# Forecast and Explain Electricity Price

## QRT Challenge Data

https://challengedata.ens.fr/participants/challenges/97/

# Agenda

1. Overview
2. Exploratory Data Analysis
3. Feature Engineering
4. Machine Learning
5. Conclusion

# Overview

# Project Introduction

This challenge asks participants to explain (not long-horizon forecast) daily changes in 24-hour electricity futures for France and Germany using multiple drivers: weather (temperature, rain, wind), commodity returns (gas, coal, carbon), generation mix (nuclear, hydro, solar, wind, gas, coal, lignite), and electricity usage/flows (consumption, residual load, imports/exports, DE/FR exchanges).

The dataset provides 1494 training and 654 test rows with 35 features, keyed by ID, DAY_ID (anonymized), and COUNTRY. Participants submit TARGET values (daily price variation) for test IDs. Evaluation is by Spearman's rank correlation between predictions and actual changes.

Detailed exploratory data analysis and feature engineering were performed on the raw dataset, and 5 machine learning models were used and compared on the engineered data. The entire IT implementation consists of 3 Jupyter notebooks, and the corresponding project are stored in the GitHub repository https://github.com/AInnovationQL/electricity_price

The best performance was by the LightGBM model that resulted with a score metric of **27.27%**. This score ranks **103** in the public ranking of this data challenge

# Overview of Notebooks

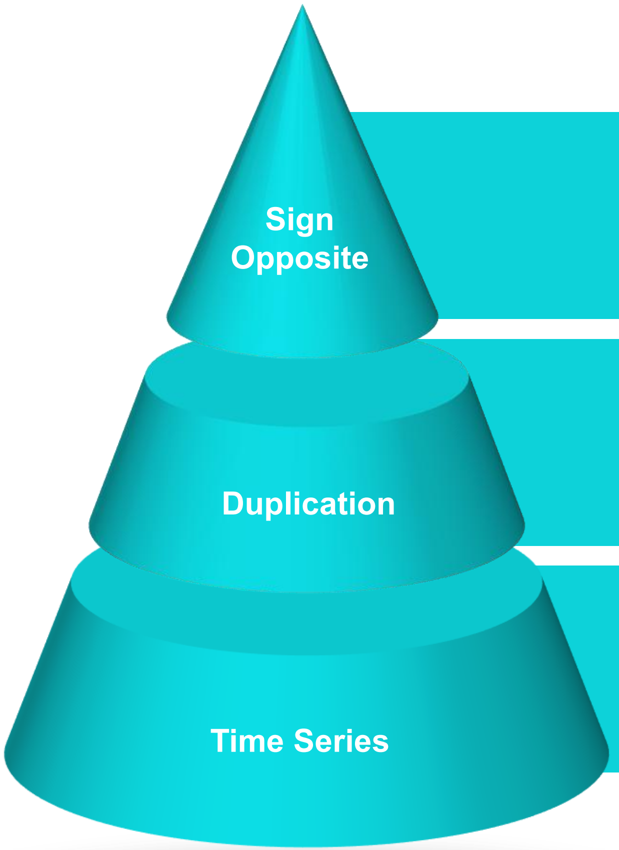
**EDA.ipynb:** Initial exploration of the provided data sets, including statistical summaries and visualizations to understand the distributions of various features and the target variable.

**feature_enginnering.ipynb:** Steps taken to clean the data, including handling missing values, outlier identification and treatment and feature engineering which involves the selection, manipulation and transformation of raw data into features used in supervised learning.

**ml_prediction.ipynb:** The main notebook, perform 5 different ML model (TabPFN, RandomForest, XGBoost, LightGBM and CatBoost) on the engineered data, and using the model with the best score for accuracy estimation of electricity price.

# Exploratory Data Analysis

# First Discovery of Data




The values of some column pairs (DE_FR_EXCHANGE vs. FR_DE_EXCHANGE, DE_NET_IMPORT vs. DE_NET_EXPORT and FR_NET_IMPORT vs. FR_NET_EXPORT) are sign opposite number of each other.

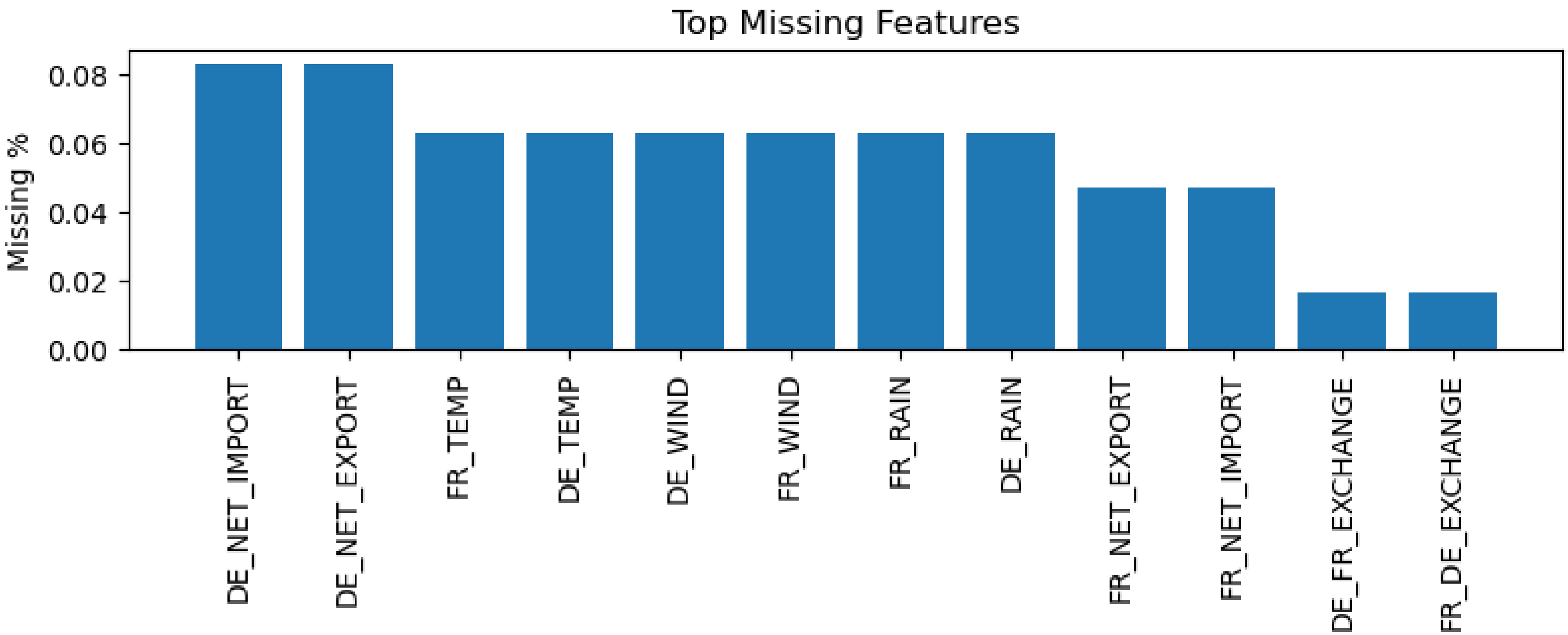
The data in each column (except for the column COUNTRY) is consistent for the same day (with identical DAY_ID).

