# Can you explain the price of electricity?

*A Machine Learning Approach to Short-Term Electricity Futures Modeling*

**2023 Season of the ENS Data Challenge — Designed by QRT**

*A submission by Cazals A.*

November 12, 2025

# Contents

# 1 Introduction

This report presents the complete methodology and results for the ENS Data Challenge, which focuses on modeling the daily price variation of 24H electricity futures contracts in France (FR) and Germany (DE). The primary goal is to explain the target price using simultaneous economic, meteorological, and production variables.

## 1.1 Challenge Context and Objectives

### 1.1.1 Daily Price Fluctuation Drivers

The modeling of electricity prices is inherently complex due to the confluence of diverse, high-impact factors. Daily price fluctuations are driven by:

- **Meteorological Dynamics:** Local weather variations (*temperature, wind*) directly affect both electricity **demand** (e.g., heating/cooling) and **generation** (e.g., solar, wind power).
- **Geopolitical and Commodity Shocks:** Events like geopolitical conflicts can drastically impact the price of key inputs for thermal generation (*natural gas, coal, carbon futures*), which are essential components of the energy mix in countries like France and Germany.
- **Market Interconnectedness:** Europe operates under a highly dynamic, interconnected market where energy trade (*import/export, cross-border exchanges*) instantly transmits price shocks and imbalances between neighboring countries.

### 1.1.2 Financial Context: Commodity Derivatives

The target variable is the daily price variation of **electricity futures contracts**, which are derivatives designed to manage price risk in the European energy market.

- **Futures Contracts:** These are **standardized** agreements traded on organized exchanges (e.g., EEX). They mitigate counterparty risk through a central clearing house and feature daily settlement (*mark-to-market*) via margin calls.
- **Forwards Contracts:** These are **customized** contracts traded Over-The-Counter (OTC). They carry high counterparty risk, and payment/delivery is usually settled only at maturity.

The market for these short-term energy futures is highly liquid. For instance, the European Energy Exchange (EEX) reported a total trading volume of approximately $\mathbf{8,438.6}$ **TWh in 2024** for its European power derivatives.

### 1.1.3 Goals and Model Interpretation

The primary aim is to **model and explain** the current daily price variation of these futures contracts using a set of simultaneous explanatory variables. Crucially, this task is explicitly defined as an **explanation problem**, focusing on determining the fundamental factors that influence the price *today*, rather than a pure forecasting problem.

### 1.1.4 Evaluation Metric: Spearman Rank Correlation

The challenge's success is formally evaluated using the **Spearman Rank Correlation Coefficient** ($r_s$) between the model's output and the actual daily price changes over the testing data set sample. Spearman's correlation measures the strength and direction of the monotonic relationship between two ranked variables. Mathematically, it is calculated based on the differences between the ranks of the paired observations:

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

where $n$ is the number of observations and $d_i$ is the difference between the ranks of the $i$-th paired observation ($d_i = R(X_i) - R(Y_i)$).

**In our context:** The variable $\mathbf{X}$ represents the **model's predictions** on the test set, and $\mathbf{Y}$ represents the **actual target values** (**TARGET**). $R(X_i)$ is therefore the rank of the $i$-th predicted

price change, and $R(Y_i)$ is the rank of the $i$-th actual price change. The coefficient $r_s$ measures how closely the rank order of the predictions matches the rank order of the actual outcomes.
It is a coherent metric for this problem for several reasons:

- **Robustness to Extremes:** Electricity prices are prone to extreme spikes (*price jumps*) due to sudden outages or demand changes. Since $r_s$ operates on **ranks** rather than the raw magnitude of the data, it is inherently **insensitive to these extreme values** (unlike Pearson correlation), making it more robust and stable for assessing model quality.
- **Non-Linearity:** Unlike the Pearson coefficient, which only measures linear relationships, Spearman correlation captures **monotonic (non-linear) relationships**. This is crucial, as the economic relationship between generation capacity, weather, and price is rarely linear (e.g., a drop in capacity leads to a non-linear price spike).

## 1.2 Methodology Overview

Our approach followed a structured, phased pipeline, starting with a fundamental linear assessment before moving to complex ensemble techniques:

1. **Data Preprocessing (`DataCleaner`):** Initial phase for missing value treatment (interpolation, iterative imputation) and coherence checking.
2. **Baseline Assessment (LASSO):** The first modeling step used LASSO Regression. This provided a crucial baseline and allowed for an early determination that the relationship between inputs and the target was **not linear**, validating the subsequent use of non-linear models.
3. **Signal and Lag Detection (`FeatureSignalDetector`):** This class performed the core time-series analysis, computing Spearman correlation coefficients at various lags to identify signal dependency and coherence, a way to evaluate the results of the Feature Engineering phase.
4. **Feature Engineering (`FeatureEngineering`):** Based on the detected signals and domain knowledge (e.g., wind power transformations), this phase generated necessary time-series features and physical derivatives.
5. **Ensemble Modeling:** The final phase involved a comparative study of regression models (**XGBoost** and **RandomForest**), optimized for the Spearman rank correlation.

# 2 Data Description and Preprocessing

## 2.1 Input and Target Variables

The provided dataset includes 1494 training rows and 654 test observations. Input data comprises 35 daily features, grouped into four main categories.

