Project Forecasting (ML & Time Series)

Elie Menassa

Introduction

This project has as its objective to evaluate the capacity of different statistical models and machine learning to forecast the EUR/CHF exchange rate on a short-term horizon (one day in advance). The foreign exchange market, and in particular the EUR/CHF pair, constitutes a relevant field of study: the data are continuous, liquid, and widely used in applications of treasury management, hedging, and financial risk analysis.

To implement this study, we rely on a dedicated Python environment, integrating the main libraries of data processing and modeling (pandas, statsmodels, scikit-learn, yfinance, etc.), as well as a project architecture guaranteeing reproducibility (separation raw/processed data, scripts, notebooks, and reports).
The approach adopted follows a progression in nine main steps:

The approach adopted follows a progression in nine main steps:

1. **Definition and collection of data**
   Formulation of the research question, definition of the target variable (daily log returns), choice of the period and frequency, establishment of a reproducible project architecture.
2. **Preparation of data**
   Cleaning of data, detection of extreme values, study of distributions, normality and basic statistical properties. Chronological splitting into training and test sets, and preparation of variables for modeling.
3. **First models and evaluation tools**
   Implementation of performance metrics (RMSE, MAE, MAPE), testing of naïve baselines and introduction of statistical models such as ARIMA, ARIMAX and GARCH, as well as linear and tree models.
4. **In-depth baselines**
   Detailed study of simple reference models (Naïve, Moving Averages, Simple Exponential Smoothing), in order to establish a minimal performance bound and to better understand the limits of trivial prediction.
5. **Classical statistical models**
   Deepening of ARIMA and ARIMAX models, diagnostic tests and validation of assumptions of stationarity and residuals. These models serve as historical references in the literature on time series.
6. **Baselines in machine learning**
   Evaluation of simple linear models and comparison with SES, in order to test whether basic machine learning methods bring added value compared to classical statistical approaches.

7. **Advanced machine learning models**
   Deployment of more sophisticated models such as Random Forest and XGBoost, analysis of variable importance and comparison with linear and statistical approaches.
8. **Global synthesis**
   Comparison of performances of different models (statistical vs machine learning), identification of relevant use cases in finance and discussion of the limits of the project.
9. **Conclusion and perspectives**
   Recapitulation of key lessons, identification of methodological limits and proposal of a roadmap: integration of exogenous variables, modeling of volatility and implementation of interactive visualization and reporting tools.

Thus, this project is not limited to a simple comparison of models, but proposes a complete, structured and reproducible end-to-end approach, going from the collection of data to the discussion of application perspectives.

CHAPTER 1 – DEFINITION & DATA COLLECTION 7

1.1 Define the question & the target 7

1.2 Define frequency & period 7

1.3 Choose the source & ticker 8

1.4 Project organization (data directory) 9

1.5 Setting up the Python environment 10

1.7 Quick checks (data quality) 11

1.8 Final documentation of the collection 12

CHAPTER 2 : DATA PREPARATION 14

2.1 Loading Raw Data 14

2.2 Transformation & Outlier Detection 15

2.3 Statistical Tests & Visualization of Log-Returns 16

2.4 Preliminary Analysis of Returns 18

2.5 Chronological Train/Test Split 20

2.6 ML Feature Preparation & Scaling 21

2.7 Documentation & Saves 21

2.8 Synthesis of Data Preparation 25

CHAPTER 3 - FIRST MODELS AND EVALUATION TOOLS 26

3.0 Evaluation Utilities 26

3.1 Baselines (Unscaled Returns) 27

3.2 ARIMA / ARIMAX (Classical Statistical Models) 28

3.3 AR(1)–GARCH(1,1): Returns & Volatility 30

3.4 Linear ML Models (Ridge & Lasso) 32

3.5 Tree-Based Models (RandomForest, XGBoost) 33

3.6 – Model Comparison & Leaderboard 35

3.7 – Results & Applied Finance Interpretation 36

CHAPTER 4 : BASELINE FORECASTING 38

4.1 Setup 38

4.2 Naïve Forecast 38

4.3 Moving Average Forecasts 40

4.3 Simple Exponential Smoothing (SES) 42

4.4 — Comparison & Results 44

CHAPTER 5: CLASSICAL STATISTICAL MODELS 46

5.1 SETUP 45

5.2 – ARIMA 47

5.3 – SARIMA 50

5.4 – Holt-Winters (Exponential Smoothing) 53

CHAPTER 6 - MACHINE LEARNING BASELINES 56

6.1 – Feature Engineering (Explanatory Variables) 56

6.2 – Linear Regression 57

6.3 – Cross-Validation 60

6.4 – SES vs Linear Regression 61

CHAPTER 7 - ADVANCED ML MODELS 63

7.1 – Random Forest 64

7.2 – Gradient Boosting (sklearn) 66

7.3 – XGBoost & LightGBM (Advanced Boosters) 69

7.4 – Ensembles (Stacking & Blending) **71**

7.5 Advanced Diagnostics **73**

7.6 Walk-Forward Backtesting **76**

7.7 – Final Comparison (Advanced ML vs. References) **78**

7.8 – Chapter Conclusion **81**

CHAPTER 8 — FINAL EVALUATION **84**

8.1 Introduction **82**

8.2 Consolidated Metrics **85**

8.3 Comparative Visualizations **87**

8.4 Chapter Conclusion **88**

CHAPTER 9 - FINAL SYNTHESIS & BUSINESS VALUE **89**

9.1 Executive Summary **89**

9.2 Key Results & Business Interpretation **90**

9.3 Limitations & Risks **91**

9.4 Business Impact & Use Cases **93**

9.5 — Short Roadmap (3–5 High-Leverage Actions) **94**

9.6 — Final Conclusion: EUR/CHF Project (2015–2025) **97**

Chapter 1 – Definition & Data Collection

1.1 Define the question & the target

Objective

Determine whether the daily evolution of the EUR/CHF exchange rate can be forecasted with a short-term horizon (T+1).

Setup

Target variable: daily logarithmic return :

$$r_{t+1} = \log(P_{t+1}) - \log(P_t)$$

Horizon: one day ahead (T+1).

Methodology

- Use returns rather than prices → better stationarity.
- Choice of a simple and realistic horizon for treasury and short-term trading.

Analysis & Interpretation

The T+1 approach allows us to test models without excessive complexity while remaining relevant for real-world financial applications.

Conclusion

The target is set: one-day-ahead log returns of EUR/CHF.

1.2 Define frequency & period

Objective

Select a frequency and history consistent with modeling.

Setup

- Frequency: daily closes (business days).
- Period: 2015-01-01 → 2025-09-22 (≈ 2,798 business days).

Methodology

- Exclude weekends.
- Use a business-day calendar to avoid gaps.

Results

- Minimum: 0.924
- Maximum: 1.203
- Mean: 1.054
- Standard deviation: 0.071

Analysis & Interpretation

A 10-year period ensures a balance between robustness (long history) and current relevance.

Conclusion

The dataset is sufficiently long and balanced to train and test models.

1.3 Choose the source & ticker

Objective

Ensure simple, free, and reproducible collection.

Setup

- Source: Yahoo Finance via *yfinance*.
- Ticker: EURCHF=X.

Methodology

- Direct download with `yf.download`.
- No API key required.

Results

Data successfully loaded: 2,798 observations.

Analysis & Interpretation

Yahoo Finance is suitable for an academic/portfolio project.

Conclusion

The source and ticker are validated for EUR/CHF.

1.4 Project organization (data directory)

Objective
Guarantee clarity and reproducibility.

Setup
Proposed structure:

```
eur_chf_forecasting/
│
├── data/
│   ├── raw/            # original downloaded data (Yahoo Finance, untouched)
│   ├── processed/      # cleaned datasets, train/test splits, engineered features
│
├── src/                # Python scripts (Part1.py … Part8.py)
│
│
│
├── results/
│   ├── baseline/       # outputs from Naïve, MA, SES
│   ├── stats/          # ARIMA, SARIMA, Holt-Winters results
│   ├── ml_baseline/    # Linear Regression, SES vs ML comparisons
│   ├── ml_advanced/    # RF, XGB, LGBM, Stacking, Blending
│   ├── final/          # consolidated evaluation (Part 8)
│
├── reports/
│   ├── Report.pdf      # final report (exported from Word/LaTeX)
│   ├── figs/           # main figures (leaderboard, forecasts, residuals)
│
├── environment.yml     # Conda environment for reproducibility
├── requirements.txt    # alternative (pip) dependencies
├── LICENSE             # license file (e.g., MIT)
└── README.md           # project documentation (this file)
```

Analysis

- Separation *raw/processed* → traceability.
- Separation *notebooks/src* → cleanliness.

Conclusion

The structure ensures reproducibility and facilitates sharing.

1.5 Setting up the Python environment

Objective

To ensure reproducibility and avoid dependency conflicts, we created a dedicated Conda environment named *forecast* with Python 3.11.

Setup

- Conda env: *forecast*, Python 3.11.
- Packages: numpy, pandas, matplotlib, scikit-learn, statsmodels, pyarrow, yfinance, plotly.

Results

All imports validated, environment operational.

Analysis & Interpretation

This environment covers the entire pipeline (collection → ML → visualization).

Conclusion

The Python environment is ready for the next steps.

1.6 Collection of EUR/CHF

Objective

Download and clean the main EUR/CHF series.

Methodology

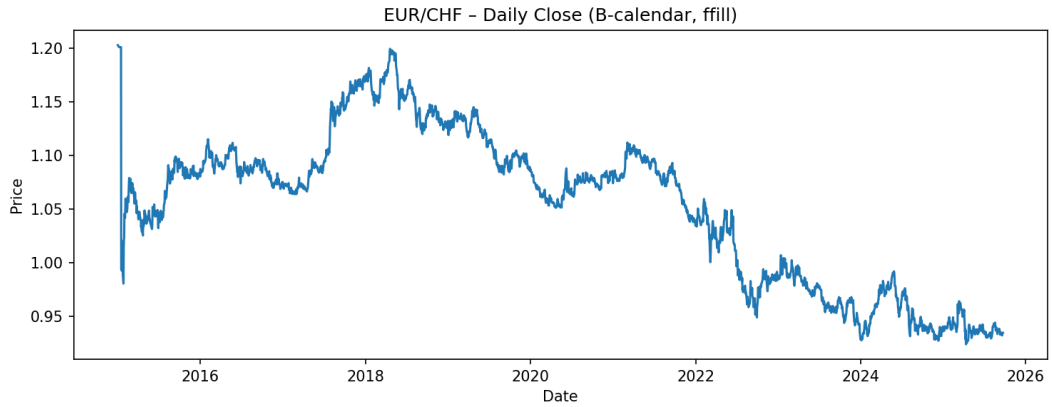
- Source: Yahoo Finance, via the Python library *yfinance*.
- Ticker: EURCHF=X (EUR/CHF exchange rate).
- Period: from 01/01/2015 up to the last available business day.
- Frequency: Daily (business day calendar).
- Target variable: the adjusted closing price, renamed *price*.

Results

- 2,798 observations (2015–2025).
- Min: 0.924, Max: 1.203, Mean: 1.054.
- No missing values or duplicates.

Analysis & Interpretation

The data are clean and suitable for transformation into returns.



Curve of the EUR/CHF price (2015–2025)

Conclusion

The raw EUR/CHF series is ready for Step 2.

1.7 Quick checks (data quality)

Objective

After the collection and cleaning of the EUR/CHF dataset, it is essential to perform quick checks in order to verify its coherence, integrity, and compatibility with the future modeling steps.

Methodology

Three types of verifications were carried out:

- Monotonicity of the index: ensure that the ariale index is strictly increasing.
- Presence of missing business days: compare the DataFrame index to the complete business-day calendar between the extreme dates.
- Visual and statistical inspection: display of the first and last observations, as well as a statistical summary of the prices.

Results

- Monotonicity: True (the index is strictly increasing).
- Missing business days: 0 (no gaps after reindexing and forward-fill).
- Missing values in the *price* column: 0.
- Duplicates: 0.
- Descriptive statistics:
  - Mean ≈ 1.05
  - Minimum ≈ 0.92
  - Maximum ≈ 1.20
  - Standard deviation ≈ 0.07
- A chart of the EUR/CHF pair (2015–2025) was generated for visual inspection.

These simple but fundamental checks confirm that the EUR/CHF dataset is clean, complete, and consistent with a business-day calendar. The absence of missing values or duplicates guarantees that the transformation into log returns (Part 2) can be done without additional correction.

Conclusion

The EUR/CHF dataset is validated for the next steps.

1.8 Final documentation of the collection

Objective

Provide a clear and documented synthesis of the collected series, in order to ensure traceability and reproducibility of the data pipeline.

Methodology

A Python script automatically generates a summary table gathering the essential metadata (series name, ticker, number of observations, covered period, min/max prices, frequency).