By combining the training and testing data, each column of numerical data will yield a complete time series based on DAY_ID.

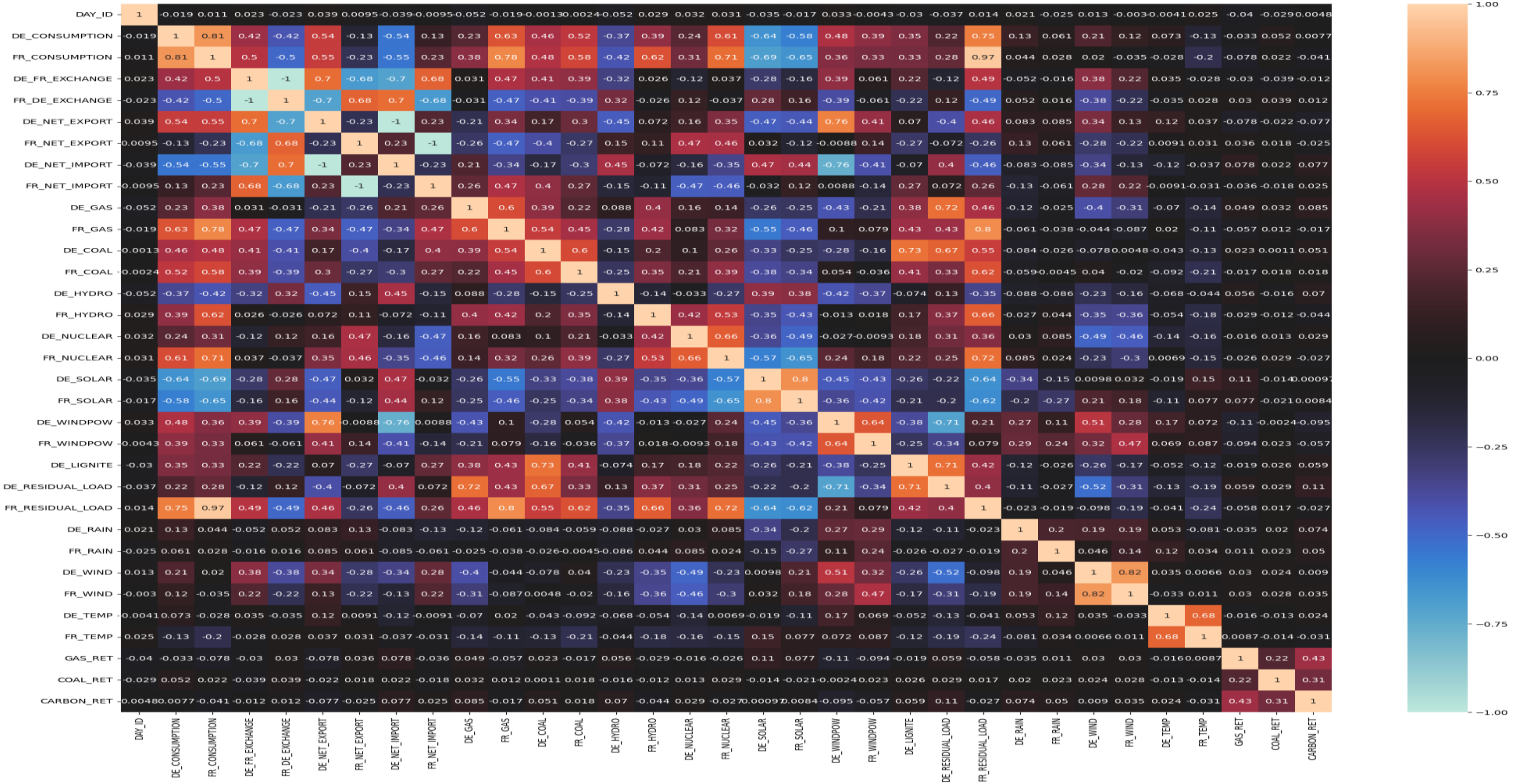
# EDA Notebook

- A concise Jupyter notebook for analysing the QRT electricity-price dataset.
- It loads X_train/Y_train (1494 rows) and profiles missingness (largest in DE_NET_IMPORT/DE_NET_EXPORT ≈ 8.3%; temperature/wind/rain ≈ 6.3%).
- It visualizes distributions, outliers, and feature–feature correlations (e.g., DE_FR_EXCHANGE vs. FR_DE_EXCHANGE, DE_NET_IMPORT vs. DE_NET_EXPORT and FR_NET_IMPORT vs. FR_NET_EXPORT show perfect anticorrelation).
- It reports feature–target correlations (modest signals from DE_NET_IMPORT, DE_NET_EXPORT and DE_WINDPOW) and repeats analyses per country (FR/DE).
- Time series work includes rolling means (7/14/30), ACF, and cross-correlation with TARGET.
- It compares simple imputers (median vs. k-nearest neighbor) using a quick HistGradientBoosting baseline and performs simple feature selection (variance threshold + 0.99 correlation filter), mutual information ranking (top: DE_RESIDUAL_LOAD, DE_WINDPOW), and PCA (≈18 components for ~95% variance).