### 2.1.1 Input Features (Explanatory Variables)

The 35 daily features are detailed below:

- **Identifiers (3)**:
  - `ID`: Unique row identifier, associated with a day (`DAY_ID`) and a country (`COUNTRY`).
  - `DAY_ID`: Day identifier (dates anonymized).
  - `COUNTRY`: Country identifier (**DE** = Germany, **FR** = France).
- **Commodity Price Variations (3)**: Daily return (percentage variation) of key European energy commodities:
  - `GAS_RET`: European natural gas.
  - `COAL_RET`: European hard coal.
  - `CARBON_RET`: Carbon emissions futures.
- **Country-Specific Measures (x = DE or FR):** These features (total 29 columns) capture the domestic energy ecosystem.
  1. **Weather Measures (3 per Country):**
     - `x_TEMP`: Temperature.
     - `x_RAIN`: Rainfall.
     - `x_WIND`: Wind speed.
  2. **Energy Production Measures (7 per Country):** Daily production in the country $x$.
     - `x_GAS`: Natural gas production.
     - `x_COAL`: Hard coal production.
     - `x_HYDRO`: Hydro reservoir production.
     - `x_NUCLEAR`: Daily nuclear production.
     - `x_SOLAR`: Photovoltaic (solar) production.
     - `x_WINDPOW`: Wind power production.
     - `x_LIGNITE`: Lignite production.
  3. **Electricity Use and Exchange Metrics (9 total):**
     - `x_CONSUMPTON`: Total electricity consumption in country $x$.
     - `x_RESIDUAL_LOAD`: Electricity consumption after using all renewable energies (in country $x$).
     - `x_NET_IMPORT`: Imported electricity from Europe (in country $x$).
     - `x_NET_EXPORT`: Exported electricity to Europe (in country $x$).
     - `DE_FR_EXCHANGE`: Total daily electricity exchange from Germany to France.
     - `FR_DE_EXCHANGE`: Total daily electricity exchange from France to Germany.

### 2.1.2 Output Variable (Target)

The output data set is composed of two columns:

- `ID`: Unique row identifier (corresponding to the input identifiers).
- `TARGET`: Daily price variation for futures of 24H electricity baseload.

The solution files submitted by participants shall follow this output data set format, containing the `ID` and the predicted `TARGET`.

## 2.2 Missing Value Treatment

Missing values were treated through a multi-strategy approach to maintain data integrity, given the time-series nature of the inputs. The `DataCleaner` class implemented four complete imputation strategies for comparative analysis:

1. **Iterative Imputation (MICE):** Using the `sklearn.impute.IterativeImputer` class, this method models and predicts missing values based on all observed features.
2. **K-Nearest Neighbors (KNN Imputation):** A method implemented via `sklearn.impute.KNNImputer` where missing values are imputed using the average of values from the $k$ nearest non-missing samples.
3. **Interpolation (Polynomial Order 3):** For numerical time-series features, interpolation was performed using **Cubic Splines** (*polynomial interpolation of order 3*) via the **Pandas** `DataFrame.interpolate()` method.
4. **Mean Imputation (Benchmark):** This simpler method, served as a linear benchmark against which the performance of the more complex models (Iterative, KNN, Interpolation) was measured during the coherence check.

### 2.2.1 Imputation Coherence Check

To validate the robustness of the imputation strategy, a `CheckFilledData` method was used. This technique involved intentionally masking a subset of valid data points and comparing the imputed values against the known actual values using the Root Mean Squared Error (RMSE) (Table 1).

Table 1: Comparison of RMSE per Feature for Missing Data Imputation Methods

| Feature | Iterative (MICE) | Mean Imputation | KNN Imputation | Interpolate | Best Method |
|---|---|---|---|---|---|
| DE_CONSUMPTION | **0.022** | 0.414 | 0.067 | 1.091 | MICE |
| FR_CONSUMPTION | **0.019** | 0.979 | 0.045 | 2.564 | MICE |
| DE_FR_EXCHANGE | **0.046** | 0.933 | 0.105 | 1.970 | MICE |
| FR_DE_EXCHANGE | **0.046** | 0.909 | 0.111 | 1.753 | MICE |
| DE_NET_EXPORT | **0.042** | 0.664 | 0.083 | 1.708 | MICE |
| FR_NET_EXPORT | **0.087** | 1.201 | 0.099 | 2.619 | MICE |
| DE_NET_IMPORT | **0.044** | 0.940 | 0.116 | 1.742 | MICE |
| FR_NET_IMPORT | **0.069** | 1.271 | 0.093 | 2.664 | MICE |
| DE_GAS | **0.150** | 0.753 | 0.167 | 1.817 | MICE |
| FR_GAS | **0.115** | 0.709 | 0.101 | 1.764 | MICE |
| DE_COAL | **0.132** | 0.566 | 0.107 | 1.759 | MICE |
| FR_COAL | 0.101 | 0.233 | **0.051** | 0.514 | KNN |
| DE_HYDRO | 0.827 | 1.260 | **0.469** | 2.574 | KNN |
| FR_HYDRO | **0.307** | 1.265 | 0.309 | 2.881 | MICE |
| DE_NUCLEAR | 0.243 | 0.661 | **0.110** | 1.622 | KNN |
| FR_NUCLEAR | **0.039** | 0.779 | 0.074 | 1.958 | MICE |
| DE_SOLAR | **0.089** | 1.145 | 0.163 | 2.894 | MICE |
| FR_SOLAR | **0.159** | 1.331 | 0.189 | 3.038 | MICE |
| DE_WINDPOW | **0.019** | 1.167 | 0.129 | 3.099 | MICE |
| FR_WINDPOW | **0.092** | 1.467 | 0.445 | 3.359 | MICE |
| DE_LIGNITE | **0.177** | 0.755 | 0.167 | 1.542 | KNN |
| DE_RESIDUAL_LOAD | **0.012** | 0.701 | 0.086 | 1.868 | MICE |
| FR_RESIDUAL_LOAD | **0.018** | 0.869 | 0.054 | 2.294 | MICE |
| DE_RAIN | **0.649** | 1.006 | 0.670 | 2.290 | MICE |
| FR_RAIN | 1.121 | 1.438 | **0.851** | 2.734 | KNN |
| DE_WIND | **0.140** | 0.898 | 0.193 | 2.385 | MICE |
| FR_WIND | **0.173** | 1.284 | 0.245 | 2.673 | MICE |
| DE_TEMP | **0.348** | 0.963 | 0.443 | 1.917 | MICE |
| FR_TEMP | 0.422 | 0.806 | **0.350** | 2.079 | KNN |
| GAS_RET | 1.085 | 1.134 | **0.920** | 2.880 | KNN |
| COAL_RET | 0.797 | 0.926 | **0.740** | 2.188 | KNN |
| CARBON_RET | **0.841** | 1.351 | 0.983 | 2.771 | MICE |
| **RMSE Global Mean** | **0.263** | 0.962 | 0.273 | 2.219 | MICE |

# 3 First Approach: Monotonic Penalized Linear Regression (RLP)

## 3.1 Theoretical Framework and Objectives

The primary goal is to find a model that maximizes the **Spearman Rank Correlation ($\rho$)** by imposing a soft constraint on the predicted order. Unlike standard Linear Regression, the Monotonic Penalized Linear Regression (RLP) minimizes the Mean Squared Error (MSE) while adding a penalty term proportional to the **violation of the intended order** between observation pairs.