The following Table 1 presents the comparative documentation:

Name	Ticker	Rows	Start	End	Min price	Max price	Frequency
EUR/CHF (FX)	EURCHF=X	2798	2015-01-01	2025-09-22	0.92	1.20	B (daily)

Analysis & Interpretation

This table serves as a reference for the rest of the project. Even if the core of the analysis is focused on EUR/CHF, the pipeline structure is designed to accommodate other assets if necessary.

Conclusion

This summary table constitutes a quick entry point for any project user. It allows one to verify the covered period and data completeness before starting the statistical transformations of Chapter 2.

Chapter 2 : Data Preparation

2.1 Loading Raw Data

Objective

Ensure that the EUR/CHF raw dataset is correctly loaded into memory with a clean DatetimeIndex, no missing values, and consistent formatting to serve as the foundation for further transformations.

Setup

> **Column used**: Adjusted closing price, standardized as `price`.

Methodology

1. Load the dataset using `pandas.read_csv` with explicit date parsing.
2. Set the `Date` column as the DataFrame index.
3. Standardize column names for clarity (`price`).
4. Run validation checks: row count, date range, monotonic index, duplicates, missing values.

Results

- Rows: **2,798**
- Date range: **2015-01-01** → **2025-09-22**
- Index monotonic: **True**
- Missing values: **0**
- Duplicates: **0**
- Non-positive prices: **0**

Analysis & Interpretation

The raw dataset is complete, continuous on business days, and free from anomalies. With 10 years of history and nearly 2,800 observations, it provides a robust basis for statistical modeling.

Conclusion

The EUR/CHF dataset is successfully loaded, validated, and standardized into a clean DataFrame ready for transformation.

2.2 Transformation & Outlier Detection

Objective

Transform the raw price series into logarithmic returns for stationarity, and identify potential outliers that could bias statistical analysis.

Methodology

1. Compute log-returns:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

2. Generate descriptive statistics (mean, std, skewness, kurtosis).

3. Detect outliers using:

    - **Z-score method**: |Z| > 3.
    - **IQR rule**: outside [Q1 − 1.5×IQR ; Q3 + 1.5×IQR].

Results

- Mean: **≈ −0.00009** (close to 0).
- Std: **0.0049** (≈ 0.5% daily volatility).
- Skewness: **−19.8** (extremely left-skewed).
- Kurtosis: **764** (heavy-tailed).
- Outliers detected: **8 (Z-score)**, **103 (IQR)**.

Analysis & Interpretation

- The extreme skewness and kurtosis reflect heavy-tailed distribution typical of FX returns.
- The most extreme negative return corresponds to January 2015 (Swiss National Bank decision to abandon the EUR/CHF floor).
- For modeling, such heavy tails confirm the need for robust models (GARCH, fat-tailed distributions).

Conclusion

Log-returns were computed and validated. Outliers have been flagged but not removed, ensuring future models account for extreme shocks.

2.3 Statistical Tests & Visualization of Log-Returns

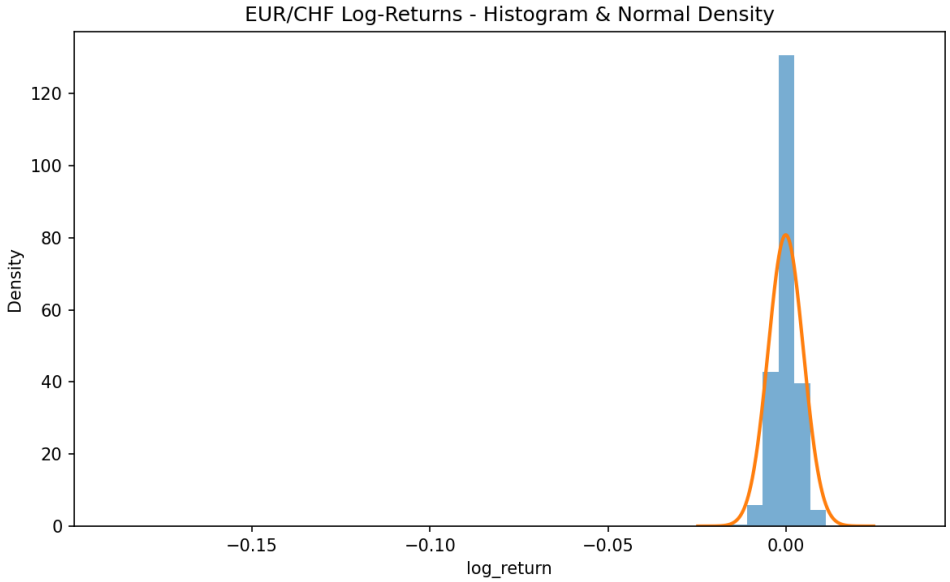
Objective

Evaluate the statistical distribution of log-returns, test for normality, and visualize their properties.

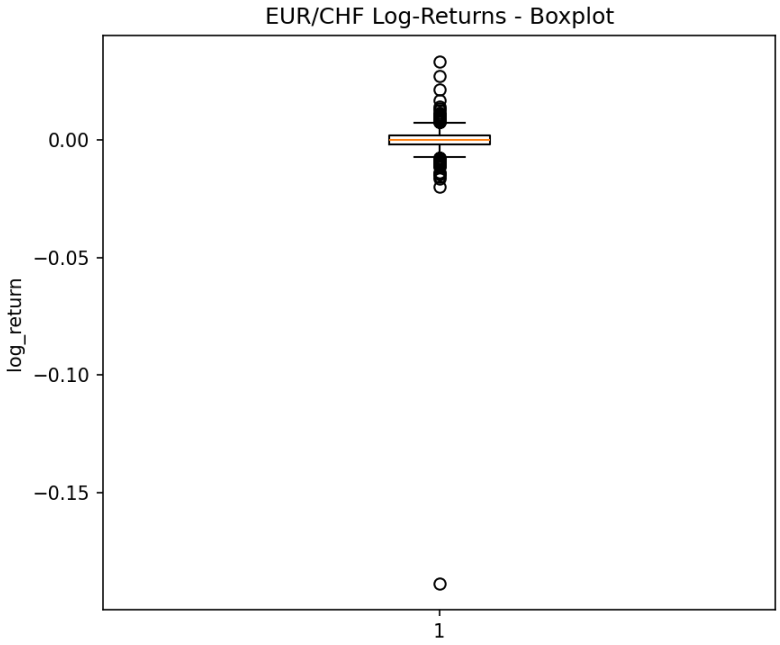
Methodology

1. Compute descriptive statistics (again, for confirmation).
2. Apply Shapiro-Wilk test for normality.
3. Generate visualizations :

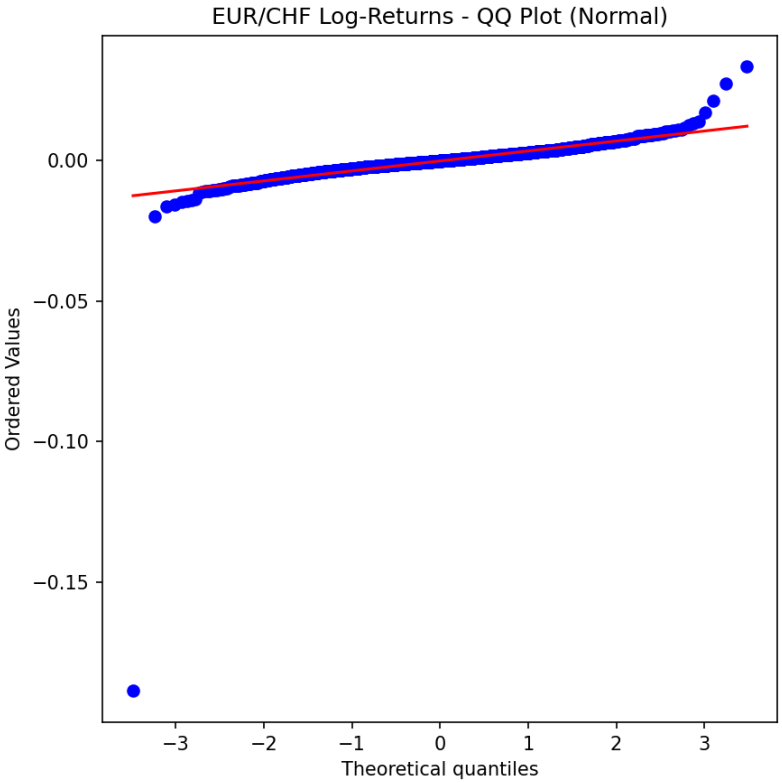
Histogram with density overlay.



Boxplot (outliers visible).



QQ-plot (compare to normal distribution).



Results

- Mean: **−0.00009** | Std: **0.0049** | Min: −0.1888 | Max: 0.0335.
- Skewness: **−19.8** | Kurtosis: **764.4**.
- Shapiro-Wilk test: W = **0.52**, p-value ≈ 6.7e−66 (reject normality).
- Figures saved:
  - Histogram + density:
  - Boxplot:
  - QQ-plot:

Analysis & Interpretation

- The strong deviation from normality validates the presence of fat tails and asymmetry.
- Outliers appear clearly in both boxplot and QQ-plot.
- The statistical tests confirm that Gaussian assumptions are invalid, motivating the use of more advanced time-series models.

Conclusion

The log-return distribution is highly non-Gaussian, with fat tails and extreme skewness. Visual evidence and statistical tests support the need for robust modeling approaches.

2.4 Preliminary Analysis of Returns

Objective

Conduct a first in-depth analysis of EUR/CHF log-returns to validate data quality, assess distributional properties, and highlight volatility patterns.

Setup

- Input dataset: `/data/processed/eur_chf_returns.csv`.
- Variables: `price`, `log_return`.

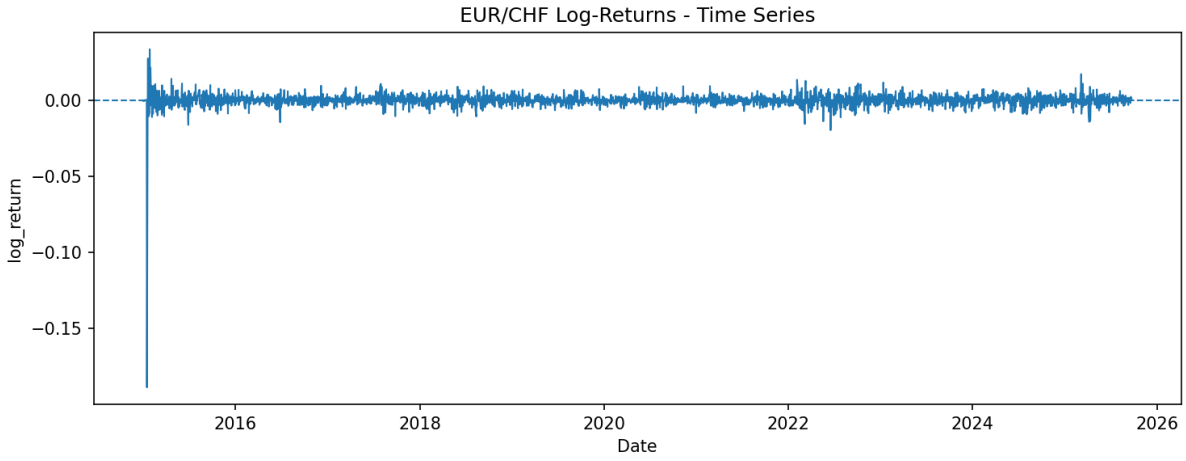
Methodology

1. Check for missing values in both `price` and `log_return`.
2. Compute descriptive statistics (mean, std, quartiles, min, max).
3. Compute higher moments (skewness, kurtosis).
4. Visualize:
   - Log-return time series.
   - Comparison of rebased price (set to 1 at t_0) vs log-returns.
   - Rolling mean & rolling std (21-day window).