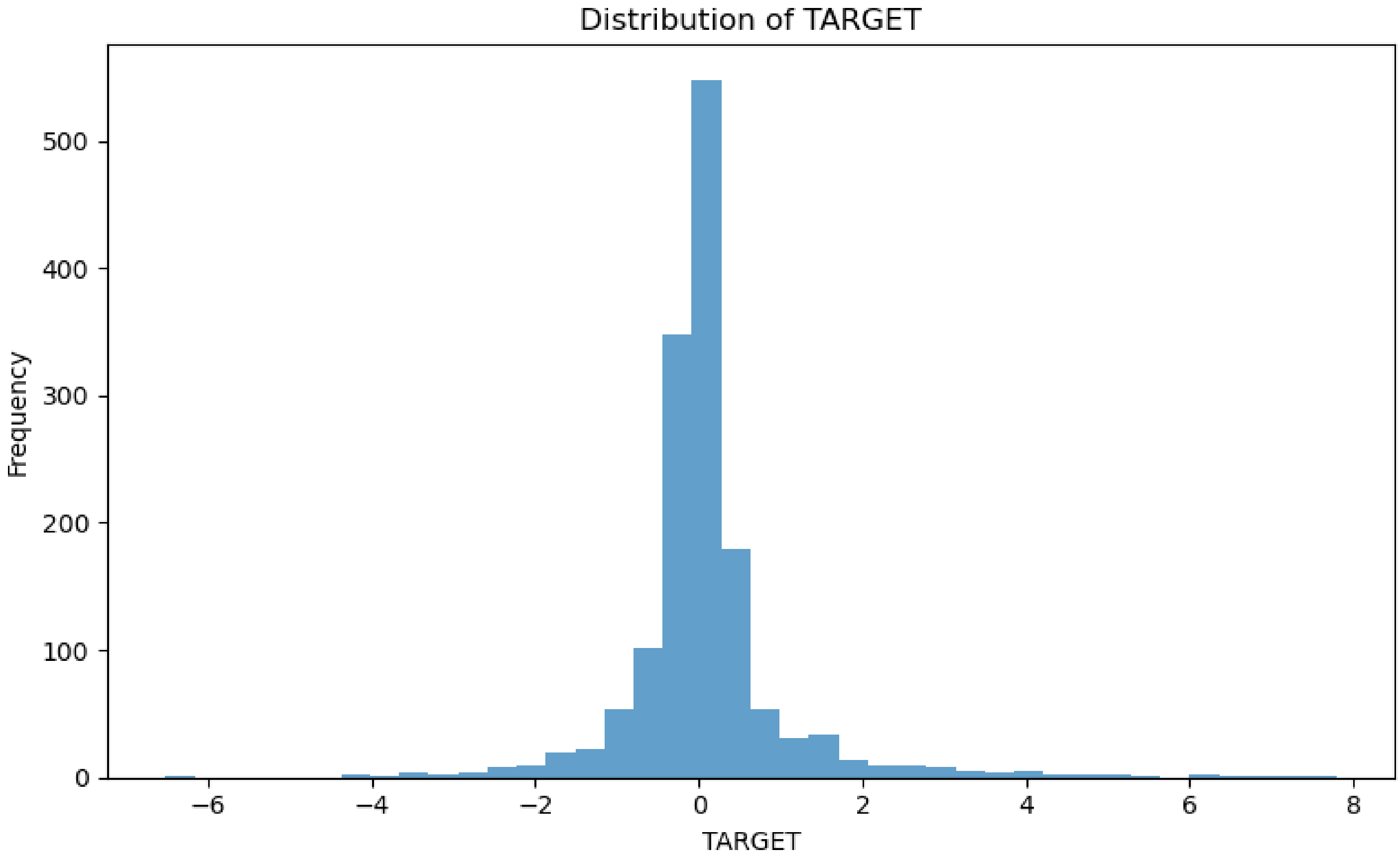
# Missing Data Profiling



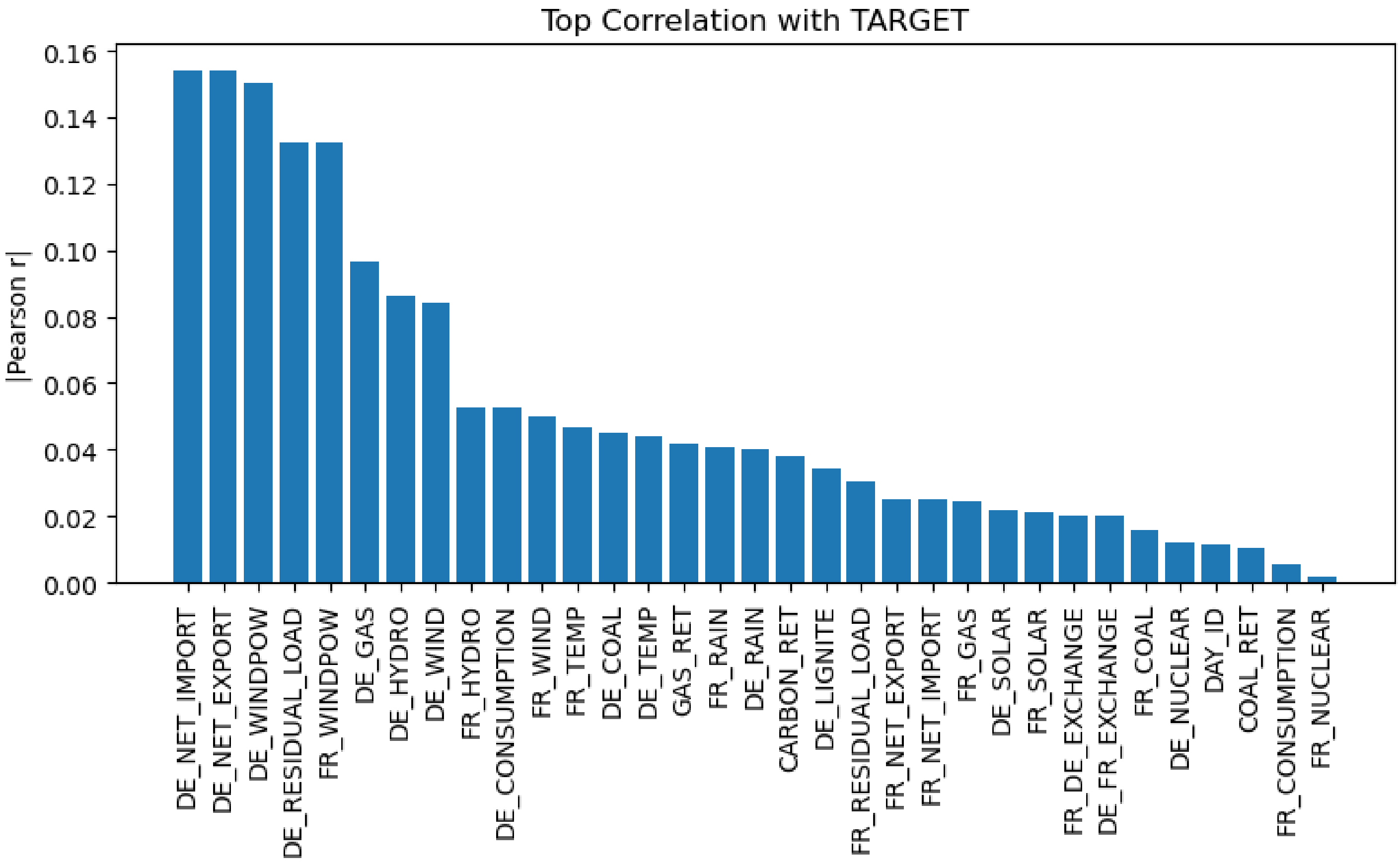
# Feature-Feature Correlations



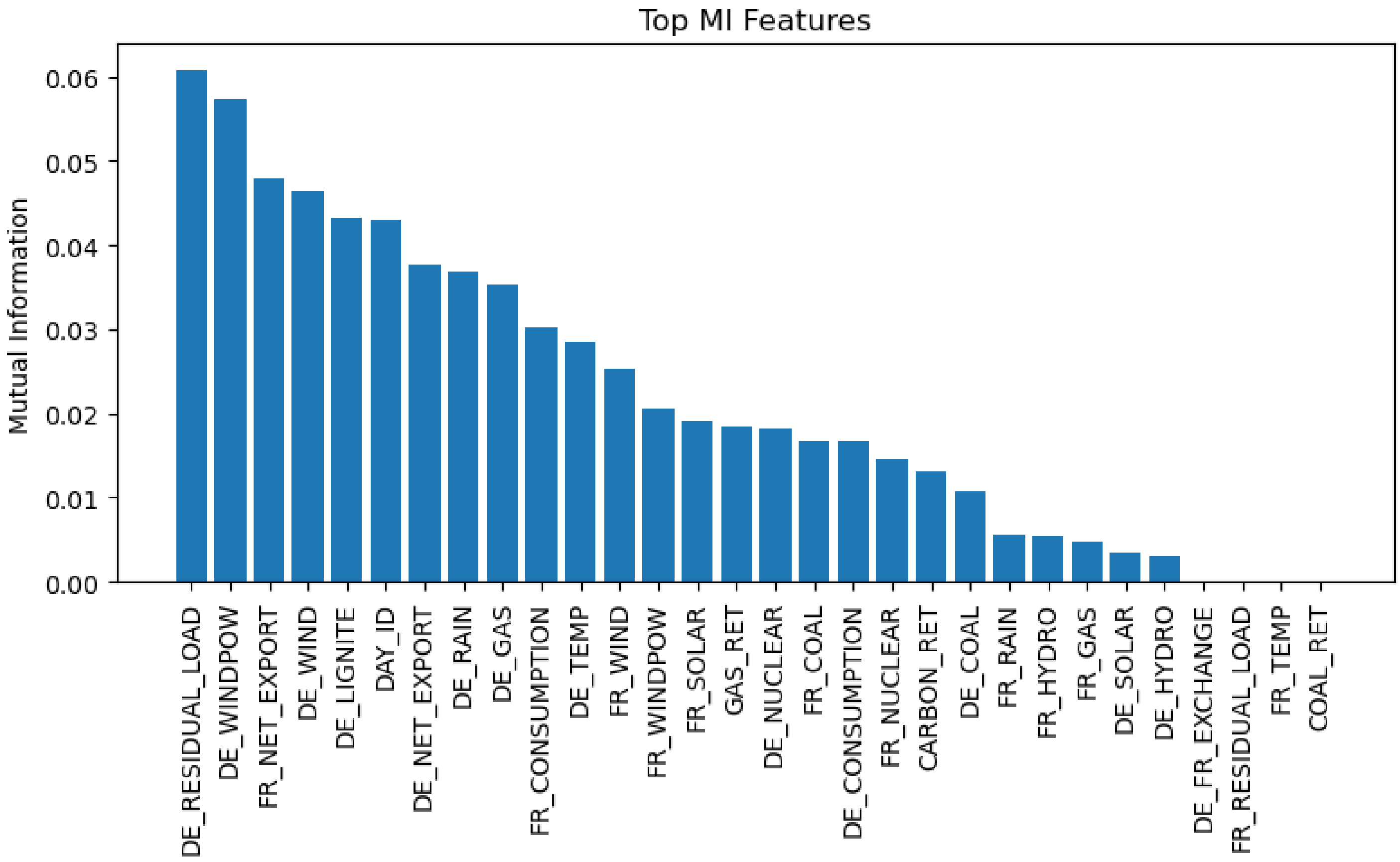
# Target Distribution



# Feature-Target Correlations



# Mutual Information



# Feature Engineering

# Feature Engineering Notebook

- **T**his Jupyter notebook builds a parameterized feature-engineering pipeline for the QRT electricity-price challenge data, aiming to improve supervised model performance.

- **I**t detects outliers via median absolute deviation z-scores and imputes missing values using simple imputation (null/mean/median), multiple imputation by chained equation, k-nearest neighbor, and interpolation/ extrapolation, with two modes: per-dataset or a combine time series aligned by DAY_ID to keep same-day values consistent.