**RLP Objective Function:** The model finds the vector of coefficients ($\boldsymbol{\beta}$) that minimizes the following objective function:

$$\min_{\boldsymbol{\beta}} \left( \|\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}\|^2 + \lambda \sum_{i<j, y_i>y_j} \max(0, -(\mathbf{X}_i - \mathbf{X}_j)\boldsymbol{\beta}) \right)$$

- **MSE Term ($\|\mathbf{X}\boldsymbol{\beta} - \mathbf{Y}\|^2$):** This is the standard Least Squares term, driving the model toward absolute predictive accuracy.
- **Monotonicity Penalty (Hinge Loss):** This term penalizes the model when an observation $\mathbf{X}_i$ that should have a higher target value ($y_i > y_j$) is incorrectly ranked below observation $\mathbf{X}_j$.
  - The term $\mathbf{X}_i\boldsymbol{\beta} < \mathbf{X}_j\boldsymbol{\beta}$ implies an order violation (since $y_i$ should be greater than $y_j$).
  - The $\max(0, \dots)$ ensures that only **negative violations** are penalized (similar to a Hinge Loss in SVM).
- **Hyperparameter $\lambda$:** Controls the strength of the monotonic penalty. By optimizing $\lambda$ using $\rho$ as the scoring metric, we find the optimal trade-off between absolute accuracy (MSE) and reliable ranking.

---

**Prerequisite: Feature Standardization**

For the RLP model to be effective, the **standardization** of all features is crucial. Standardization ensures that the contribution of each feature to the prediction is unbiased by its original scale, allowing the coefficients ($\boldsymbol{\beta}$) to reflect true predictive relevance.

- **Issue of Scale:** Without standardization, variables with naturally large scales (e.g., millions) would necessarily have coefficients with small magnitudes to compensate. These coefficients would be unfairly penalized compared to features with naturally small scales.
- **Illustrative Example of Penalty Bias:** Consider two features, $\mathbf{X_A}$ (e.g., surface in $m^2$) and $\mathbf{X_B}$ (e.g., surface in $mm^2$), which have the exact same predictive importance.
  - Due to the vast scale difference ($\mathbf{X_B} = 10^6 \times \mathbf{X_A}$), the model must set $\beta_\mathbf{B}$ to be $10^6$ times smaller than $\beta_\mathbf{A}$ for the two terms ($\beta X$) to have equal impact on $Y$.
  - **Impact of Scale on Violation Magnitude (Bias Propagation):** The RLP penalty term, $\lambda \sum \max(0, \mathbf{X}_j\boldsymbol{\beta} - \mathbf{X}_i\boldsymbol{\beta})$, is based on the difference between the two predictions. This difference is:
    $$\Delta\hat{Y} = \hat{Y}_j - \hat{Y}_i = \sum_k (\mathbf{X}_{j,k} - \mathbf{X}_{i,k})\boldsymbol{\beta}_k$$
    For the feature $\mathbf{X}_A$ (small unit of measure, large $\boldsymbol{\beta}_A$) compared to $\mathbf{X}_B$ (large unit, small $\boldsymbol{\beta}_B$):
    - **Instability Due to Large $\boldsymbol{\beta}$:** Because $\beta_A$ is huge (to compensate for the small unit $m^2$), a tiny, numerically insignificant change in $\beta_A$ (e.g., $10^{-4}$) causes an important swing in the violation magnitude ($\Delta\hat{Y}$).

    - ∗ **Solveur Bias:** The RLP optimizer (CVXPY) will struggle to correct violations via the large coefficients ($\boldsymbol{\beta}_A$), viewing them as unstable and penalizing the model's overall stability based on this arbitrary scale difference.

- **Fair Penalty:** By standardizing all features to have a mean $\mu \approx 0$ and a standard deviation $\sigma \approx 1$, the linear term ($\mathbf{X}\boldsymbol{\beta}$) is balanced. This ensures that the penalty for violating the monotonic order ($\max(0, \dots)$) is applied **equitably** across all features, making the resulting constraint meaningful for true predictive power.

## 3.2 Preventing Data Leakage and the Role of the Pipeline

The methodology for handling standardization during Cross-Validation (CV) remains critical to prevent **Data Leakage**. We must ensure that the scaling statistics ($\mu$ and $\sigma$) are learned **only** on the Training subset and applied to the separate Validation subset for each fold.

- **The Risk:** Calculating $\mu$ and $\sigma$ from the entire dataset before CV biases the model's performance estimation.
- **The Solution:** We utilize the `Pipeline` object from `scikit-learn` to enforce the sequential execution of `StandardScaler` and the `MonotonicLinearRegressor` estimator within each CV fold.

## 3.3 Hyperparameter Optimization for Ranking ($\lambda$)

The optimal regularization hyperparameter ($\lambda$) for the monotonic penalty was determined through rigorous optimization using **Cross-Validation** (CV).

- **Optimization Tool:** Given the computational complexity of solving the constrained optimization problem (using tools like `CVXPY`) at each step, a dedicated manual CV loop or specialized wrapper was necessary instead of `GridSearchCV`.
- **Custom Scoring Metric:** The optimization process was strictly guided by a **Custom Scorer** designed to maximize the **Spearman Rank Correlation ($\rho$)**. This ensured that the selected $\lambda$ provided the best achievable ranking performance, which is the core objective of the RLP.
- **Cross-Validation Strategy:** The standard **K-Folds Cross-Validation** ($\mathbf{K = 5}$) was employed to estimate the generalization capability of the model accurately.