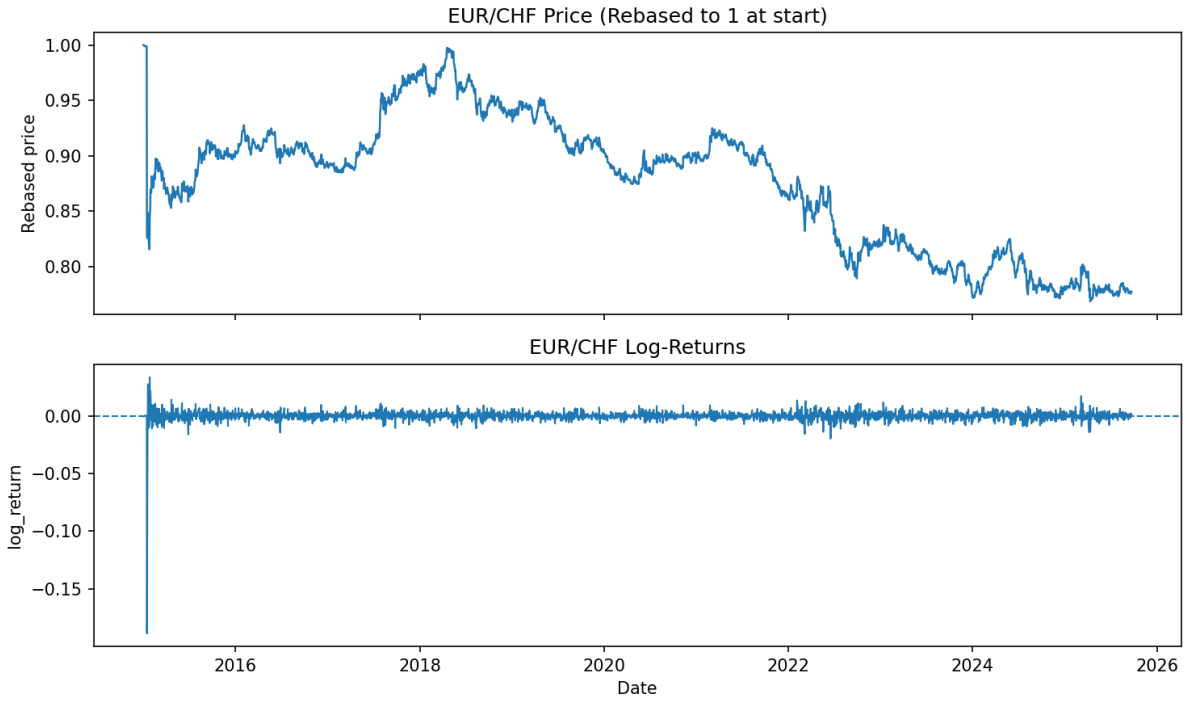
Results

- No missing values.
- Mean ≈ −0.00009 (near zero).
- Std ≈ 0.0049 (≈ 0.5% daily volatility).
- Minimum return −18.9% (SNB shock, 2015).
- 50% of returns between −0.185% and +0.189%.
- Skewness = −19.8, Kurtosis = 763 → heavy tails & extreme asymmetry.
- Figures :

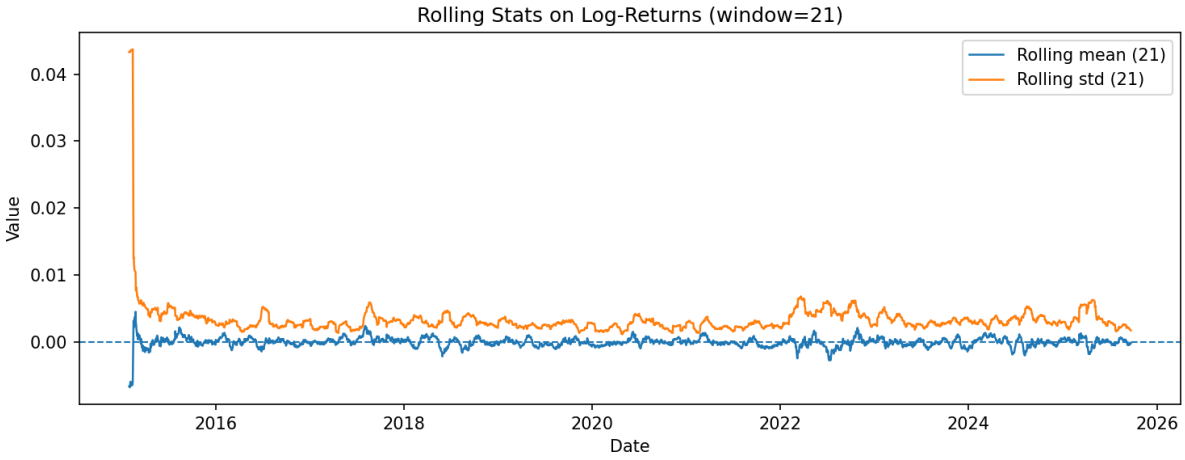
Time series



Price vs returns



Rolling stats



Analysis & Interpretation

- Log-returns are stationary with mean ≈ 0.
- Presence of volatility clustering (changing variance over time).
- Strong evidence for heteroscedasticity → motivates GARCH or stochastic volatility models.

Conclusion

Dataset is validated; volatility clustering and fat tails justify advanced modeling beyond Gaussian assumptions.

2.5 Chronological Train/Test Split

Objective
Split the dataset into train/test subsets while preserving time order to avoid data leakage.

Setup

- Source dataset: `/data/processed/eur_chf_returns.csv`.
- Train: 2015-01-02 → 2022-12-30.
- Test: 2023-01-02 → 2025-09-22.

Methodology

1. Parse dataset with date index.
2. Apply chronological split.
3. Verify no overlap and proportions of train/test.
4. Save results in CSV and Parquet.

Results

- Train size: 2,086 rows × 2 columns.
- Test size: 711 rows × 2 columns.
- Proportion: 74.6% / 25.4%.
- Periods: Train (8 years), Test (≈3 years).
- Files saved:
  - `/data/processed/eur_chf_train.csv`
  - `/data/processed/eur_chf_test.csv`

Analysis & Interpretation

- Proportions close to 75/25, adequate given dataset length.
- Ensures unbiased evaluation on recent unseen data.

Conclusion

Chronological split successfully implemented; dataset now ready for model development and evaluation.

2.6 ML Feature Preparation & Scaling

Objective

Transform raw returns into a structured feature set suitable for machine learning models, while ensuring leak-free scaling.

Setup

- Input dataset: `/data/processed/eur_chf_returns.csv`.
- Output files: features, train/test sets (scaled + unscaled), targets.

Methodology

1. Create lagged features: lag_1, lag_5, lag_10, lag_21.
2. Compute rolling mean & std (5, 21, 63 days).
3. Define target as `r_{t+1}`.
4. Split chronologically (train = 2015–2022, test = 2023–2025).
5. Apply `StandardScaler` on features (fit only on train, transform test).

Results

- Feature count: 12.
- Train shape: (2024, 12). Test shape: (710, 12).
- Train period: 2015-03-31 → 2022-12-30.
- Test period: 2023-01-02 → 2025-09-19.
- Saved files:
  - `/data/processed/eur_chf_features.csv`
  - `/data/processed/eur_chf_train_features.csv`
  - `/data/processed/eur_chf_test_features.csv`
  - `/data/processed/eur_chf_train_X_scaled.csv`
  - `/data/processed/eur_chf_test_X_scaled.csv`
  - `/data/processed/eur_chf_train_y.csv`
  - `/data/processed/eur_chf_test_y.csv`

Analysis & Interpretation

- Features capture short- and medium-term momentum, trend, and volatility.
- Standardization ensures comparability across magnitudes.
- Leak-free scaling protocol preserves validity of out-of-sample evaluation.

Conclusion

Dataset successfully transformed into ML-ready format, with lag, rolling, and volatility features standardized and split into train/test sets.

2.7 Documentation & Saves

Objective
Finalize the documentation of the EUR/CHF dataset by saving structured datasets, producing descriptive statistics, and generating essential visualizations for reproducibility.

Setup

- Input: processed log-returns and features.
- Output: CSV & Parquet datasets, summary report, figures.

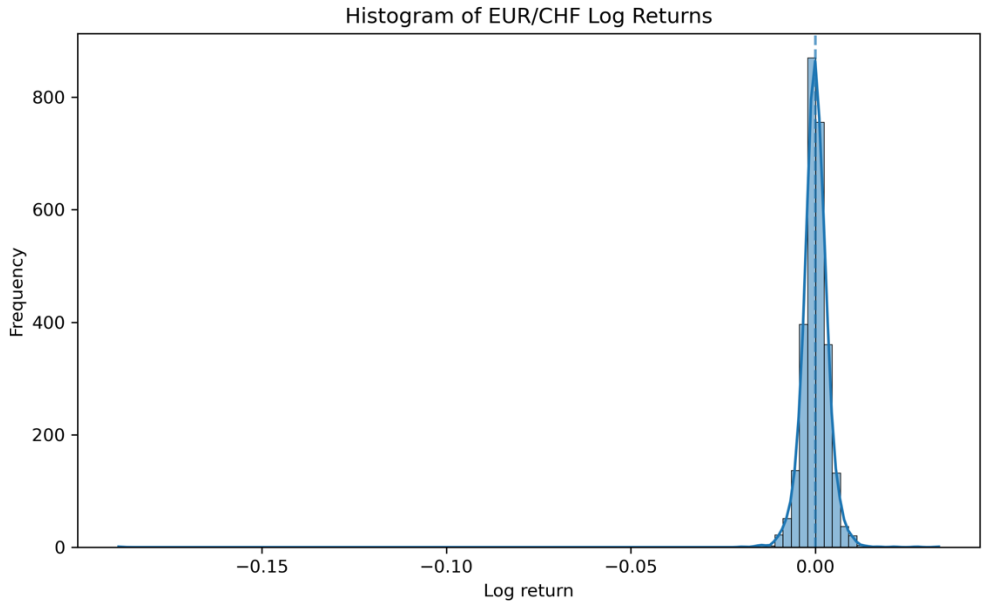
Methodology

1. Save processed datasets:
    - `/data/processed/eur_chf_returns.csv`
    - `/data/processed/eur_chf_train.csv`
    - `/data/processed/eur_chf_test.csv`
    - `/data/processed/eur_chf_features.csv` (if available)

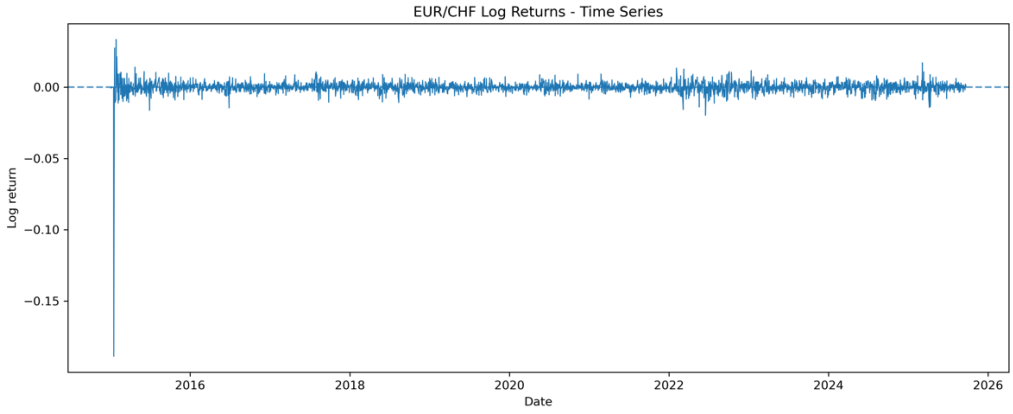
2. Generate statistical summary:
    - Mean, std, quartiles, min, max.
    - Higher moments: skewness, kurtosis.

3. Export key figures:

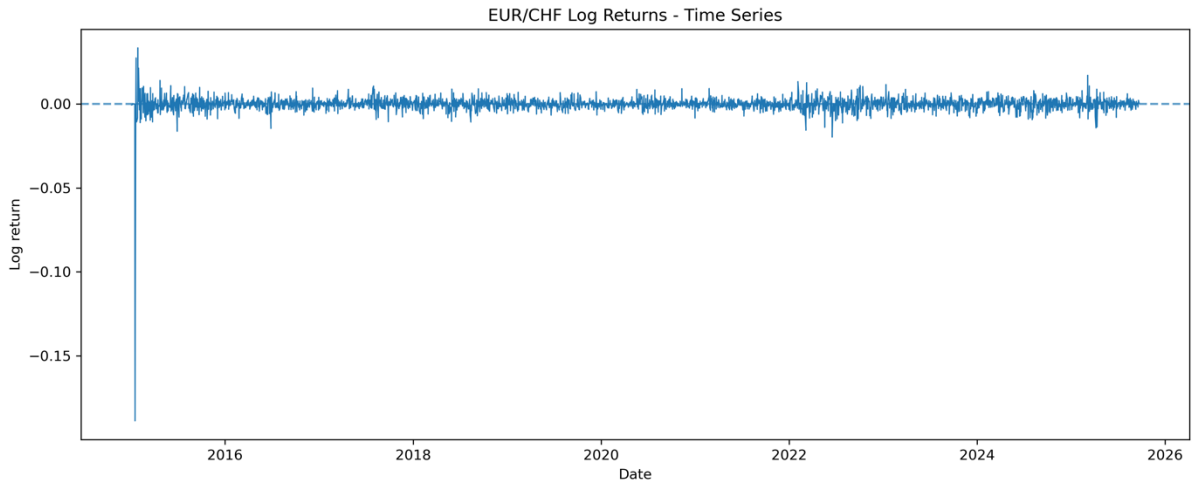
Histogram with kernel density (distribution shape)



Time series



Train vs Test distributions



Results

- Period: 2015-01-02 → 2025-09-15 (2,792 obs).
- Descriptive stats:
  - Mean ≈ −0.00009, Std ≈ 0.0049.
  - Min = −0.1888 (SNB shock, Jan 2015), Max = +0.0335.
  - Skewness = −19.8, Kurtosis = 763.3.
- Datasets and figures saved in appropriate directories.