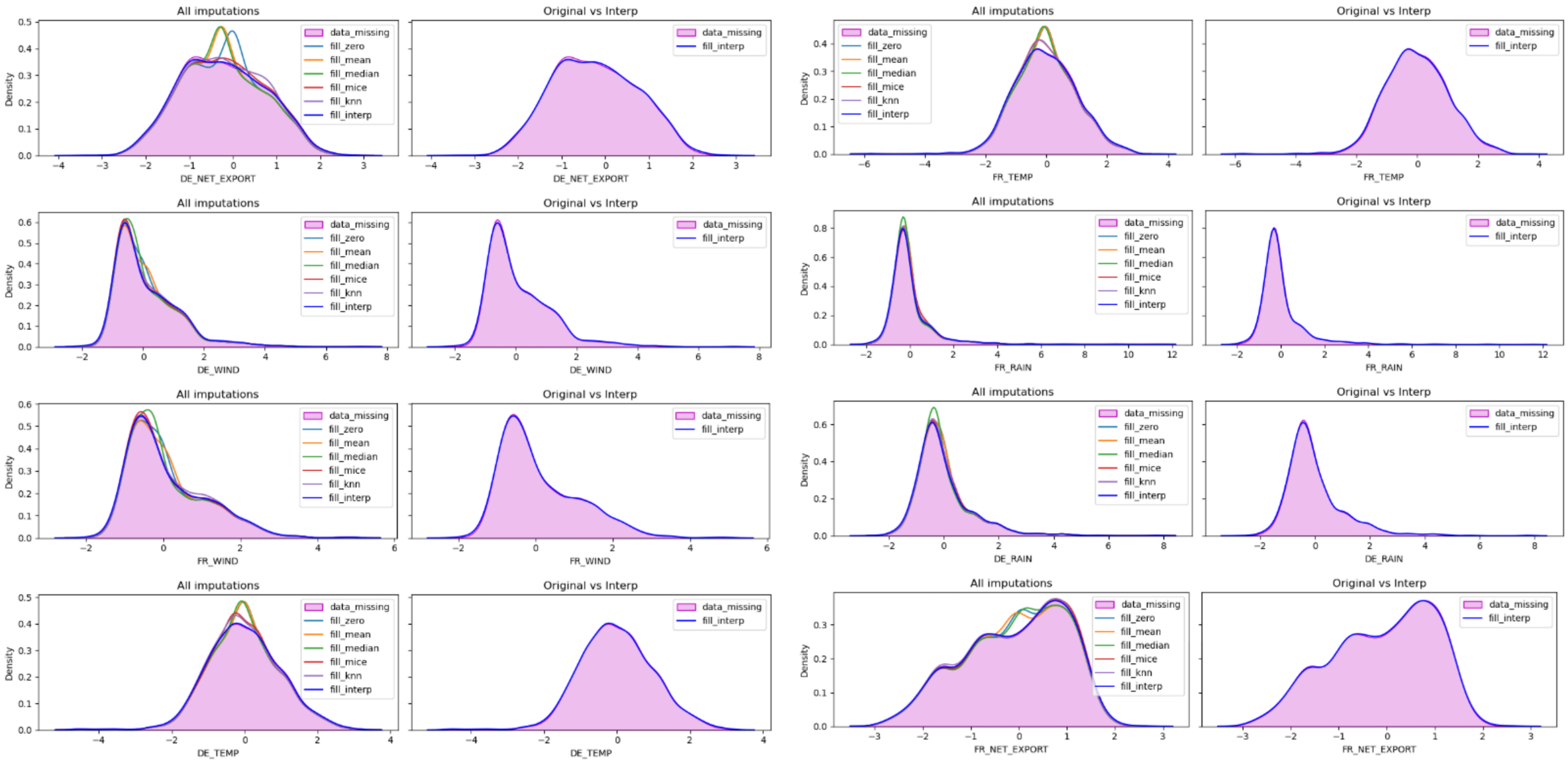
- **A**fter imputation, it computes correlations and mutual information and then creates features: residuals from highly correlated pairs, standardized golden characteristic features, group target-encodings (EU commodities, weather, energy usage, renewable energy and non-renewable energy), optional 7-day mean rollings, and SUM_/DIFF_ polynomials between DE and FR.

- **I**t then prunes by low mutual information and low target correlation, removes highly correlated features, drops columns with heavy missing/outliers (e.g. FR_COAL), and selects features according to null importance, producing a leaner feature vector.

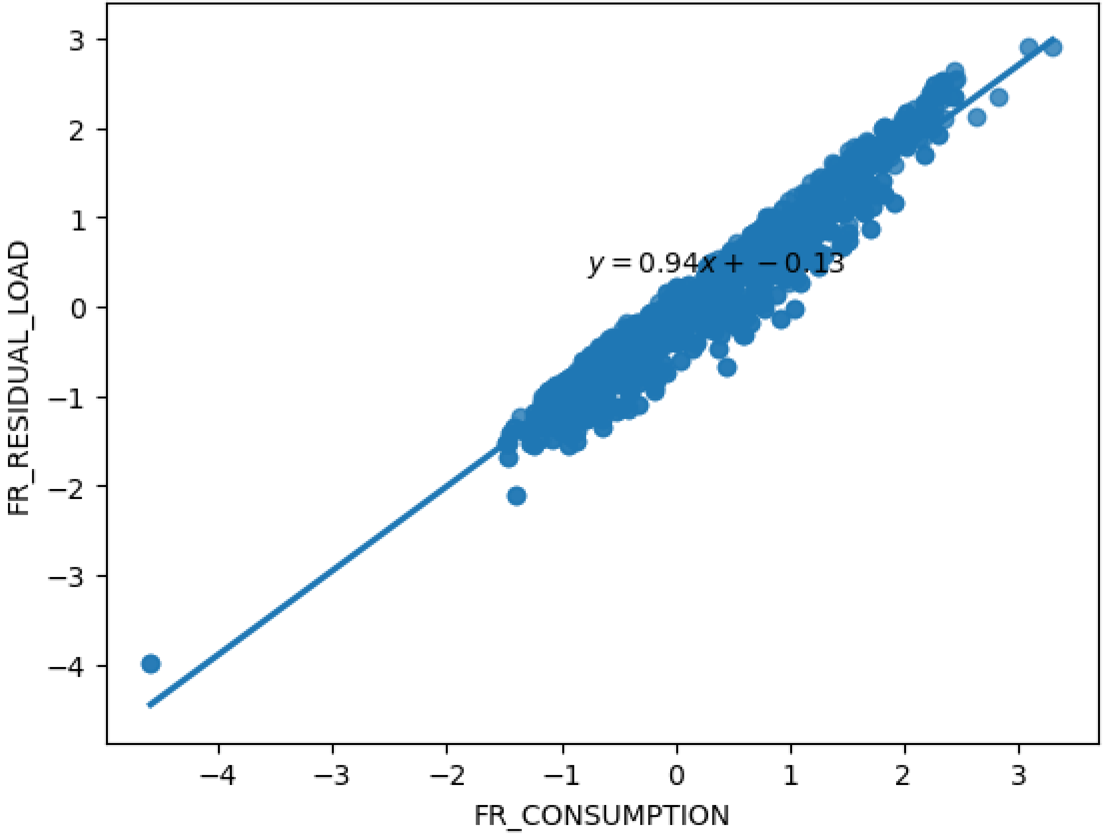
- **A** simple TabPFN regression benchmark is used to score-check the engineered features.

# Handling Missing Values

Interpolation performs best



# Creating Features Regression Residual




**F**or the strongly correlated features (e.g. FR_CONSUMPTION and FR_RESIDUAL_LOAD). The residual value is the difference between the observed value (FR_CONSUMPTION) and the estimated value (FR_RESIDUAL_LOAD). The column with the residual values can replace the column with the estimated value to offset strong correlation.