## 3.4. RLP Results and Monotonicity Analysis

### 4. Optimization of the Monotonicity Constraint ($\lambda$)

The crucial hyperparameter $\lambda$ controls the trade-off between minimizing MSE and enforcing rank preservation. Its optimal value was determined by maximizing the mean Spearman correlation ($\rho$) across cross-validation folds.

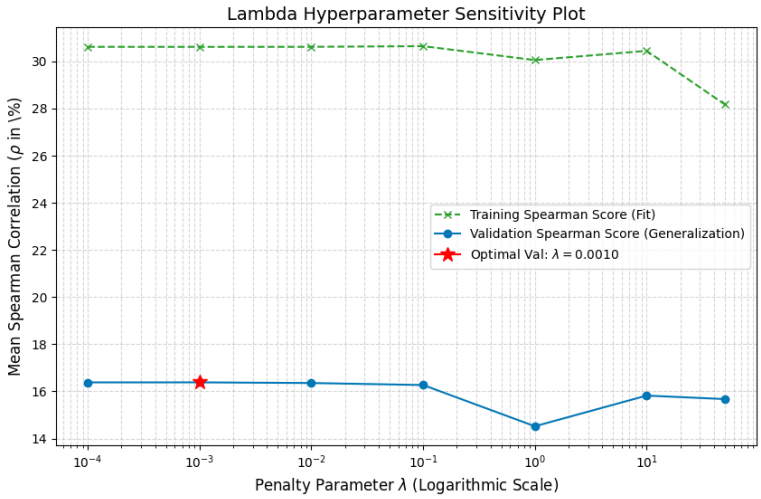



Figure 1: Sensitivity of the Monotonic Penalty Hyperparameter ($\lambda$). The optimal $\lambda$ minimizes the gap between the Training Score (Fit) and the Validation Score (Generalization).

**Analysis of $\lambda$ Sensitivity:**

- **Optimal $\boldsymbol{\lambda}$ Selection:** As shown in Figure 1, the optimal penalty was found to be $\boldsymbol{\lambda = 0.0010}$. At this low value, the model is minimally constrained, suggesting that the primary force driving the fit is the MSE term, with the monotonic penalty acting only on the most severe local rank violations.
- **Generalization Gap:** A significant difference persists between the Training Score ($\rho \approx 30\%$) and the Validation Score ($\rho \approx 16.5\%$). This large gap confirms that the simple linear model **overfits the noise** in the training set, and the true generalization capability is limited.

### 3.4.2. Predictive Performance and Rank Visualization

The model's performance confirms the limitations of linearity for predicting energy price ranks.

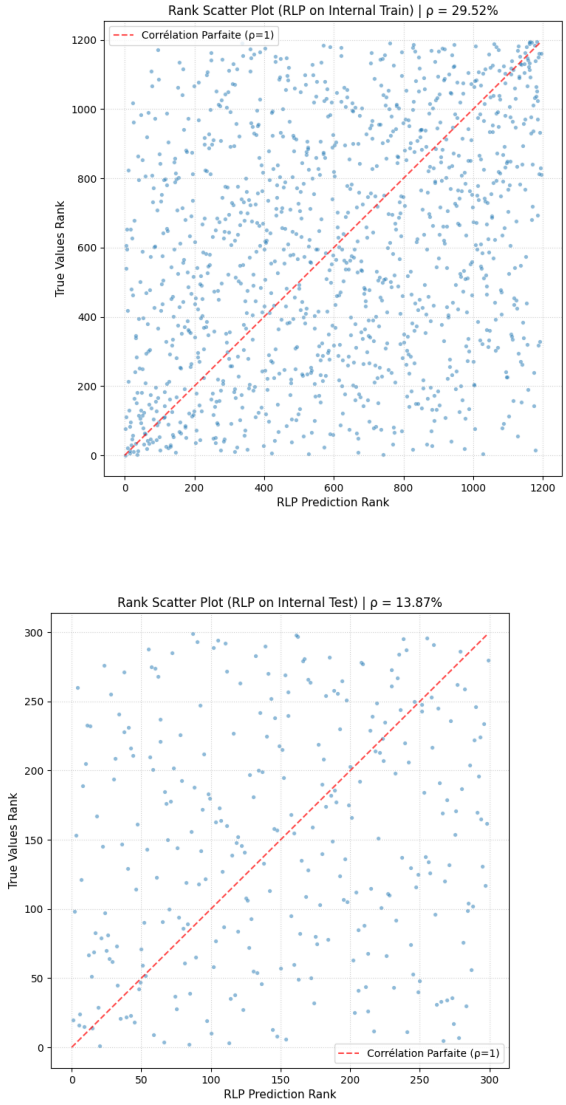



Figure 2: Rank scatter plots for the RLP Model. **Left:** Internal Train Set ($\rho \approx 29.52\%$). **Right:** Internal Test Set ($\rho \approx 13.87\%$).

**Analysis of the Rank Scatter Plot:**

- **Validation Performance:** The low Spearman correlation on the Internal Test Set ($\boldsymbol{\rho \approx 13.87\%}$) validates the observation from the $\lambda$ tuning: even with an optimal penalty, the linear model struggles to reliably predict the relative ordering of high-value vs. low-value price movements.
- **Visible Scatter:** The wide scatter of points around the ideal line ($\rho = 1$) in Figure 2 (Right) confirms that the model is often unable to distinguish between observations that should rank in the 50th position and those that should rank near the 250th position.

# 4. Structural Analysis via Basic Rank-Based Linear Regression (RLR)

An other idea was to implement a Rank-Based Linear Regression (RLR) model. This approach is defined by its core transformation: the continuous target variable ($Y$) is converted into its corresponding ranks($\mathbf{Y}_{\text{ranks}}$) before the standard LinearRegression model is trained.

$$\text{Model Target:} \quad Y_{\text{rank}} = \text{rankdata}(Y)$$

This strategy intentionally shifts the focus of the regression from predicting the absolute magnitude of the energy price to predicting its relative ordering within the training set.

