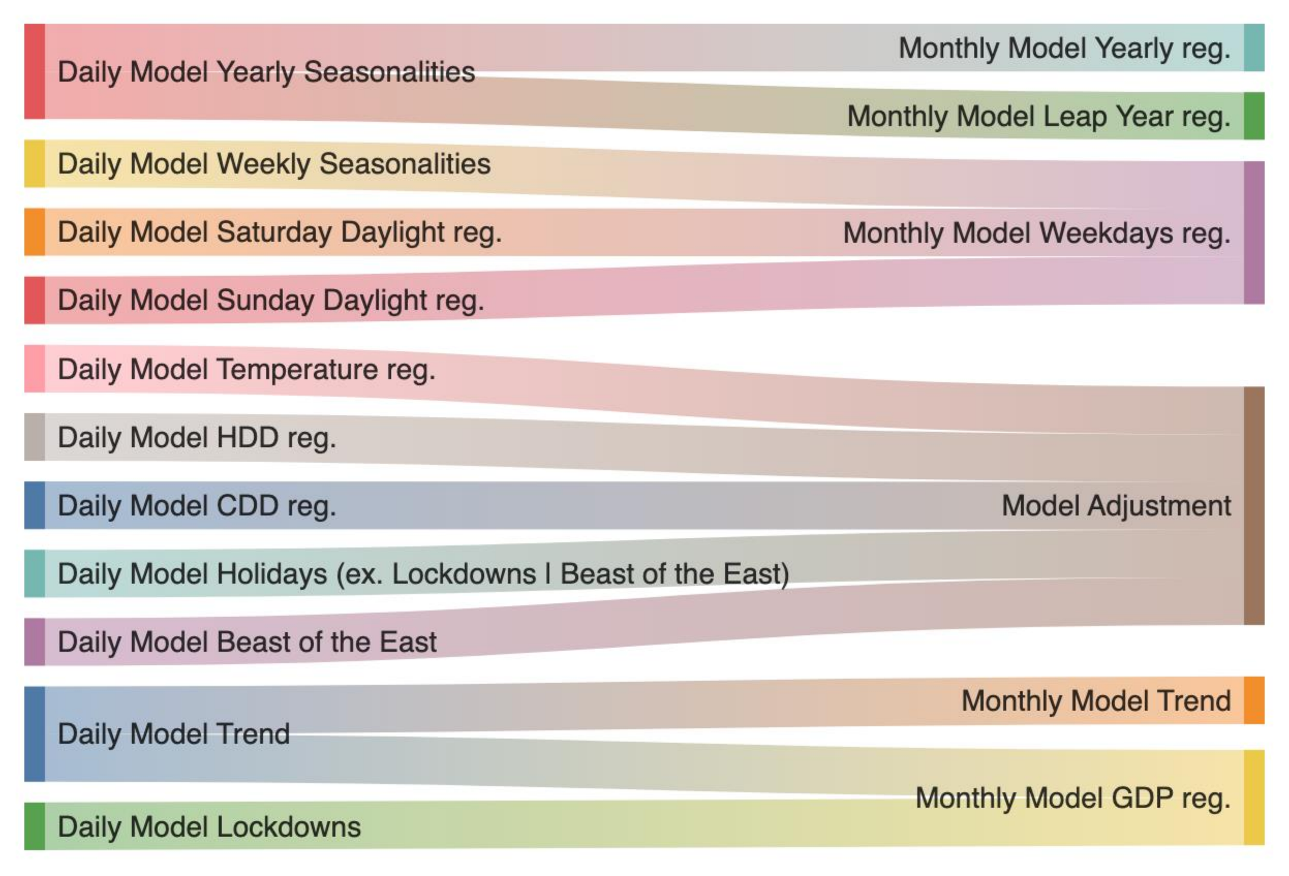


GB MONTHLY MODEL

Example

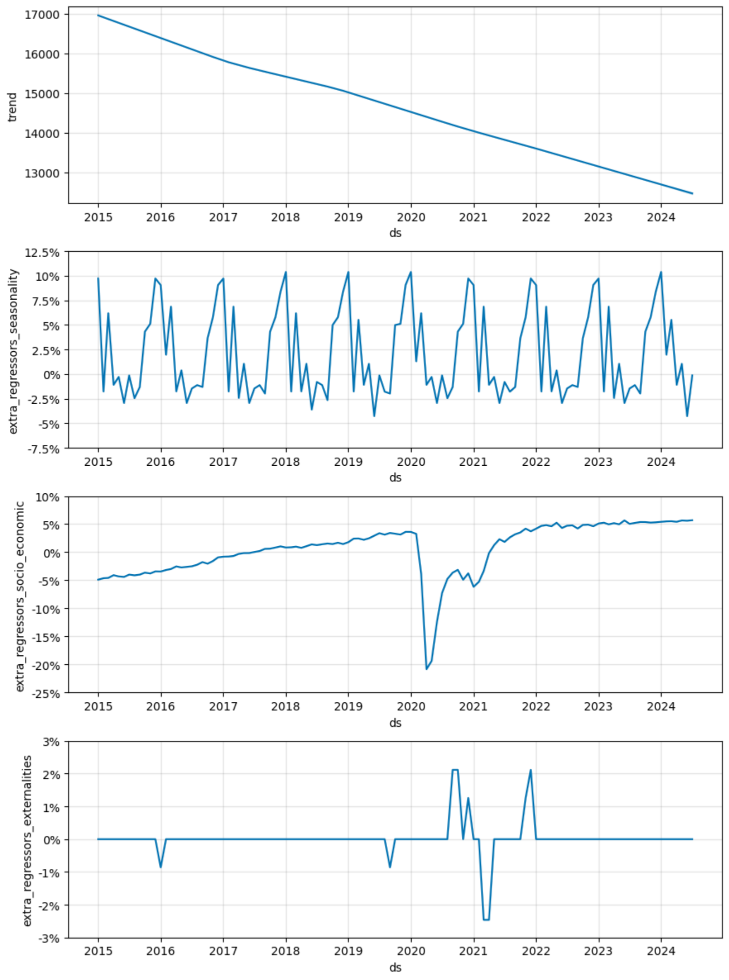
Into Monthly Modelling

- What we are taking out (adjusting) from the Daily Model are all the variables that are not being caught by the Monthly Model.