# Creating Features Target Encoding

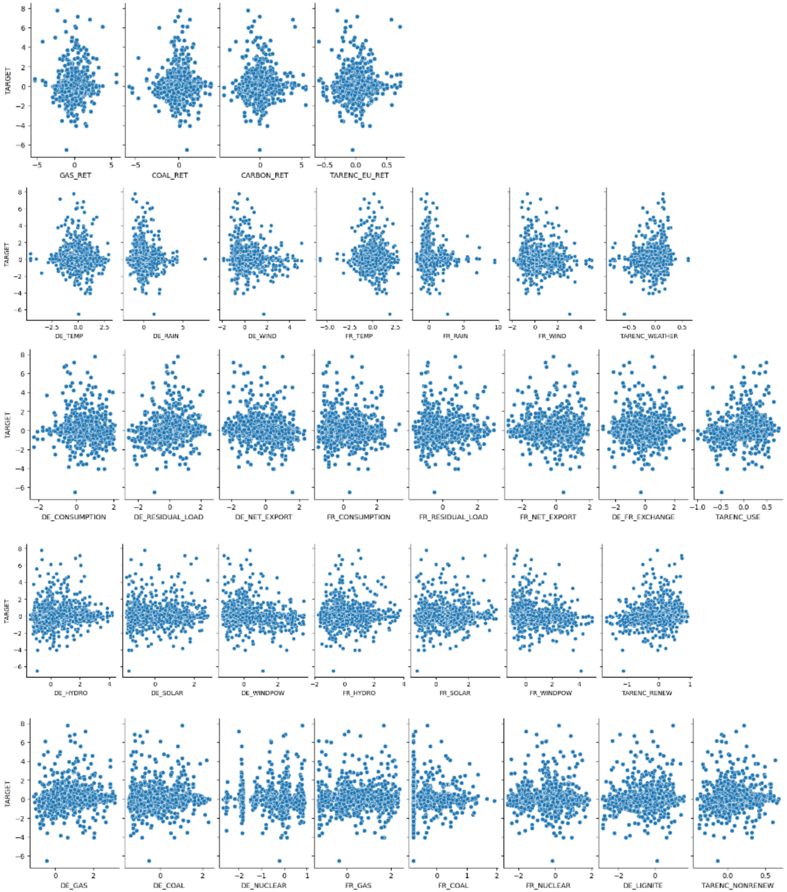
**EU Commodities Price Variations**

**Weather Measures**

**Electricity Usage Metrics**

**Renewable Energy Measures**

**Non-Renewable Energy Measures**



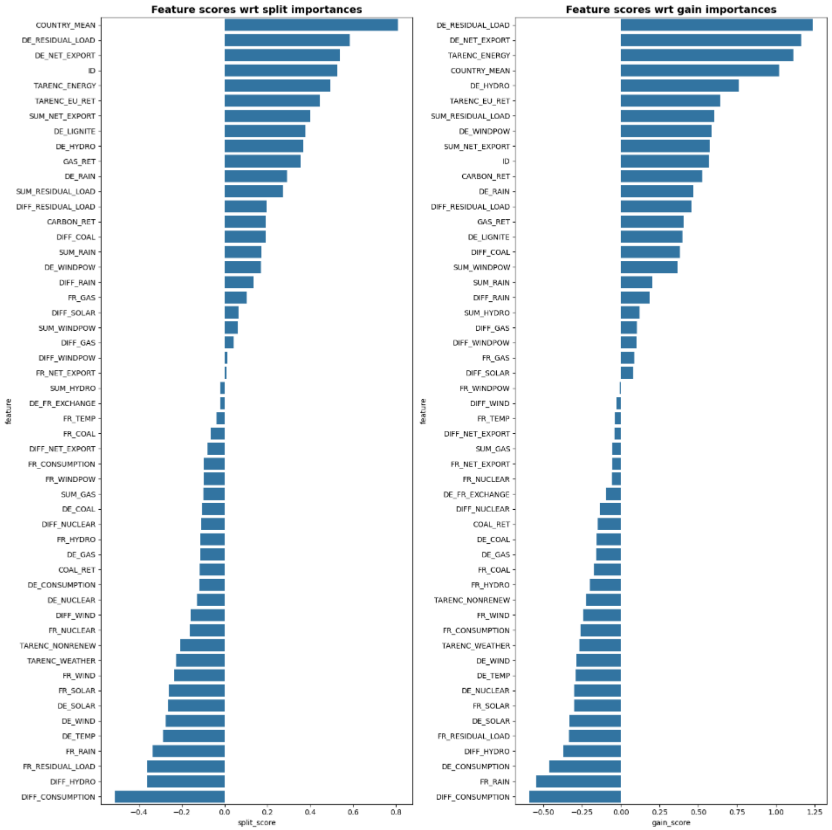
# Feature Deletion Correlation & Mutual Information