1. **Target Transformation and Scaling:** The target $Y \in \mathbb{R}$ is transformed to its rank $Y_{\text{rank}} \in \{1, 2, \ldots, N\}$. Concurrently, the features $\mathbf{X}$ are standardized using a StandardScaler to ensure equal contribution from all covariates in the linear fitting process.

2. **Model Formulation:** The RLR model minimizes the standard Ordinary Least Squares (OLS) loss, but with respect to the rank prediction ($\hat{Y}_{\text{rank}}$):

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left( Y_{\text{rank}}^{(i)} - \left( \beta_0 + \sum_{j=1}^{P} \beta_j X_{\text{scaled},j}^{(i)} \right) \right)^2$$

**Performance Overview and Context:** The RLR model's performance was evaluated using the Spearman Rank Correlation ($\rho$).

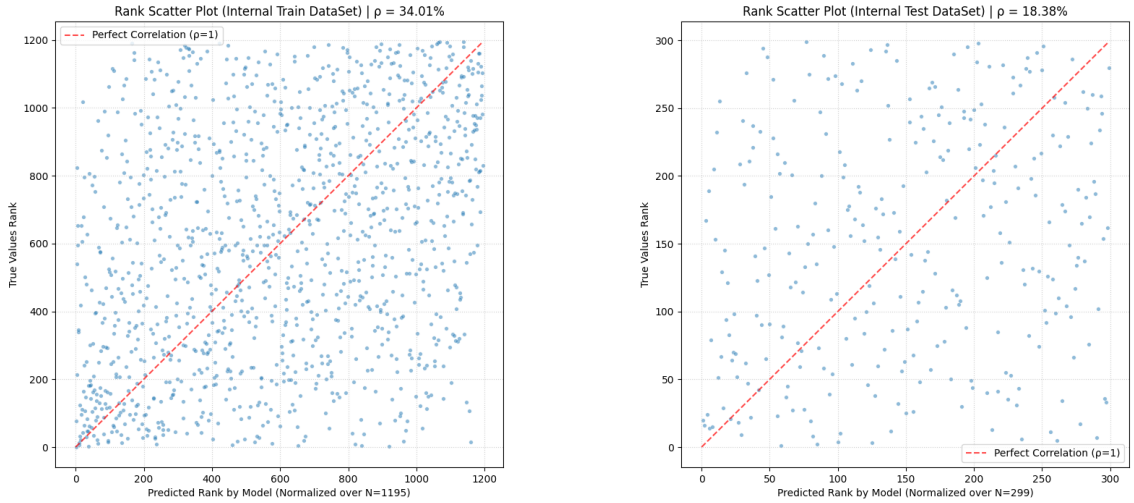



Figure 3: Rank Scatter Plots for the RLR Model. Left: Training Set ($\rho = 34.01\%$). Right: Internal Test Set ($\rho = 18.38\%$). The limited capacity of the linear model is visible through the wide scatter around the perfect correlation line ($\rho = 1$).

The focus on ranks made this baseline approach more effective for the challenge objective compared to an attempt penalized by order constraints (such as a penalised MSE). Crucially, the final submission based on this RLR structure achieved an official Spearman correlation score of $\rho = \mathbf{0.1599}$, successfully surpassing the initial benchmark established by the challenge organizer.

**Critical Conclusion:** While the RLR provides an interpretable, rank-aware baseline for structural analysis, its simple linear structure cannot capture the complex, non-linear dependencies in the energy price time series.

# 4 Feature Signal Detection (FeatureSignalDetector)

The FeatureSignalDetector class was designed to understand relationships between variables and the TARGET across different time scales, providing diagnostics for the feature engineering coherence. An attempt to detect treshholds effect was made, but resulted in poor results : implementation failed.

## 4.1 Lag and Correlation Analysis

The core functionality of FeatureDetector was to systematically explore temporal dependencies, focusing on both linear and monotonic relationships:

1. **Optimal Lag Detection (find_lag):** For each input feature **X**, the class computed the correlation between the current target $\mathbf{Y_t}$ and the feature lagged by $k$ periods, $\mathbf{X_{t-k}}$, across a range of time lags ($k$). Specifically, it calculated the **Pearson correlation** (linear relationship) and the **Spearman Rank Correlation Coefficient** (monotonic relationship).

Table 2: Feature Signal Detector Results (Full Feature Set Ranked by Best Spearman $\rho$)

| **Feature** | **Best Lag ($k$)** | **Spearman $\rho$ (Best $\lvert k\rvert$)** | **Pearson $\rho$ (Best $\lvert k\rvert$)** |
|---|---|---|---|
| DE_NET_EXPORT | 0 | -0.19 | -0.15 |
| DE_NET_IMPORT | 0 | 0.19 | 0.15 |
| DE_WINDPOW | 0 | -0.19 | -0.15 |
| DE_RESIDUAL_LOAD | 0 | 0.18 | 0.13 |
| FR_WINDPOW | 0 | -0.16 | -0.13 |
| DE_HYDRO | 0 | 0.15 | 0.09 |
| DE_GAS | 0 | 0.13 | 0.10 |
| DE_WIND | 0 | -0.10 | -0.08 |
| CARBON_RET | 5 | 0.09 | 0.05 |
| FR_WIND | 0 | -0.07 | -0.05 |
| FR_HYDRO | 0 | 0.06 | 0.05 |
| FR_RESIDUAL_LOAD | -5 | 0.04 | 0.04 |
| FR_TEMP | -4 | -0.04 | -0.06 |
| FR_COAL | -2 | 0.04 | 0.04 |
| FR_SOLAR | -4 | 0.04 | 0.04 |
| DE_LIGNITE | -2 | 0.04 | 0.05 |
| FR_NET_EXPORT | 5 | -0.03 | -0.03 |
| FR_NET_IMPORT | 5 | 0.03 | 0.03 |
| FR_NUCLEAR | -5 | 0.03 | 0.05 |
| GAS_RET | 5 | 0.03 | 0.05 |
| COAL_RET | 4 | 0.03 | 0.04 |
| DE_COAL | -5 | 0.03 | 0.06 |
| FR_CONSUMPTION | -4 | -0.03 | -0.04 |
| DE_CONSUMPTION | 4 | 0.03 | 0.06 |
| DE_TEMP | 0 | -0.02 | -0.04 |
| DE_RAIN | -2 | 0.02 | 0.05 |
| FR_GAS | -4 | -0.01 | -0.04 |
| DE_NUCLEAR | 3 | -0.01 | -0.04 |
| DE_FR_EXCHANGE | 5 | -0.01 | -0.03 |
| FR_DE_EXCHANGE | 5 | 0.01 | 0.03 |
| FR_RAIN | -1 | -0.00 | -0.05 |