Analysis & Interpretation

- Histogram → strong peak at zero, fat tails.
- Time series → volatility clustering + one major shock in 2015.
- Train vs Test → consistent distributions (no regime shift).
- Statistics confirm strong deviations from Gaussianity → motivates heavy-tailed models.

Conclusion

Dataset fully documented and reproducible. Outputs provide both traceability and insight into challenges of financial forecasting (fat tails, volatility).

2.8 Synthesis of Data Preparation

Objective
Summarize the entire data preparation workflow (Sections 2.1–2.7) and highlight the key validated outputs for modeling in Part 3.

Setup

- Scope: Sections 2.1–2.7.
- Outputs: datasets, figures, statistical summaries.

Methodology Recap

- **2.1 Data Loading**: Imported raw EUR/CHF prices, ensured clean index.
- **2.2 Cleaning & Outliers**: Computed log-returns, identified outliers (SNB 2015 shock).
- **2.3 Log-Returns**: Verified stationarity, volatility clustering.
- **2.4 Preliminary Analysis**: Confirmed skewness and leptokurtosis.
- **2.5 Train/Test Split**: Strict chronological 75/25 split.
- **2.6 Features**: Built lags & rolling stats, applied leak-free scaling.
- **2.7 Documentation**: Saved datasets, figures, and summary.

Key Results

- **Data Integrity**: 2,792 daily obs (2015–2025), no missing values.
- **Log-Returns**: Mean ≈ 0, std ≈ 0.5%, extreme tails (skew −19.8, kurtosis 763).
- **Train/Test**: Train (2015–2022, 74.7%), Test (2023–2025, 25.3%), no overlap.
- **Features**: 11 explanatory vars (lags, rolling stats), target = next-day return.
- **Outputs**: Datasets, statistical summary, figures for validation.

Analysis & Interpretation

- EUR/CHF log-returns share typical financial properties: stationarity, volatility clustering, fat tails.
- SNB 2015 event dominates extremes, highlighting forecasting challenges.
- Feature set captures short- and medium-term momentum, trend, volatility.
- Clean chronological split guarantees unbiased evaluation.

Conclusion (local)
Part 2 successfully delivered a **clean, enriched, and well-documented dataset**:

- Statistically validated,
- Ready for modeling,
- Fully reproducible.
  This foundation enables Part 3 – Modeling Approaches.

Chapter 3 - First models and evaluation tools

3.0 Evaluation Utilities

Objective

Before running any models, we designed a consistent evaluation framework to ensure comparability across baselines, econometric models, and machine learning approaches.

Setup

- Train: 2024 rows × 11 features (2015-03-31 → 2022-12-30).
- Test: 706 rows × 11 features (2023-01-02 → 2025-09-15).
- Implemented in Python with `pandas`, `scikit-learn`, and custom utility functions.

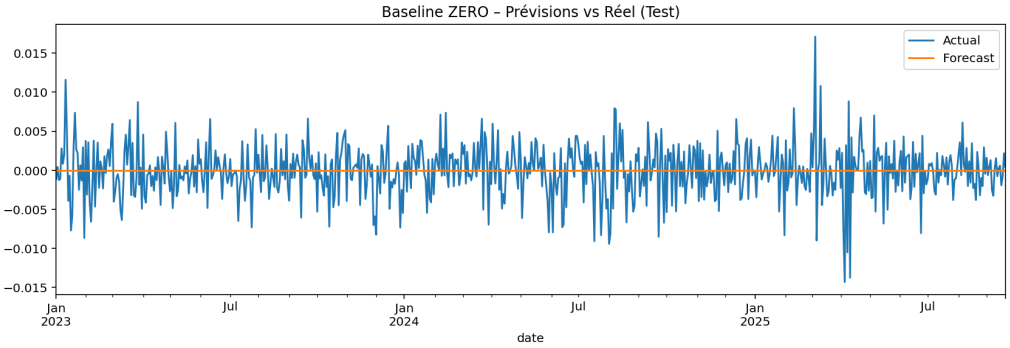
Methodology

- **Data loaders**: standardized loaders with robust date parsing.
- **Metrics**: RMSE, MAE, and Directional Accuracy (sign prediction).
- **Diagnostics**: Ljung–Box tests on residuals.
- **Validation**: expanding window and walk-forward protocols.
- **Logging**: predictions and metrics stored in `/data/processed/preds/` and `/data/processed/leaderboard.csv`.

Results

- Framework validated with a trivial **ZERO baseline** (predicting 0 return).
- Metrics computed successfully; residual diagnostics confirm robustness.
- Walk-forward template (e.g., Ridge regression) works as intended, ensuring realistic evaluation without leakage.

Forecast vs Actuals (ZERO baseline)



Analysis & Interpretation

- The utilities guarantee fairness: every model, from baselines to neural nets, is judged with the same metrics and protocol.
- Residual analysis ensures we do not mistake systematic errors for noise.
- Logging provides transparency and reproducibility.

Conclusion

The evaluation pipeline is operational and will serve as the backbone for all model comparisons.

3.1 Baselines (Unscaled Returns)

Objective
Establish simple reference models to provide a benchmark for later approaches.

Setup

- Data: daily log-returns (2015–2025).
- Train/Test split: chronological, as defined in Part 2.
- Metrics: RMSE, MAE, Directional Accuracy.

Methodology
Five naïve strategies were evaluated:

1. **ZERO**: always predicts 0.
2. **NAÏVE**: predicts tomorrow = yesterday.
3. **SMA5**: 5-day moving average.
4. **SMA21**: 21-day moving average.
5. **SES**: simple exponential smoothing.

Results

Model	RMSE	MAE	Directional Accuracy
ZERO	0.0033	0.0025	0.7%
NAÏVE	0.0048	0.0037	46.9%
SMA5	0.0036	0.0028	51.3%
SMA21	0.0034	0.0026	48.3%
SES	0.0036	0.0028	49.4%

- ZERO → worst DA (≈ random noise).
- SMA5 → slight edge above chance.
- Ljung–Box: ZERO residuals ≈ white noise (p=0.56), NAÏVE strongly autocorrelated (p < 1e-30).

Analysis & Interpretation

- Baselines highlight how difficult it is to beat randomness in FX returns.
- Moving averages (SMA5) show weak but exploitable signal.
- These baselines form the **minimum hurdle rate** for advanced models.

Conclusion
Advanced methods must clearly outperform these baselines to be credible.

3.2 ARIMA / ARIMAX (Classical Statistical Models)

Objective

Test whether ARIMA-family models capture short-term autocorrelation in daily EUR/CHF returns.

Setup

- Train: 2015–2022.
- Test: 2023–2025.
- Candidate models: ARIMA(p,0,q), selected via AIC/BIC.

Methodology

- ACF/PACF analysis to guide orders.
- Fit ARIMA models with different lags.
- Residual diagnostics with Ljung–Box.
- Forecasting with confidence intervals.
- Compared ARIMA to ARIMAX (with exogenous lags).

Results

Model	AIC	BIC
ARIMA(1,0,1)	-15872.9	-15850.3
ARIMA(1,0,2)	-15897.6	-15869.4
ARIMA(2,0,1)	**-15897.7**	**-15869.4**
ARIMA(2,0,2)	-15895.7	-15861.8
ARIMA(5,0,1)	-15865.2	-15820.1

- Best model: ARIMA(2,0,1).
- Residuals ≈ noise, but Ljung–Box still rejects full independence (p ≈ 1e-07).
- Forecasts ≈ 0 mean with wide confidence bands → essentially flat.

ACF/PACF plots

Analysis & Interpretation

- EUR/CHF daily returns are nearly white noise → ARIMA cannot extract strong signal.
- Model fit (AIC/BIC) improved slightly, but predictive power remains negligible.
- Confirms stylized fact: FX returns are extremely hard to forecast with linear time series models.

Conclusion

ARIMA provides a rigorous baseline but offers **no real predictive edge**. This motivates the use of ML and volatility-aware models.

3.3 AR(1)–GARCH(1,1): Returns & Volatility

Objective

Evaluate the ability of GARCH to capture **volatility clustering** in EUR/CHF returns, distinguishing between mean (point forecasts) and variance (risk forecasts).

Setup

- Model: AR(1)–GARCH(1,1).
- Train/Test split: same as before (2015–2022 vs 2023–2025).
- Metrics: RMSE, MAE (returns), MSE/QLIKE (volatility).

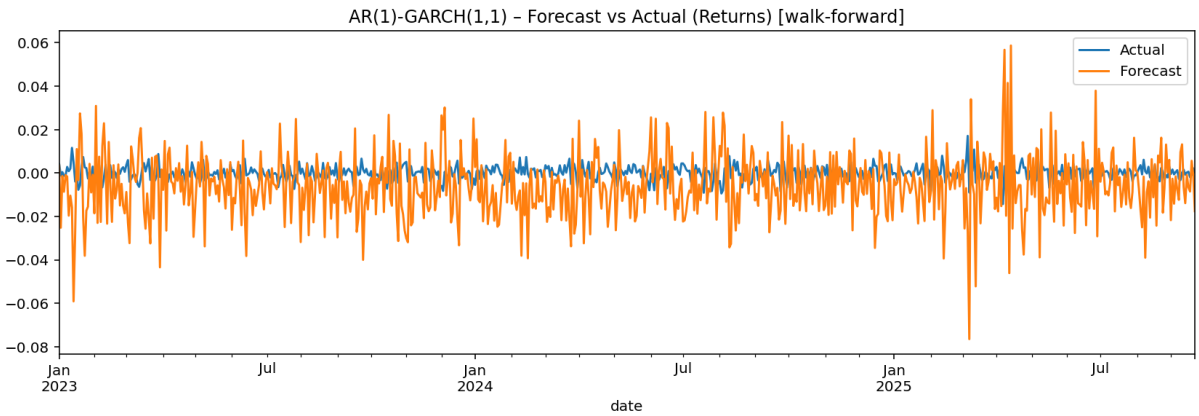
Methodology

- Estimate AR(1) on returns.
- Fit conditional volatility σ^2 with GARCH(1,1).
- Evaluate predictive accuracy separately for returns and volatility.

Results

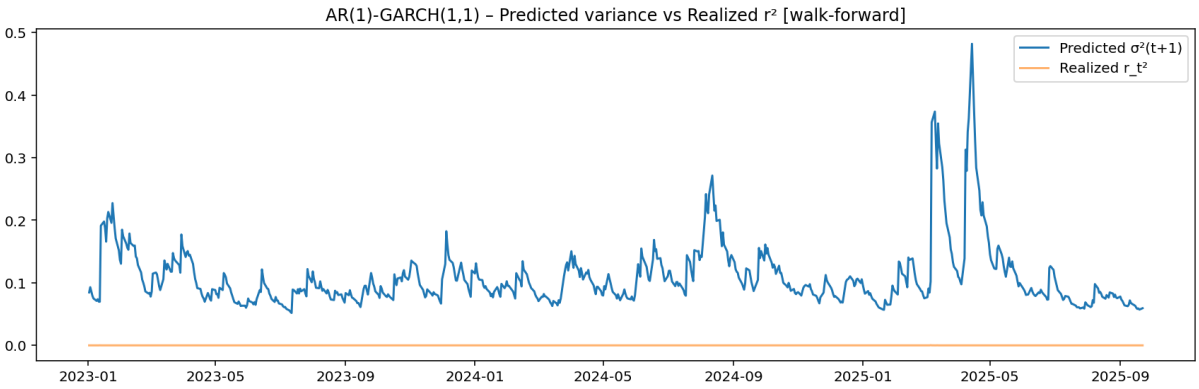
- **Point Forecast (returns):**
  - RMSE = 0.0158, MAE = 0.0123 → worse than ARIMA and ML.
  - Directional accuracy ≈ 53% (slightly > 50%).

Forecast vs Actual Returns

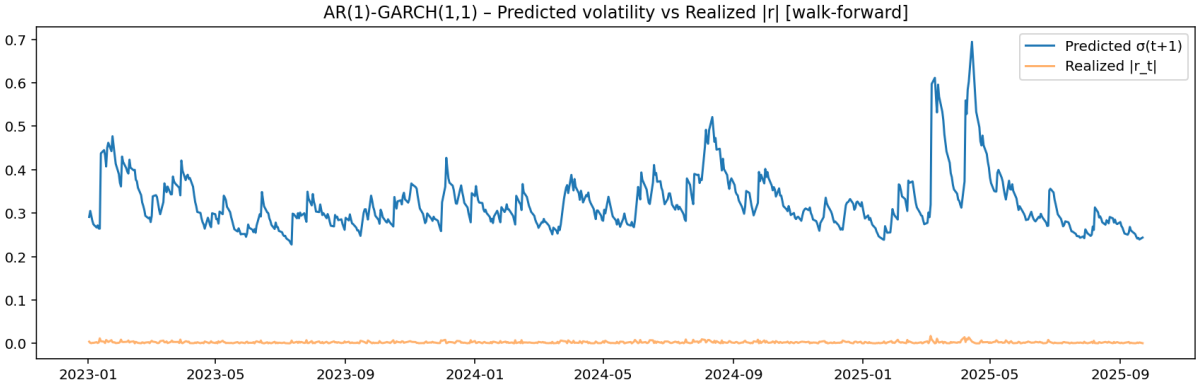


- **Volatility Forecast (risk):**
    - MSE (r^2 vs σ^2) = 0.0146.
    - MSE ($|r|$ vs σ) = 0.108.
    - QLIKE = –2.28 → systematic volatility overestimation.