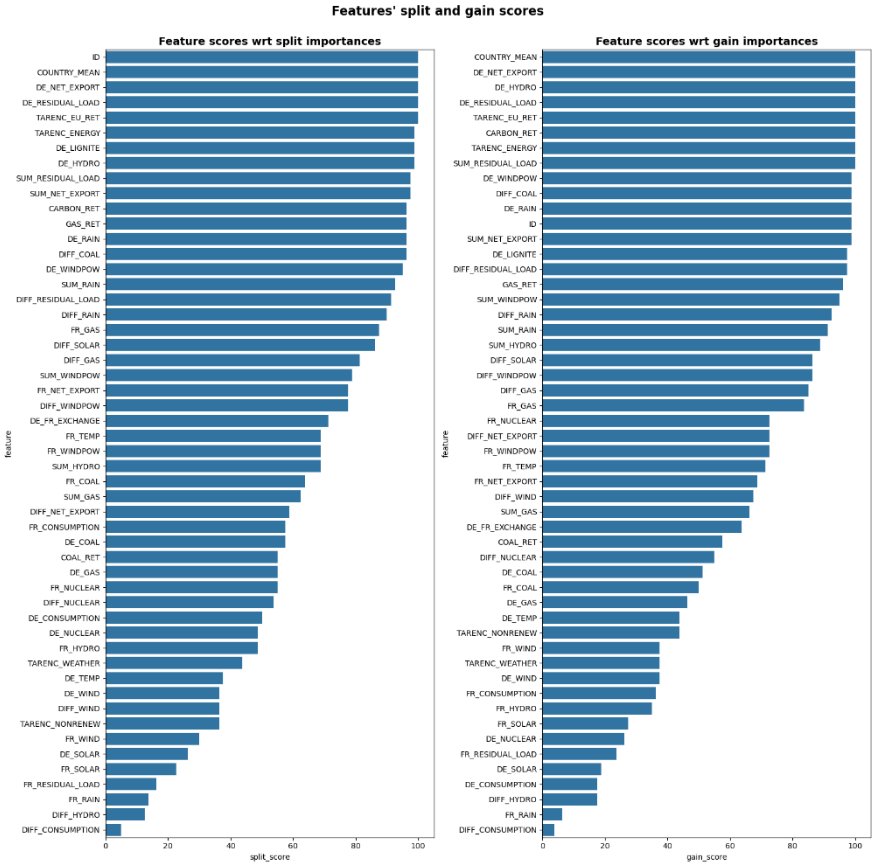
| | feat1 | feat2 | \|r\| | r |
|---|---|---|---|---|
| 0 | DE_WINDPOW | GOLD_DE_WINDPOW | 0.999968 | 0.999968 |
| 1 | CARBON_RET | GOLD_CARBON_RET | 0.999951 | 0.999951 |
| 2 | DE_RAIN | GOLD_DE_RAIN | 0.999848 | 0.999848 |
| 3 | FR_GAS | GOLD_FR_GAS | 0.999828 | 0.999828 |
| 4 | GAS_RET | GOLD_GAS_RET | 0.999723 | 0.999723 |
| 5 | DE_TEMP | GOLD_DE_TEMP | 0.999710 | 0.999710 |
| 6 | FR_RAIN | GOLD_FR_RAIN | 0.999658 | 0.999658 |
| 7 | DE_GAS | GOLD_DE_GAS | 0.999359 | 0.999359 |
| 8 | DE_SOLAR | GOLD_DE_SOLAR | 0.998920 | 0.998920 |
| 9 | DE_HYDRO | GOLD_DE_HYDRO | 0.998081 | 0.998081 |
| 10 | DE_CONSUMPTION | GOLD_DE_CONSUMPTION | 0.997565 | 0.997565 |
| 11 | FR_SOLAR | GOLD_FR_SOLAR | 0.997557 | 0.997557 |
| 12 | DE_RESIDUAL_LOAD | GOLD_DE_RESIDUAL_LOAD | 0.997431 | 0.997431 |
| 13 | DE_NET_EXPORT | GOLD_DE_NET_EXPORT | 0.996748 | 0.996748 |
| 14 | FR_CONSUMPTION | GOLD_FR_CONSUMPTION | 0.995941 | 0.995941 |
| 15 | FR_WIND | GOLD_FR_WIND | 0.991805 | 0.991805 |
| 16 | DE_WIND | GOLD_DE_WIND | 0.990358 | 0.990358 |
| 17 | DE_LIGNITE | GOLD_DE_LIGNITE | 0.981081 | 0.981081 |
| 18 | FR_CONSUMPTION | SUM_CONSUMPTION | 0.959291 | 0.959291 |
| 19 | GOLD_FR_CONSUMPTION | SUM_CONSUMPTION | 0.955112 | 0.955112 |
| 20 | FR_WIND | SUM_WIND | 0.953515 | 0.953515 |
| 21 | DE_CONSUMPTION | SUM_CONSUMPTION | 0.952950 | 0.952950 |
| 22 | DE_WIND | SUM_WIND | 0.951312 | 0.951312 |
| 23 | GOLD_DE_CONSUMPTION | SUM_CONSUMPTION | 0.950420 | 0.950420 |
| 24 | GOLD_FR_WIND | SUM_WIND | 0.944515 | 0.944515 |
| 25 | FR_CONSUMPTION | FR_RESIDUAL_LOAD | 0.941636 | 0.941636 |
| 26 | GOLD_DE_WIND | SUM_WIND | 0.941384 | 0.941384 |
| 27 | FR_RESIDUAL_LOAD | GOLD_FR_CONSUMPTION | 0.935795 | 0.935795 |
| 28 | GOLD_DE_NET_EXPORT | TARENC_USE | 0.922120 | -0.922120 |
| 29 | DE_NET_EXPORT | TARENC_USE | 0.921848 | -0.921848 |
| 30 | TARENC_RENEW | TARENC_ENERGY | 0.913896 | 0.913896 |
| 31 | TARENC_RENEW | SUM_WINDPOW | 0.904811 | -0.904811 |
| 32 | FR_TEMP | SUM_TEMP | 0.904375 | 0.904375 |
| 33 | FR_RESIDUAL_LOAD | SUM_CONSUMPTION | 0.900091 | 0.900091 |

| | TARGET |
|---|---|
| GOLD_DE_NUCLEAR | 0.009531 |
| GOLD_FR_NET_EXPORT | 0.009037 |
| FR_NET_EXPORT | 0.008699 |
| DE_NUCLEAR | 0.007052 |
| SUM_NUCLEAR | 0.006183 |
| DAY_ID | 0.003901 |
| FR_NUCLEAR | 0.003619 |
| DIFF_TEMP | 0.001500 |
| GOLD_FR_NUCLEAR | 0.001109 |

| | MI |
|---|---|
| DE_FR_EXCHANGE | 0.0 |
| FR_RESIDUAL_LOAD | 0.0 |
| GOLD_FR_HYDRO | 0.0 |
| GOLD_FR_RESIDUAL_LOAD | 0.0 |
| GOLD_FR_COAL | 0.0 |
| FR_TEMP | 0.0 |
| GOLD_DE_COAL | 0.0 |
| RES_FR_RESIDUAL_LOAD | 0.0 |
| COAL_RET | 0.0 |
| GOLD_FR_NUCLEAR | 0.0 |
| GOLD_FR_WINDPOW | 0.0 |
| GOLD_FR_TEMP | 0.0 |
| GOLD_COAL_RET | 0.0 |
| SUM_SOLAR | 0.0 |
| SUM_COAL | 0.0 |

# Feature Deletion Null Importance



# Machine Learning

# Machine Learning Notebook

- **T**his notebook automates feature engineering by calling a parameterized "*feature_ engineering.ipynb*" (via Papermill/Scrapbook)

- **I**t selects the best setup using spearman correlation and root mean square deviation, benchmarks five regressors — TabPFN, Random Forest, XGBoost, LightGBM and CatBoost. TabPFN, Random Forest and LightGBM perform better.

- **A**t least the notebook runs the selected machine learning regression model with cross validation on GPU, generates test predictions, and saves the results as csv files.

# Regression Models

## TabPFN

<<tabpfn>>
TabPFNRegressor

**T**abPFN is a transformer-based foundation model for tabular data that leverages prior-data based learning to achieve strong performance on small tabular regression tasks without requiring task-specific training.

## Random Forest

<<sklearn>>
RandomForestRegressor

**T**he bootstrapping Random Forest algorithm combines ensemble learning methods with the decision tree framework to create multiple randomly drawn decision trees from the data, averaging the results to output a new result that often leads to strong predictions.

## XGBoost

<<xgboost>>
XGBRegressor

**X**GBoost, which stands for extreme gradient boosting, is an optimized and scalable implementation of gradient boosting for tree-based models. It is designed for both efficiency and performance and is widely used for large-scale machine learning tasks such as classification and regression.