**Analysis of Feature Signal Detector Results:**

The systematic analysis of lag and simultaneity (Table 2) underlined several critical points regarding the data structure:

- **Dominance of Instantaneous Factors:** The strongest correlations ($|\rho| \approx \mathbf{0.18 - 0.19}$) are consistently observed at **Lag** $k = 0$ for key German supply and balance factors (`DE_NET_EXPORT`, `DE_WINDPOW`, `DE_RESIDUAL_LOAD`). This indicates that the most relevant information for current price variation is available instantaneously.
- **Weak Signal and Non-Causality:** The overall magnitude of correlations remains weak ($|\rho| \leq \mathbf{0.20}$), confirming that raw features, even when lagged, do not offer a robust predictive signal. Longer positive lags (e.g., $k = 5$ for `CARBON_RET`) are treated as non-causal statistical artifacts due to the market's efficiency.

## 4.2 Feature Engineering Validation

The underlying class structure, initially conceived as a 'FeatureSignalDetector', was primarily designed to find underlying temporal dependencies that might not be readily captured by standard regression or tree-based models, which focus exclusively on the feature-target association. However, the analysis failed to reveal any relevant underlying signals. Moving forward, this framework will be repurposed to evaluate the coherence of new features derived from feature engineering by providing a stable, rank-based metric (like $\rho$) for assessing their correlation with the target before integration into the final ensemble models.

# 5 Advanced Feature Engineering (FeatureEngineering)

Using domain knowledge, a set of features was added to enhance the predictive power of the ensemble models.

## 5.1 Generated Features and Physics-Based Rationale

The engineering process created over 100 new variables across four categories, guided by the necessity to capture non-linear market behavior:

1. **Mathematical and Physics-Based Transformations:** Designed to capture non-linear effects and physical realities.
   - **Wind Power Physics ($\mathbf{X^3}$):** Given the theoretical model $\mathbf{P} \propto \mathbf{u^3}$ (Power $\propto$ Wind Speed Cubed), $\mathbf{X^3}$ provides the model with a feature that directly represents the **maximum physical input potential** of wind farms.
   - General Non-Linearity: Polynomial ($\mathbf{X^2}, \mathbf{X^3}$) and Signed Root ($\text{sign}(\mathbf{X}) \cdot \sqrt{|\mathbf{X}|}$) transformations were applied to key drivers (Temp, Load).
   - Market Physics: The **Solar Efficiency Index** (SOLAR/TEMP) captures the thermodynamic loss of efficiency in solar panels due to heat.
2. **Domain-Specific Interactions (Market Logic):** Created ratios and products reflecting known economic mechanisms, aiming to model complex price formation mechanisms.
   - **Marginal Price Proxy (Residual Load $\times$ Gas Price):** This interaction models the **marginal production cost**. Under the *Merit Order* principle, high demand (`Residual Load`) often forces the grid to rely on the most expensive flexible source—the gas power plants—explicitly linking high demand pressure with the primary driver of price spikes (fuel cost).
   - **Fuel Substitution Ratio (GAS_RET/COAL_RET):** This ratio captures the market's incentive for **fuel switching**. A high ratio (expensive gas relative to coal) indicates producers are incentivized to burn coal, impacting dispatch flexibility and available supply.
   - **Aggregate Renewable Capacity (Wind + Solar + Hydro):** This sum quantifies the total **non-dispatchable, low-cost capacity**. It provides a simple measure of overall system stress: high renewable capacity pushes marginal prices down, while low capacity increases reliance on costly thermal plants.
3. **Temporal Aggregations (Rolling Features):** Used Rolling Mean (**ROLL_MEAN**) and Standard Deviation (**ROLL_STD**) over $\mathbf{3, 7, 14}$ days for highly correlated features, including RAIN to model hydrological inertia.
4. **Cross-Features and Shock Detection:** Designed to exploit inter-country dynamics (FR_LOAD − DE_LOAD) and sudden shifts (Delta $= \mathbf{X_t} - \mathbf{X}_{\text{ROLL\_MEAN\_7D}}$).

## 5.2 Critique of Evaluation and Final Feature Strategy

The feature engineering successfully amplified the signal, yielding over 30 features with $|\rho| \geq 0.1$ (see Table 3). However, the primary challenge lay in correctly evaluating the generated features:

- **Limitation of Spearman Evaluation:** The initial strategy relied on Spearman $\rho$ to assess feature quality. Crucially, for positive-valued features (like wind speed), monotonic transformations ($\mathbf{X^2}, \mathbf{X^3}, \sqrt{\mathbf{X}}$) **do not change the rank ordering**. Consequently, the Spearman correlation of the raw feature is mathematically identical to the Spearman correlation of the transformed feature.
- **The Result:** This means that while we successfully created physically relevant features (like $\mathbf{X^3_{\text{WIND}}}$), the initial Spearman analysis **failed to correctly distinguish** their predictive value from the raw input.