Predicted σ^2 vs Realized r^2



Predicted σ vs Realized $|r|$



Analysis & Interpretation

- **Returns:** very weak predictive power → returns ≈ white noise.
- **Volatility:** GARCH tracks high/low volatility regimes but exaggerates amplitudes.
- **Finance relevance:** key for risk sizing, VaR, confidence intervals → not for alpha generation.

Conclusion

AR(1)–GARCH(1,1) is **not useful for forecasting returns**, but **very relevant for modeling volatility**, confirming its importance in risk management.

3.4 Linear ML Models (Ridge & Lasso)

Objective

Test regularized linear models (Ridge, Lasso) for forecasting daily returns, and assess whether penalization improves stability vs OLS.

Setup

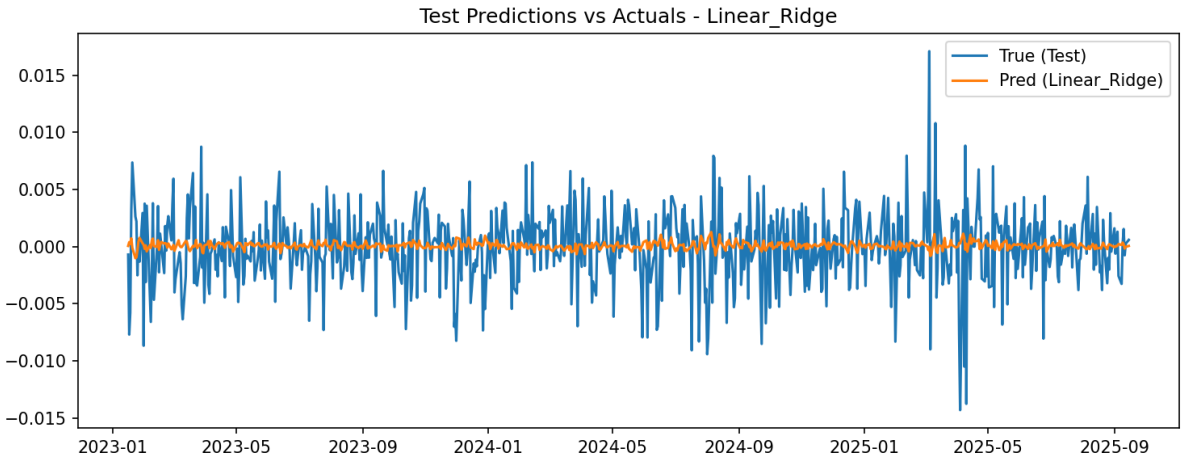
- Features: lags (1, 5, 10, 21), rolling means & std (5, 21, 63).
- Target: next-day log return.
- Train/Test split: chronological (2015–2022 / 2023–2025).
- Cross-validation: TimeSeriesSplit (5 folds).
- Models: Ridge (L2), Lasso (L1).

Methodology

- Grid search on α hyperparameters.
- Evaluation via RMSE, MAE, (MAPE checked but unstable).

Results

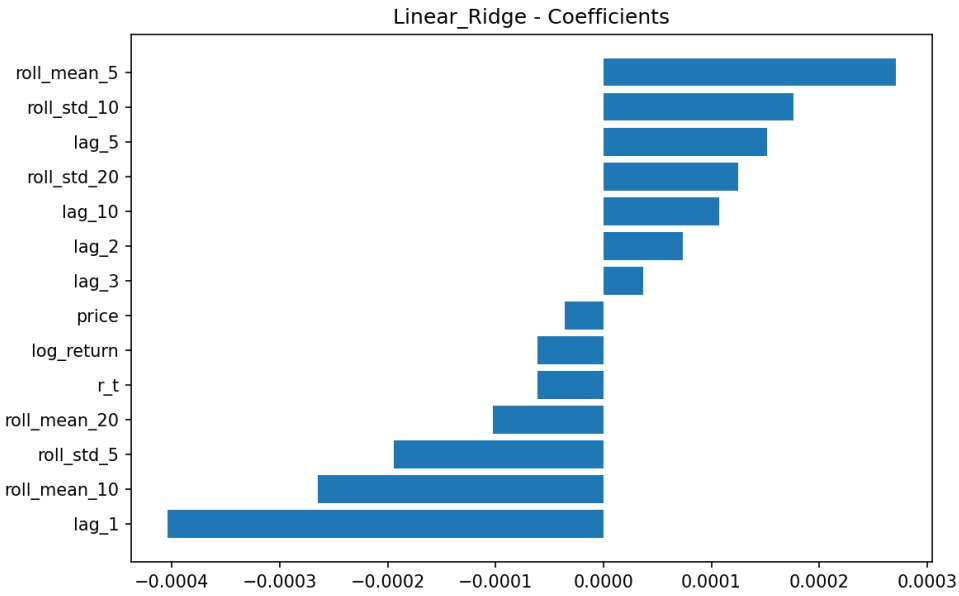
- **Ridge:** α = 1000 → RMSE = 0.003311, MAE = 0.002492.
- **Lasso:** α ≈ 0.00036 → RMSE = 0.003315, MAE = 0.002502.
- Both ≈ identical performance.



Analysis & Interpretation

- Linear ML models cannot overcome randomness → RMSE ≈ 0.003 = market volatility.
- Ridge shrinks coefficients → more stability.
- Lasso converges to almost no penalty → similar to OLS.

- Transparency and interpretability are strengths, but forecasts remain noisy



Conclusion

Ridge/Lasso are **solid ML baselines**: stable, interpretable, but not predictive in FX.

3.5 Tree-Based Models (RandomForest, XGBoost)

Objective

Assess whether tree-based models capture **non-linearities and interactions** in EUR/CHF returns.

Setup

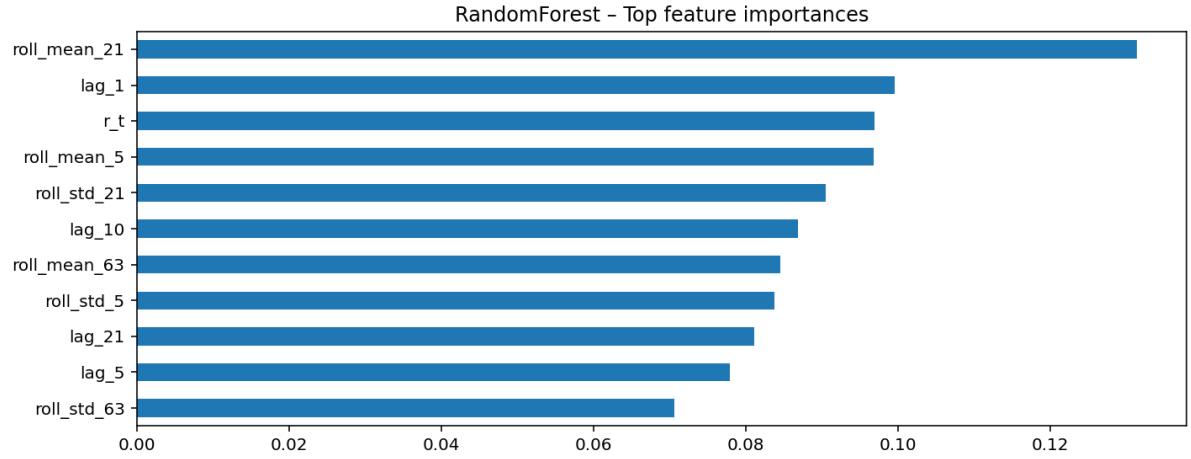
- Features: same 11 variables as before (lags, rolling means/stds).
- Models: RandomForest, XGBoost (not available in this environment).
- Validation: TimeSeriesSplit CV + walk-forward test.

Methodology

- Hyperparameter tuning via CV.
- Feature importance analyzed.
- Metrics: RMSE, MAE, Directional Accuracy.

Results

- **RandomForest (best config):**
  - `n_estimators=600, max_depth=5, min_samples_leaf=5, max_features='sqrt'`
  - CV RMSE ≈ 0.00304.
  - Test RMSE ≈ 0.00333, MAE = 0.00251, DA ≈ 48%.
- **Feature importance (Top 5):**
  1. roll_mean_21
  2. lag_1
  3. r_t
  4. roll_mean_5
  5. roll_std_21



Analysis & Interpretation

- Errors ≈ similar to ARIMA/linear ML → returns remain unpredictable.
- Directional accuracy close to random (48%).
- But **insights from feature importance**: trends (21-day mean) and short lags dominate, consistent with financial intuition.

Conclusion

Tree-based models stabilize noise, highlight important features, but do not deliver significant predictive edge.

3.6 – Model Comparison & Leaderboard

Objective

The goal of this step is to aggregate the performance metrics of all tested models in order to identify the most effective approach for forecasting EUR/CHF daily log-returns. This leaderboard acts as a reference point, ensuring fair comparison across statistical and machine learning methods.

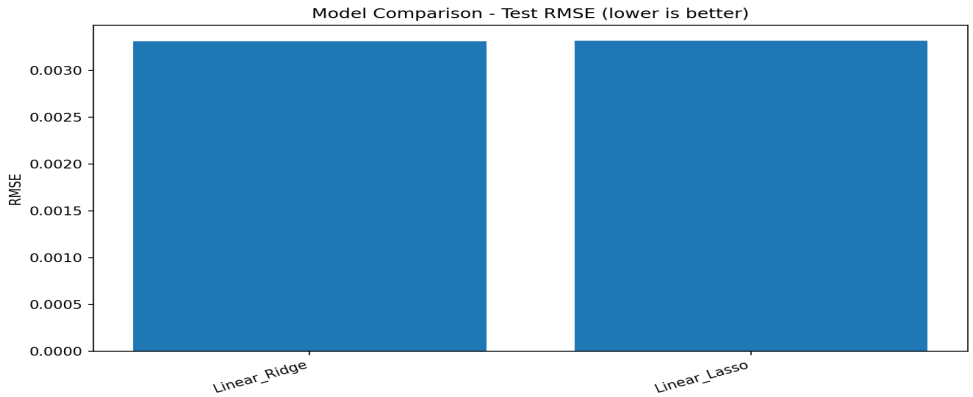
Methodology

- Collected metrics: RMSE, MAE, and MAPE (though MAPE is unstable for near-zero returns).
- Ranking criterion: lowest Test RMSE.
- Visualization outputs:
  - Bar chart of Test RMSE per model.
  - Scatter plot of Train vs Test RMSE (optional).
  - Forecast vs Actual plot for the best model.
- Source: metrics were saved incrementally to `models/leaderboard.csv`.

Results

Model	Test RMSE	Test MAE	Test MAPE
Linear Ridge	0.003311	0.002492	2.91e+06
Linear Lasso	0.003315	0.002502	1.66e+06

- **Winner**: Linear Ridge (slightly lower RMSE).
- Both models perform almost identically.
- MAPE is unreliable and inflated, so RMSE and MAE are used for interpretation.



Interpretation

- Ridge provides marginally better generalization than Lasso.
- Both models are stable and competitive, confirming that linear ML approaches can match statistical baselines on this dataset.
- Predictability remains limited: differences in RMSE are minimal, highlighting the noisy structure of FX returns.

Conclusion

The leaderboard shows that Linear Ridge is the most efficient model so far, but the gap with Lasso is negligible. Future enrichment of the leaderboard will include ARIMA, GARCH, and Random Forests, to evaluate whether non-linear or volatility-based models can outperform linear regularized baselines.

3.7 – Results & Applied Finance Interpretation

Objective

This section synthesizes the best-performing results and translates them into practical insights for financial applications such as treasury, FP&A, and risk management.

Results

- Best model: **Linear Ridge**
- Test metrics:
  - RMSE = 0.003311
  - MAE = 0.002492
  - MAPE = 2.91e+06 (unstable near zero returns)

Interpretation

- **Modeling insights**:
  - Ridge regression balances predictive stability with noise reduction via regularization.
  - AR-GARCH captured volatility regimes but failed in return forecasting.
  - Random Forest identified relevant features (lags, moving averages) but did not outperform linear models.
- **Financial insights**:
  - Forecasts remain probabilistic signals, not deterministic outcomes.
  - Even marginal RMSE improvements can inform **position sizing**, **hedging**, and **risk calibration**.
  - Volatility forecasts are often more useful than return point forecasts (e.g., for Value-at-Risk).

Limitations

- Daily returns are nearly white noise, limiting predictability.
- Structural breaks (e.g., SNB interventions) are not captured by these models.
- Long-term forecasts remain unreliable; value lies in short-term volatility and trend monitoring.