## LightGBM

<<lightgbm>>
LGBMRegressor

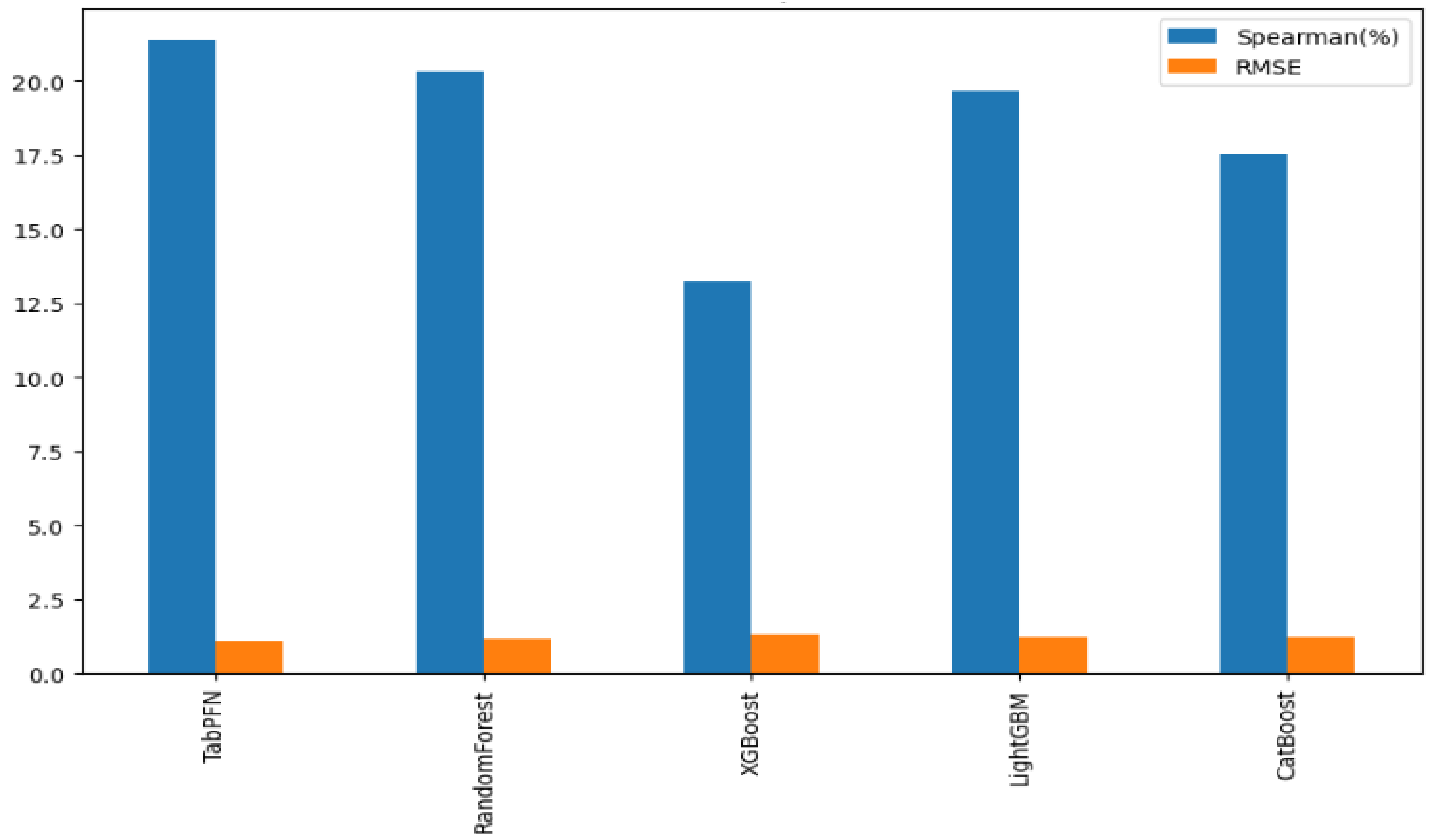
**L**ightGBM, short form for light gradient boosting machine, is an gradient boosting framework that uses tree-based machine learning algorithms. It is designed to be efficient and scalable for large-scale machine learning tasks, such as classification and regression.

## CatBoost

<<catboost>>
CatBoostRegressor

**C**atBoost, which stands for categorical boosting, is a supervised machine learning method that is used by the train using AutoML tool and uses decision trees for classification and regression.

# Model Comparing



# Conclusion

# Explain Electricity Price

- ## Cross-border position (dominant)

**DE_NET_IMPORT ↑ → Price ↑ (corr ≈ +0.15)**
**DE_NET_EXPORT ↑ → Price ↓ (corr ≈ −0.15)**
Germany tends to export when it has surplus low-cost wind and import when supply is tight. This matches the strong correlates: DE_NET_EXPORT and DE_NET_IMPORT are almost perfect opposites, and both are tightly tied to DE_WINDPOW (|corr| ≈ 0.76). European market coupling transmits these imbalances quickly to prices.

- ## Residual load & demand

**DE_RESIDUAL_LOAD ↑ → Price ↑ (corr ≈ +0.13; MI tops ≈ +0.06)**
Residual load is demand after renewables. When it's high, gas and coal units more often set the marginal price, pushing the electricity price up.

- ## Wind & other renewables (price-depressing)

**DE_WINDPOW, DE_WIND, FR_WINDPOW, FR_WIND ↑ → Price ↓ (corrs ≈ −0.15 to −0.05; all show up with non-trivial MI)**
Because wind power is almost free to produce, it adds lots of cheap supply, reduces the demand left for fossil plants (residual load), makes Germany export more, and lowers prices in both Germany and France.

- ## Non-renewables

**GAS_RET ↑ → Price ↑ (corr ≈ +0.04)**
**DE_GAS, DE_COAL ↑ → Price ↑ (corrs ≈ +0.10 and +0.05)**
If gas gets more expensive, or we use more gas and coal plants, the last unit of power costs more, so the market price rises. That's the merit-order effect.

- ## Weather (demand & hydro availability)

**TEMP ↓ (colder) → Price ↑ in both countries (negative corr)**
**RAIN ↑ → Price ↓ (negative corr)**
In a heating-dominated system, colder days raise demand and residual load. More rainfall typically improves hydro availability and eases prices.

- ## Seasonality

DAY_ID has meaningful mutual information. That signals non-linear seasonality and structural regimes (e.g., policy shifts, market phases) that a linear correlation won't capture.

# Lesson Learned

**Strategy**

**D**ue to a lack of experience in data prediction models, the strategy involved trying various combinations of feature engineering and machine learning models (even deep learning) after gaining an initial understanding of the data to achieve relatively good results. However, the solution that ultimately yielded relatively good results was the relatively simple model combined with basic feature engineering.

**Lesson**

**S**ometimes we should apply the Occam's Razor, a principle that suggests the simplest explanation or solution is usually the most likely to be correct.

# Thank You

For Your Attention