Table 3: Feature Signal Detector Results (Top 30 Features Ranked by Best Spearman $\rho$)

| **Feature** | **Best Lag ($k$)** | **Spearman $\rho$ (Best $\lvert k\rvert$)** | **Pearson $\rho$ (Best $\lvert k\rvert$)** |
|---|---|---|---|
| DE_NET_EXPORT | 0 | -0.19 | -0.15 |
| DE_NET_IMPORT | 0 | 0.19 | 0.15 |
| DE_NET_EXPORT_CBRT | 0 | -0.19 | -0.15 |
| DE_NET_IMPORT_CBRT | 0 | 0.19 | 0.15 |
| DE_WINDPOW | 0 | -0.19 | -0.15 |
| DE_WINDPOW_CUBE | 0 | -0.19 | -0.14 |
| DE_WINDPOW_SR | 0 | -0.19 | -0.13 |
| DE_NET_EXPORT_DELTA_VS_7D | 0 | -0.18 | -0.12 |
| DE_NET_IMPORT_DELTA_VS_7D | 0 | 0.18 | 0.12 |
| DE_RESIDUAL_LOAD | 0 | 0.18 | 0.13 |
| DE_RESIDUAL_LOAD_CUBE | 0 | 0.18 | 0.13 |
| DE_RESIDUAL_LOAD_SR | 0 | 0.18 | 0.12 |
| DE_RESIDUAL_LOAD_DELTA_VS_7D | 0 | 0.16 | 0.12 |
| DE_WINDPOW_DELTA_VS_7D | 0 | -0.16 | -0.13 |
| FR_WINDPOW | 0 | -0.16 | -0.13 |
| FR_WINDPOW_CUBE | 0 | -0.16 | -0.13 |
| FR_WINDPOW_SR | 0 | -0.16 | -0.11 |
| DE_HYDRO | 0 | 0.15 | 0.09 |
| DE_WINDPOW_ROLL_MEAN_3D | -1 | -0.15 | -0.14 |
| DE_WINDPOW_ROLL_STD_3D | -1 | -0.15 | -0.14 |
| FR_DE_RES_LOAD_DIFF | 0 | -0.14 | -0.09 |
| FR_WINDPOW_DELTA_VS_7D | 0 | -0.14 | -0.12 |
| DE_GAS | 0 | 0.13 | 0.10 |
| DE_HYDRO_DELTA_VS_7D | 0 | 0.13 | 0.06 |
| FR_WINDPOW_ROLL_MEAN_3D | -1 | -0.13 | -0.11 |
| DE_GAS_ROLL_MEAN_3D | -1 | 0.12 | 0.10 |
| DE_GAS_DELTA_VS_7D | 0 | 0.12 | 0.08 |
| DE_WINDPOW_ROLL_STD_7D | -3 | -0.12 | -0.11 |
| DE_NET_EXPORT_ROLL_MEAN_3D | -1 | -0.12 | -0.11 |
| DE_NET_IMPORT_ROLL_MEAN_3D | -1 | 0.12 | 0.11 |

## 5.3 Feature Selection Strategy for Ensemble Models: Embedded Selection

Prior to selection, non-zero time lags exhibiting a Spearman correlation ($\rho$) over 10% with the target (e.g., `DE_WINDPOW_ROLL_MEAN_3D`, lag $= -1$) were added to the feature set. The chosen solution was an Embedded Feature Selection approach, as the Spearman $\rho$ metric proved insufficient for distinguishing between redundant monotonic transformations (e.g., $\mathbf{X}$ vs $\mathbf{X^3}$). This involved retaining the top 50 influential features (e.g., based on the Gini Index for Random Forest) from a preliminary non-linear model to filter the dataset, thus enabling a more efficient and targeted hyperparameter search aimed at maximizing the final Spearman correlation ($\rho$) on the reduced feature set.

# 6 Ensemble Modeling and Optimization

The final modeling stage was to use the RandomForest model, tuned via `GridSearchCV` with the **Spearman Rank Correlation** as the optimization metric for hyperparameters.

## 6.1 Random Forest (RF)

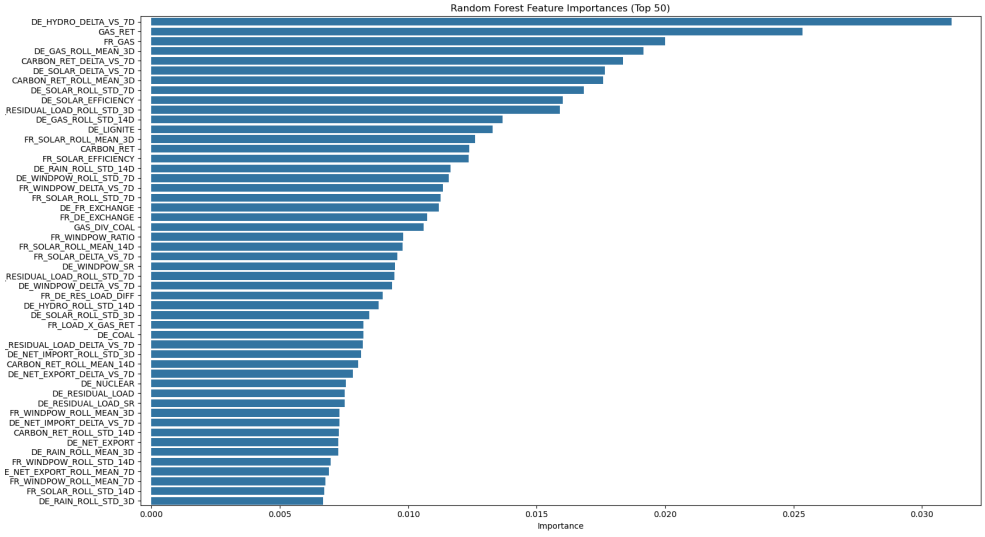
The Random Forest algorithm constructs an ensemble of $N$ decision trees (typically $N = 100$) using the **CART** method (*Classification and Regression Trees*). Each tree is trained on a random bootstrap sample of the original dataset and built by recursively splitting the data to minimize the **Mean Squared Error (MSE)** on the target variable. The final regression output of the Random Forest is obtained by averaging the predictions from all individual trees, which reduces variance and improves generalization.

## 6.2 Random Forest (RF) Results Analysis

The Random Forest model was employed, with hyperparameter tuning guided by the `Spearman Rank Correlation`. The analysis first focuses on the selection of features and then on the performance results, which reveal a significant overfitting issue.