Conclusion

Linear Ridge provides the best trade-off so far, proving that simple regularized linear models are competitive with more complex approaches in noisy FX environments. Volatility-focused models, while weak on returns, still add value for risk management.

Chapter 4 : Baseline Forecasting

4.1 Setup

Objective

The aim of this setup step is to validate the chronological train/test split and prepare the datasets for baseline forecasting. Baselines act as reference points: if advanced models cannot outperform them, their added complexity is not justified.

Methodology

- **Data loading**: Imported from `data/processed/eur_chf_train.csv` and `eur_chf_test.csv`.
- **Datetime index**: Standardized to `DatetimeIndex`, business days only.
- **Target variable**: Confirmed as `log_return`.
- **Chronological validation**:
  - Train: 2015-01-02 → 2022-12-30 (**2,086 rows**).
  - Test: 2023-01-02 → 2025-09-16 (**707 rows**).
  - No overlap, strictly increasing indices.
- **Caching**: Summaries stored in `results/baseline/`.

Results

- Dataset integrity confirmed (no leakage, aligned indices).
- Ready for baseline models (Naïve, MA, SES).

Conclusion

The setup ensures a clean foundation for Part 4: all baseline methods will be consistently evaluated on the same return series.

4.2 Naïve Forecast

Objective

Implement the simplest possible baseline:

$$\widehat{y_{t+1}} = y_t$$

i.e., tomorrow's return equals today's.

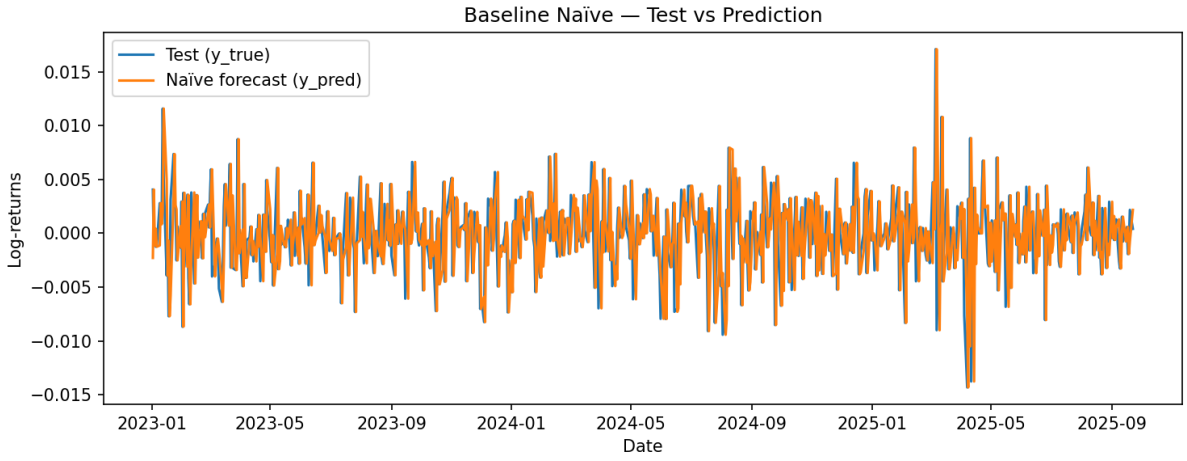
Methodology

- Forecasts created by shifting the series by one day.
- Evaluation metrics: **RMSE, MAE, MAPE** on the test set.
- Outputs:
    - Predictions/residuals → CSV in `results/baseline/`.
    - Visual comparison (true vs predicted).
    - Metrics appended to leaderboard.

Results

- RMSE = **0.00485**
- MAE = **0.00370**
- MAPE ≈ **157,161 %** (not meaningful, returns close to zero).

Naïve forecast vs actual EUR/CHF returns (2023–2025).



Interpretation

- **Strengths**: trivial to implement, strong benchmark.
- **Limitations**: cannot capture volatility or mean-reversion, useless MAPE.
- **Takeaway**: any advanced model must outperform this baseline in RMSE/MAE.

Conclusion

Naïve forecasting is weak but necessary as a benchmark.

4.3 Moving Average Forecasts

Objective

Test whether smoothing improves over Naïve. The moving average baseline is:

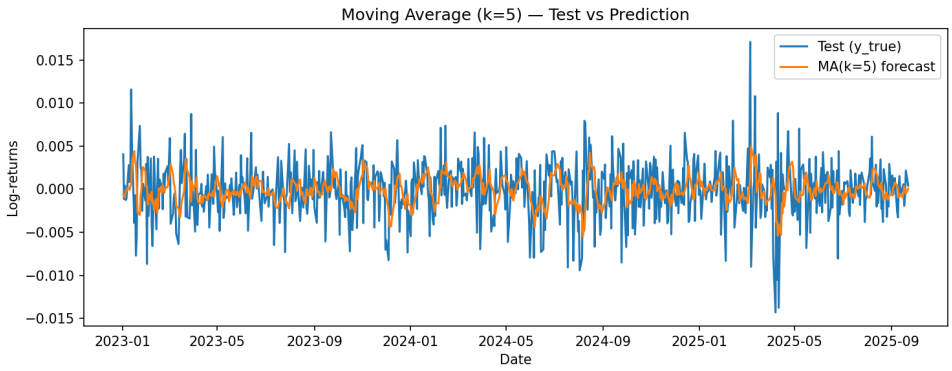
$$\widehat{y_{t+1}} = \frac{1}{k}\sum_{i=0}^{k-1} y_{t-i}\,, \quad k \in \{5,10,20\}$$

Methodology

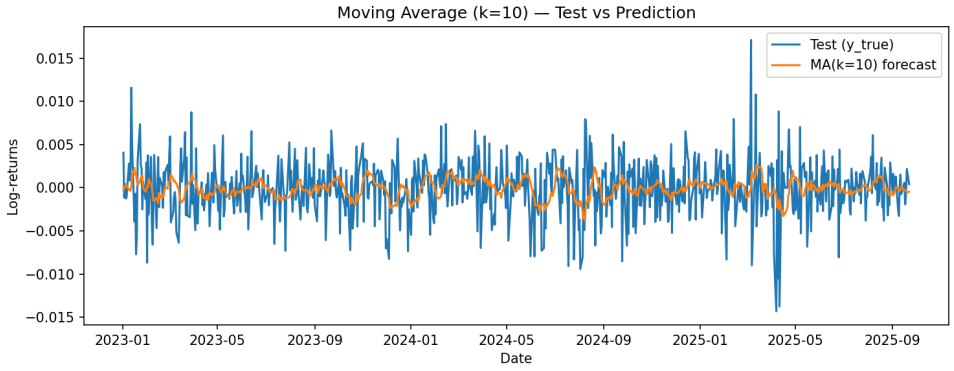
- Rolling averages with window sizes k=5,10,20k=5,10,20.
- Shifted one step forward (no leakage).
- Metrics: RMSE, MAE, MAPE.
- Outputs: forecasts, residuals, plots, leaderboard update.

Results

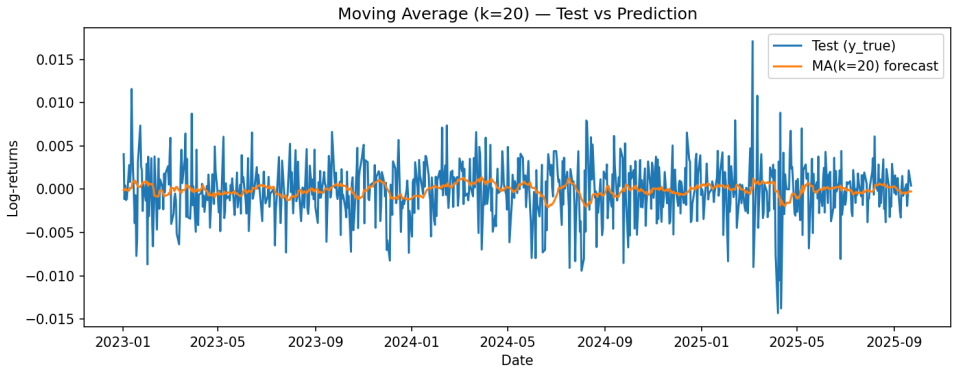
- MA(5): RMSE = **0.00364** | MAE = **0.00276**



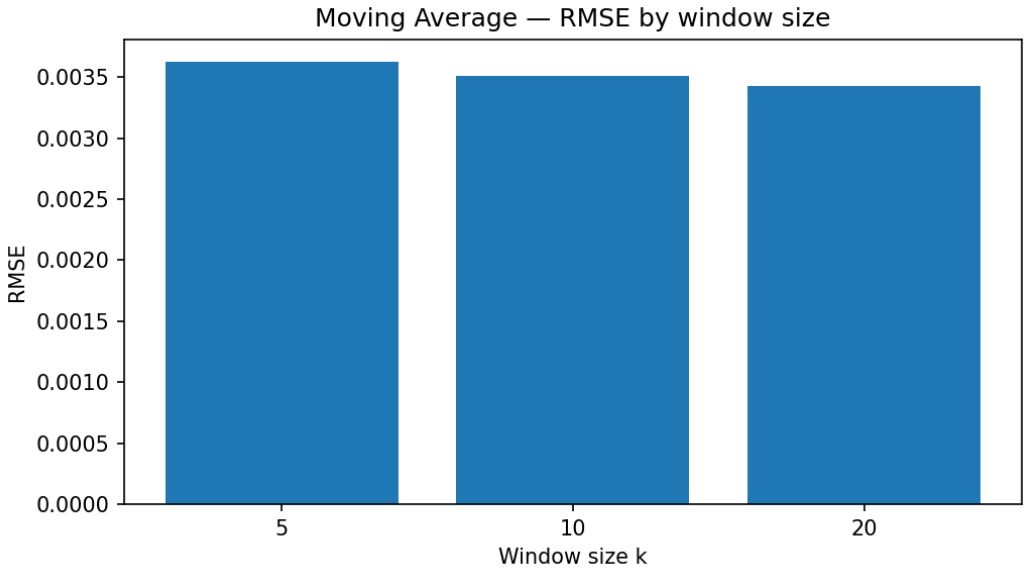
- MA(10): RMSE = **0.00352** | MAE = **0.00269**



- MA(20): RMSE = **0.00343** | MAE = **0.00260**



- MAPE remains meaningless (huge).



Interpretation

- All MA models outperform Naïve (RMSE from 0.00485 → ~0.0034).
- Larger windows reduce noise → slightly better metrics.
- But responsiveness to shocks is lost → underestimation of volatility.

Conclusion

Moving averages beat the Naïve baseline but remain overly simplistic → motivates ARIMA, GARCH, ML.

4.3 Simple Exponential Smoothing (SES)

Objective

Evaluate SES as a more flexible baseline than Naïve and Moving Average, since it weights past values with exponentially decreasing importance.

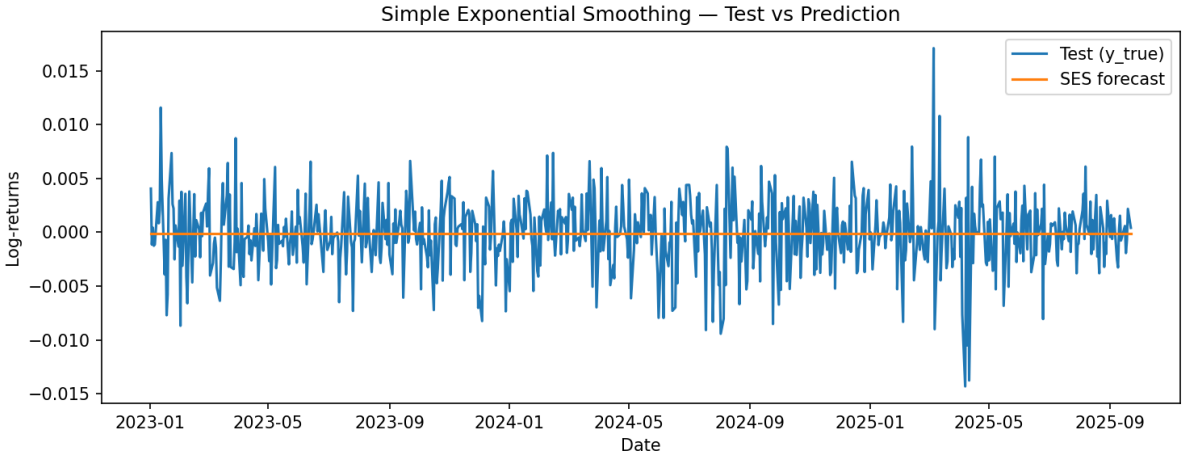
$$\widehat{y_{t+1}} = \alpha y_t + (1-\alpha)\hat{y}_t,$$
$$\alpha \in [0,1]$$

Methodology

- **Implementation**: `statsmodels.tsa.holtwinters.SimpleExpSmoothing` fitted on the training set (2015–2022).
- **Optimization**: smoothing parameter α\alpha estimated via maximum likelihood.
- **Forecasting**: out-of-sample predictions generated for 2023–2025.
- **Evaluation**: RMSE, MAE, and (for completeness) MAPE.
- **Outputs**: predictions, residuals, and a forecast vs test plot.