Main Components using weather/holidays-corrected demand…

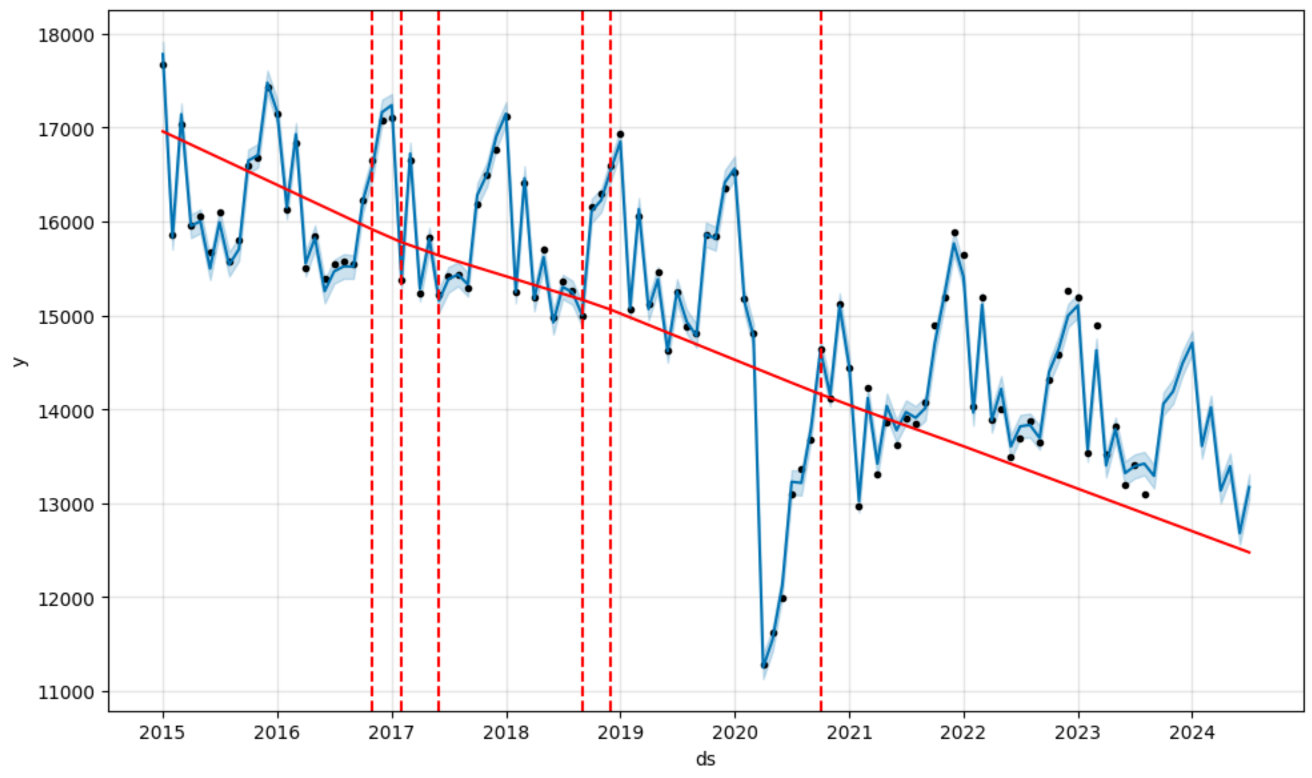
Trend, Seasonality+, GDP, anomalies



Left graph:

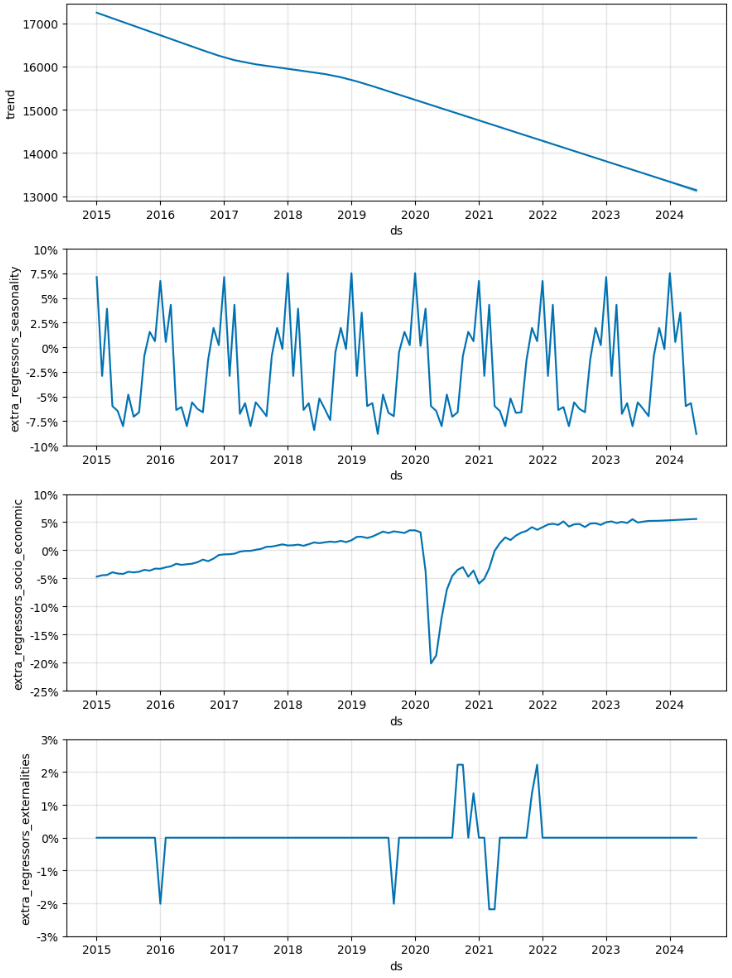
- Seasonality accounts for: leap years, number of weekdays.

Forecast, changepoints, uncertainty-intervals



Some thoughts

Trend, Seasonality+, GDP, anomalies



<u>Interpreting:</u>

- **Trend:** Without GDP as a regressor, the trend would require more changepoints and show a less pronounced downward slope, as time would be absorbing the signals typically explained by GDP.

- **Trend:** Therefore, in this model, the trend represents everything not captured by the other model variables, such as:
  - Efficiency improvements across GB,
  - The decarbonization of the GB economy,
  - Increased adoption of electric vehicles (EVs) and heat pumps,
  - Behind-the-meter generation, and more.

- **Thoughts:** This model makes creating different GDP scenarios straightforward, allowing for easy decomposition into granular model timesteps under various weather conditions.

PROPHET

SCENARIOS

Example

HYPEROPT