### 6.2.1 Feature Selection and Importance

Embedded Feature Selection approach :




- **Dominant Features:** The top of the chart is dominated by variables related to energy market dynamics between France (FR) and Germany (DE). Key predictive features include:
  - `DE_HYDRO_DELTA_VS_7D`: Weekly changes in German hydro-power capacity.
  - `GAS_RET` / `FR_GAS` / `DE_GAS_ROLL_MEAN_3D`: Variables reflecting gas prices and their short-term trends.
- **Selection Rationale:** Given that these features show the highest importance scores, they are confirmed as the most relevant set for predicting the target variable.

### 6.2.2 Model Performance: Overfitting and Test Set Size

**Overfitting (Train Set)** The model demonstrates a good fit on the training data, achieving a Spearman correlation of **90.40%**. The points tightly clustered around the $y = x$ line indicate the model has essentially **memorized** the rankings. This near-perfect score is a strong signal of severe overfitting : the model has learned the training data's noise rather than generalizable patterns.

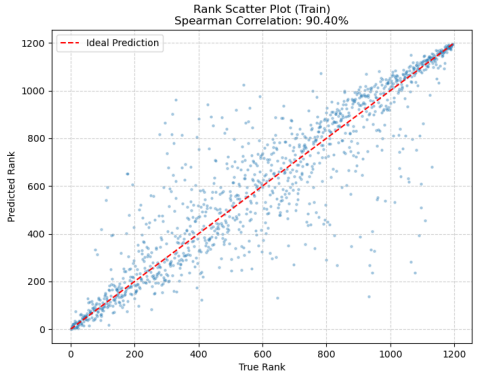



Figure 4: Rank Scatter Plot on Training Data (Spearman: 90.40%).

**Questionable Test Result (Test Set)** The generalization capacity is poor, as evidenced by the dramatic drop in performance on the test set, achieving a Spearman correlation of only **18.46%**.

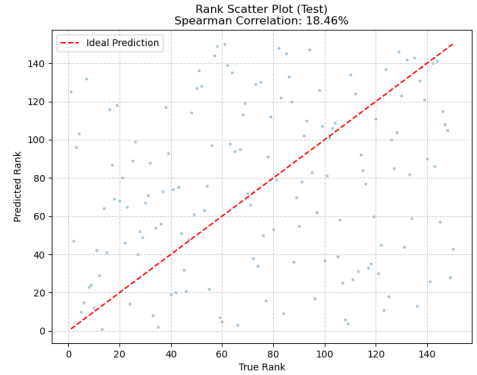



Figure 5: Rank Scatter Plot on Test Data (Spearman: 18.46%).

- **Train Max Rank:** $\sim 1200$ (indicating $\sim 1200$ data points)
- **Test Max Rank:** $\sim 150$ (indicating $\sim 150$ data points)

While the low score of 18.46% unequivocally confirms the overfitting issue, the result must be interpreted with caution:

1. **Small Sample Noise:** A test set of only $\sim 150$ points is highly susceptible to random noise. A single poorly predicted rank can have a much larger impact on the overall correlation metric compared to a larger sample.

Table 4: Random Forest Performance Summary

| Dataset | Spearman Correlation | Data Points (Approx.) |
|---|---|---|
| Training | 90.40% | $\sim 1200$ |
| Testing | 18.46% | $\sim 150$ |

### 6.2.3 Official Submission Result

Following the optimization and regularization , the final model was submitted to the official competition leaderboard (https://challengedata.ens.fr/participants/challenges/97/). This submission yielded a Spearman Rank Correlation of **0.2629**.

| | | | |
|---|---|---|---|
| 135 | 3 novembre 2025 14:07 | antoninc | 0,2629 |

Figure 6: Official Competition Leaderboard Rank (135/1050+).

This result placed the model at rank **135** out of over 1050 participants.

Furthermore, the final score of **0.2629** represents a substantial improvement of approximately **65.66%** over the initial benchmark of 0.1587.

# 7 Conclusion

## 7.1 Limitations and Further Exploration

It is important to note that the final report only presented the most successful modeling attempts. Several approaches were implemented and tested but did not yield superior results on the official public leaderboard.

## 7.2 Limitations and Further Exploration

It is important to note that the final report only presented the most successful modeling attempts. Several complex approaches were implemented and tested but did not yield superior results on the official public leaderboard.

### 7.2.1 Models and Objectives Tested

Several Tree-Based Ensemble methods, including **Random Forest (RF)**, **XGBoost**, and **LightGBM (LGBM)**, were evaluated. Surprisingly, the standard RF model, which uses Mean Squared Error (MSE) as its base loss function, yielded the best performance on the final leaderboard.

This outcome is particularly notable because more theoretically aligned methods were implemented but failed to surpass the RF:

- **Pairwise Ranking Objective:** Given the challenge's metric (Spearman Rank Correlation), the gold standard for optimization is a ranking loss function. An **XGBoost model** was implemented using the **rank:pairwise objective**—a method specifically designed to minimize mis-ranked pairs. Despite these theoretical advantages and its dedicated Python implementation, this specialized model **did not outperform** the simpler RF model on the leaderboard.
- **Dual-Model Architecture (DE/FR):** An approach involving two distinct, independent Random Forest models—one trained exclusively on German (DE) features and one on French (FR) features—was also tested. The goal was to capture country-specific consumption patterns without mutual interference, but this architecture equally failed to achieve a better official ranking than the single, combined model.

### 7.2.2 Future Improvement: Local Rank Regression

A concrete objective for future work is to enhance prediction within the terminal nodes (leaves) of the tree models. Since observations within a single leaf are **locally homogeneous**, implementing a custom leaf-fitting criterion that uses a **rank-based linear regression** on that subset could lead to better results. This approach aims to leverage the localized similarity of the data: while global linear regression yielded only $\sim 18\%$ Spearman correlation, fitting a rank-based regression locally could achieve a much higher correlation within the leaf. This strategy would combine the non-linear power of tree splitting with a target-specific rank metric at the final prediction stage.