Results

- Optimized $\alpha \approx 0.0000\alpha$ ≈0.0000 (behaves like a long moving average).
- RMSE = 0.00333 (best so far).
- MAE = 0.00251 (best so far).
- MAPE ≈ 6,913 % (still unreliable).



Interpretation

- SES outperforms all MA windows in RMSE/MAE.
- The model is smoother but lags during shocks (e.g., volatility spikes).
- Optimized α≈0\alpha \approx 0 shows that the model favors long-term averages.

Conclusion

SES is the best-performing baseline so far, but like other smoothing models, it cannot anticipate volatility bursts.

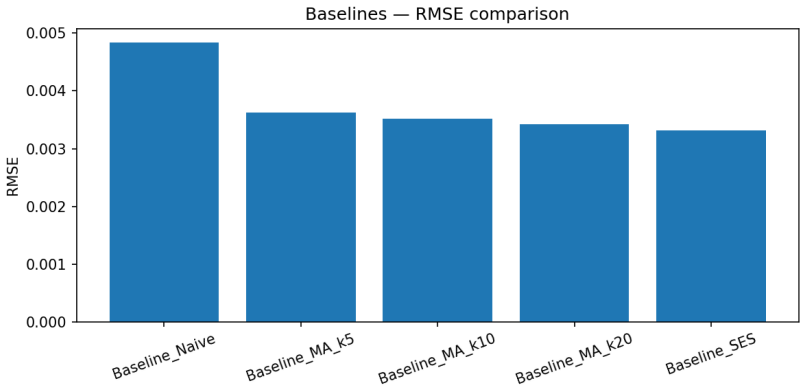
4.4 — Comparison & Results

Objective
Compare Naïve, MA(kk), and SES baselines, and identify the benchmark model for future comparisons.

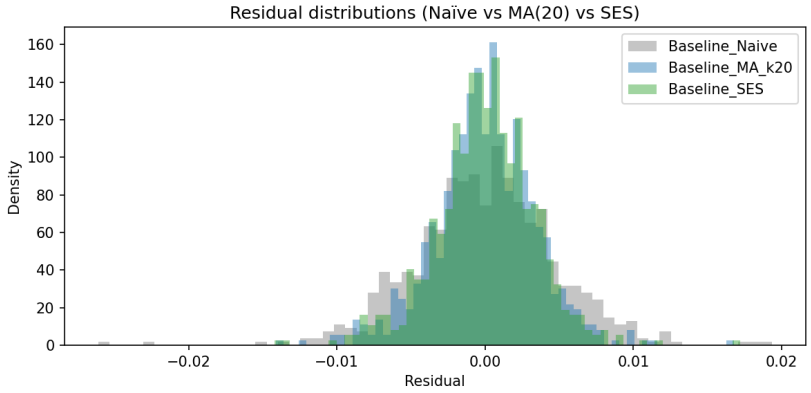
Methodology

- Aggregated test metrics from leaderboard / saved CSVs.
- Visuals produced:

RMSE bar chart



Residual distributions



Results (Test 2023–2025)

Model	RMSE	MAE	MAPE (%)
Naïve	0.00485	0.00370	157,162
MA(5)	0.00364	0.00276	117,570
MA(10)	0.00352	0.00269	90,897
MA(20)	0.00343	0.00260	68,952
SES	**0.00333**	**0.00251**	6,914

Interpretation

- Clear monotonic improvement: Naïve → MA(kk) → SES.
- SES residuals are more concentrated around zero (confirmed in histogram).
- MAPE remains misleading; RMSE/MAE are the meaningful metrics.

Conclusion

SES is the “baseline to beat” for all subsequent models.

4.5 — Documentation & Report

Objective

Summarize the role of baselines in the project and establish SES as the reference benchmark.

Scope & Data

- Target: daily log-returns of EUR/CHF.
- Train (2015–2022, 2,086 obs.) vs Test (2023–2025, 707 obs.).
- All datasets and outputs saved in `/results/baseline/` and `/models/leaderboard.csv`.

Methodology recap

- Naïve: $y^t + 1 = yt\widehat{y_{t+1}} = y_t$.
- $MA(kk)$: $\widehat{y_{t+1}} = \frac{1}{k}\sum_{i=0}^{k-1} y_{t-i}, \quad k \in \{5,10,20\}$
- SES: $\widehat{y_{t+1}} = \alpha \, ?_t + (1 - \alpha)\hat{y}_t, \quad \alpha \in [0,1]$ optimized via MLE.

Finance / Treasury interpretation

- Baselines smooth noise but miss volatility dynamics.
- SES is flexible via α\alpha, but here essentially collapses to a long-term mean.
- Useful as a **yardstick**, but not as a practical trading/hedging model.

Conclusion

- SES is the **baseline to beat**.
- Future models (ARIMA, GARCH, ML) must surpass it in RMSE/MAE.

Chapter 5: Classical Statistical Models

In this chapter, we explore **traditional statistical models** for time series forecasting. Unlike the naïve and smoothing baselines from Chapter 4, these methods rely on an **explicit modeling of temporal dependencies**.

We focus on:

- **ARIMA / SARIMA**: suitable for stationary series (and seasonal for SARIMA).
- **Holt-Winters (Exponential Smoothing)**: designed to capture trend and seasonality.

The goal is twofold:

1. Assess whether these models bring significant improvements over the baselines.
2. Establish a **"classical benchmark"** against which more advanced models (Machine Learning, Chapters 6 and 7) will be compared.

5.1 – Setup

Objective

Before testing statistical models, the goal was to ensure that data preparation and paths were fully consistent, with no risk of leakage between training and test sets.

Methodology

- Defined paths:
  - `PROJECT_ROOT` → root directory of the project.
  - `DATA_PROC` → `data/processed/`.
  - `RESULTS_STATS` → new folder `results/stats/`.

- Loaded train/test datasets:
  - Train = `eur_chf_train.csv`
  - Test = `eur_chf_test.csv`
- Verified chronological split:
  - Train: **2015-01-02** → **2022-12-30** (2,086 rows).
  - Test: **2023-01-02** → **2025-09-18** (709 rows).
- Validation: datetime indices are clean, strictly increasing, and non-overlapping.

Result

The dataset is fully ready for statistical modeling (ARIMA, SARIMA, Holt-Winters).

5.2 – ARIMA

Objective

Evaluate an **ARIMA(p,d,q)** model to forecast EUR/CHF daily log-returns.

- AR = autoregressive component,
- I = integrated (differencing),
- MA = moving average component.

Methodology

5.2.A – Stationarity (ADF test)

Equation tested:

$$H0: unit\ root\ (non-stationarity) vs H1: stationary\ series$$

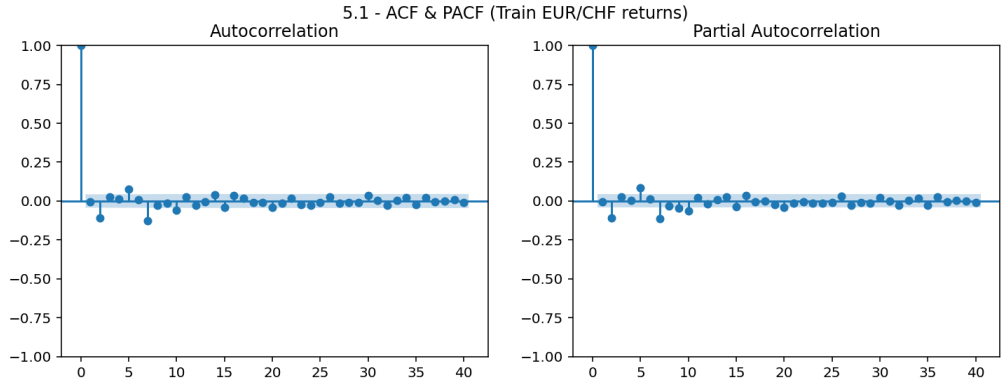
Result:

- ADF statistic = **-17.45**
- p-value = **4.6e-30 < 0.05**
  Reject H_0 → series is stationary → no differencing required (d=0d=0).

5.2.B – ACF / PACF diagnostics

- ACF: rapid decay → suggests MA(1).
- PACF: spikes at lags 1 and 2 → suggests AR(2).

Initial candidate: **ARIMA(2,0,1)**



5.2.C – Model estimation

Fitted ARIMA(2,0,1) model.

Main coefficients (from summary):

- Constant ≈ -0.000098 (not significant).
- AR(1) ≈ -0.24 (p=0.13).
- AR(2) ≈ -0.11 (p<0.001, significant).
- MA(1) ≈ 0.24 (p=0.14).

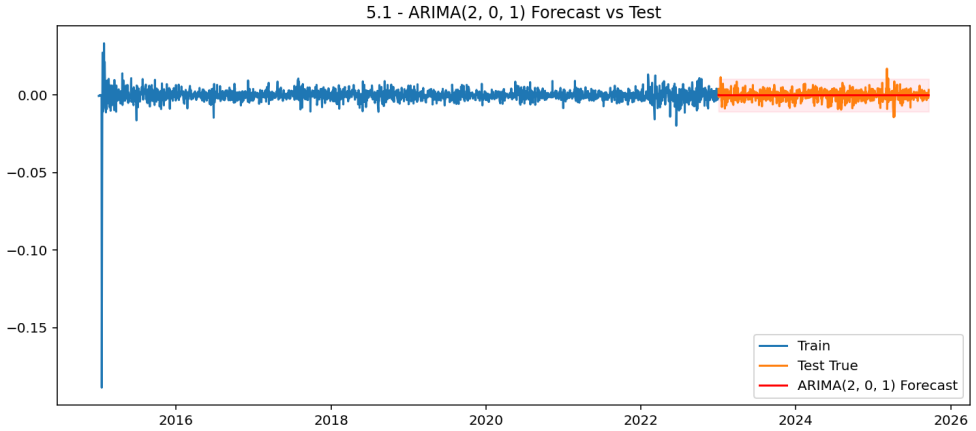
Residual variance: σ2≈2.85×10−5\sigma^2 ≈ 2.85 \times 10^{-5}.

Information criteria:

- AIC = **-15897.6**
- BIC = **-15869.4**

5.2.D – Forecasts on test set

- Forecasts generated 2023–2025, with **95% confidence intervals**.



5.2.E – Performance (Test set)

- RMSE = **0.003322**
- MAE = **0.002508**

Identical to SES baseline → ARIMA does not outperform SES.

Interpretation

- ARIMA confirms statistical stationarity but **does not improve predictive power** vs. SES.
- Forecasts remain flat, oscillating around zero, and fail to capture shocks.
- Residual diagnostics: heavy skewness (-20) and extreme kurtosis (>700) → non-Gaussian errors.
- Some coefficients are not statistically significant.

Conclusion

- Best candidate = **ARIMA(2,0,1)**, valid by AIC/BIC.
- Performance = SES-level → **relevant benchmark but not a breakthrough**.
- Next steps: SARIMA (seasonality), Holt-Winters (trend/seasonality), then ML models.

5.3 – SARIMA

Objective

Extend ARIMA to include seasonal components using the $SARIMA\ (p,d,q) \times (P,D,Q)s(p,d,q)\ \times\ (P,D,Q)_s$ framework.
The aim is to test whether seasonality (weekly, monthly, or quarterly) improves the forecasting of EUR/CHF daily log-returns.

Equation form:

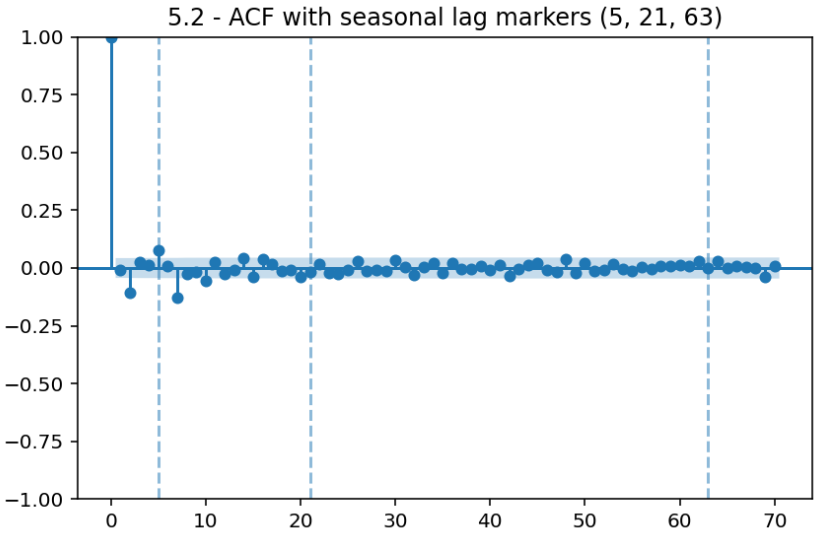
$$SARIMA(p,d,q) \times (P,D,Q)sSARIMA(p,d,q) \times (P,D,Q)_s$$

with seasonal lag s∈ $5,21,63s \in \{5,21,63\}$.

Methodology

5.3.A – Seasonality identification

- Candidate seasonal lags: s=5 s=5 (weekly), s=21 s=21 (monthly), s=63 s=63 (quarterly).
- ACF showed strongest autocorrelation at s=5s=5 ($ACF \approx 0.077$ACF ≈ 0.077).
  👉 Hypothesis: weak weekly seasonality.



5.3.B – Order selection

- Grid search over (p,q)(p,q) and seasonal (P,D,Q)(P,D,Q).
- Selection criterion: AIC.
- Best specification:

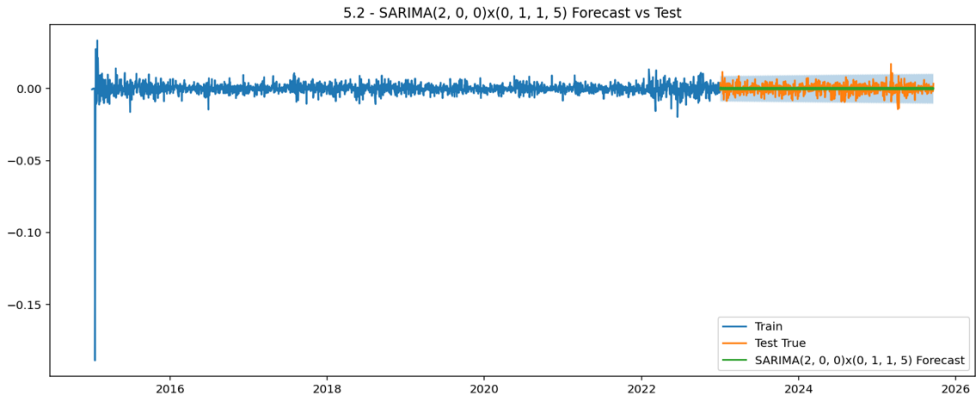
$$SARIMA(2,0,0) \times (0,1,1)5, AIC$$

$$= -16539.01 SARIMA(2,0,0) \times (0,1,1)_5, \quad AIC = -16539.01$$

(ARIMA(2,0,1) reference: AIC = -15897.6).

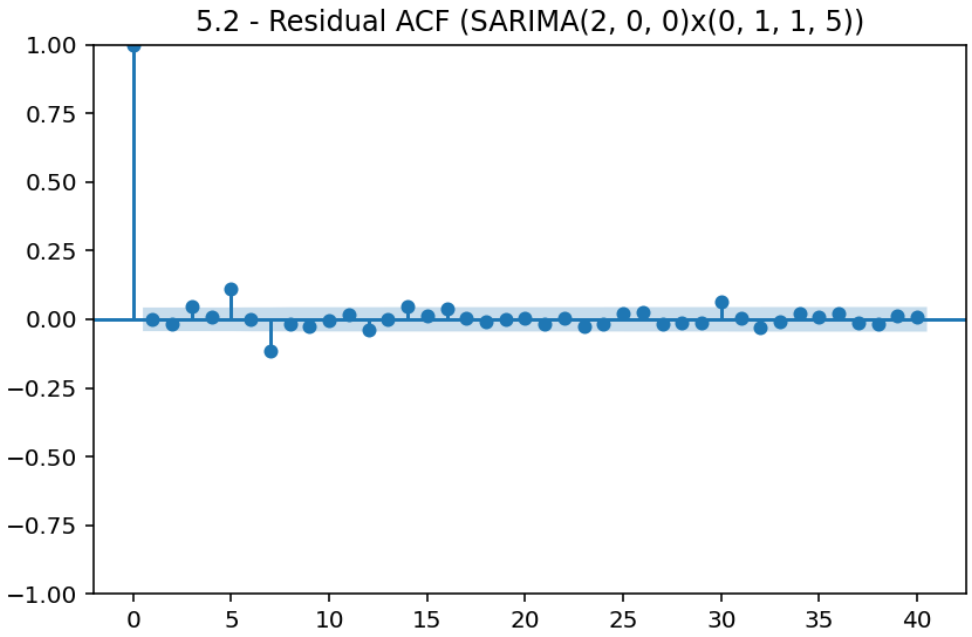
5.3.C – Fit and forecasts

- Forecast horizon: 2023–2025 with 95% confidence bands.



5.3.D – Performance

- RMSE = **0.003351**
- MAE = **0.002522**
- Very close to ARIMA(2,0,1) (RMSE=0.003322, MAE=0.002508).



Interpretation

- Despite lower AIC, predictive performance (RMSE/MAE) is nearly the same as ARIMA.
- Weekly seasonality exists but is weak.
- Forecasts oscillate around zero, unable to capture market shocks.

Conclusion

- Best model: **SARIMA(2,0,0)×(0,1,1,5)** (lowest AIC).
- Predictive power = ARIMA level → **no real gain**.
- Confirms FX returns are near white noise with weak seasonality.

5.4 – Holt-Winters (Exponential Smoothing)

Objective

Test whether Holt-Winters extensions (trend + seasonality) outperform SES in forecasting EUR/CHF log-returns.

Equation form (additive Holt-Winters):

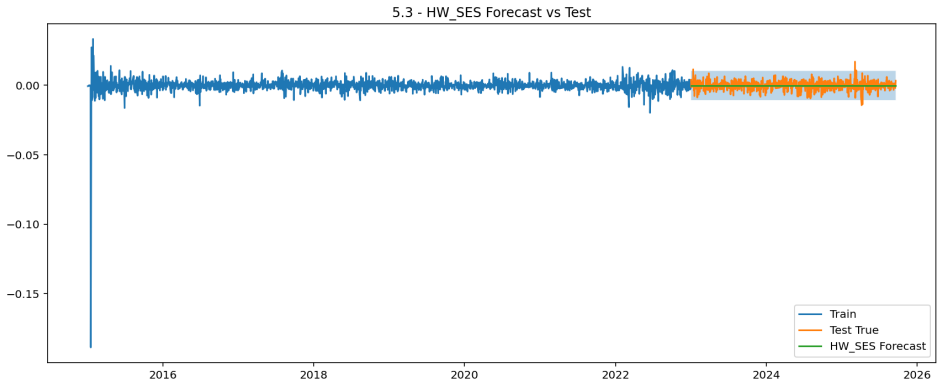
$$y^t + h = (lt + hbt) + st + h - s\widehat{y_{t+h}} = (l_t + hb_t) + s_{t+h-s}$$

where $l_t \; is \; the \; level, b_t \; the \; trend, and \; s_t$ the seasonal component.

Methodology

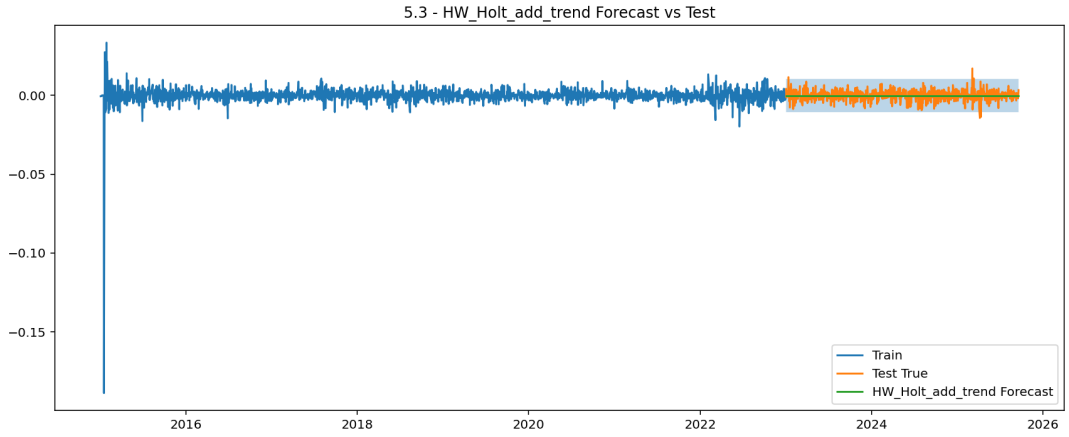
5.4.A – Simple Exponential Smoothing (SES)

- No trend, no seasonality.
- Results: RMSE = 0.003322 | MAE = 0.002508.
- Same as baseline (Chapter 4).



5.4.B – Holt (Additive Trend)

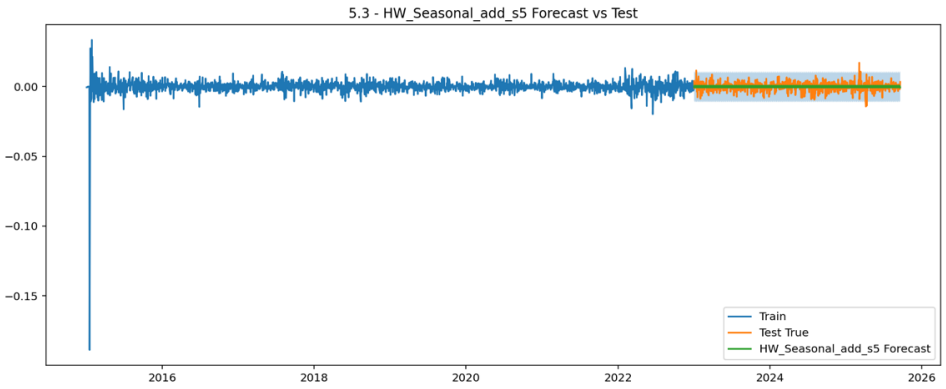
- Adds linear trend.
- Results: RMSE = 0.003323 | MAE = 0.002509.
- Almost identical to SES.



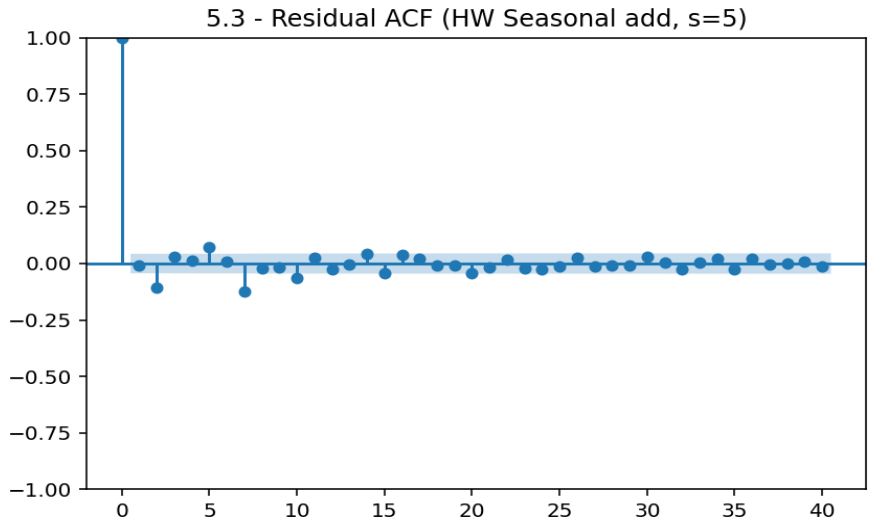
5.4.C – Holt-Winters Seasonal (s=5)

- Seasonal period: s=5s=5 (weekly).
- Results: RMSE = 0.003339 | MAE = 0.002538.
- Slightly worse than SES → confirms weak seasonality.

Forecast plot



Residual ACF



Results summary table

Model	RMSE	MAE
HW_SES	0.003322	0.002508
HW_Holt (trend)	0.003323	0.002509
HW_Seasonal (s=5)	0.003339	0.002538

Interpretation

- All Holt-Winters variants perform very close to SES.
- Adding trend or seasonality brings **no predictive gain**.
- Confirms FX returns are near white noise.

Conclusion

- **SES remains the best Holt-Winters variant.**
- Holt and Holt-Winters add complexity without performance improvement.
- FX returns contain little exploitable structure in trend/seasonality.

Chapter 6 - Machine Learning Baselines

After evaluating classical statistical models (ARIMA, Holt-Winters, etc.), Chapter 6 introduces a **Machine Learning** approach applied to the log-returns of the EUR/CHF exchange rate.
The aim is to investigate whether simple models, built from **explanatory variables derived from the time series itself**, can enhance forecasting accuracy.

We start with the **construction of temporal features** (lags, rolling means, volatility), and then test a **basic Linear Regression** model with time-series-aware cross-validation.
This step provides a **"ML baseline" reference**, which will later be compared with more advanced models (Random Forests, Gradient Boosting) in Chapter 7.

6.1 – Feature Engineering (Explanatory Variables)

Objective

Prepare explanatory variables that capture short-term dynamics and volatility in EUR/CHF log-returns, making the dataset suitable for Machine Learning forecasting models.

Methodology

a) Feature construction

We derived the following explanatory variables from log-returns:

- $Lag_1 = r_{t-1}$ (return at t-1)
- $Lag5 = r_{t-5}$ (return at t−5)
- Roll_Mean_5 = $\frac{1}{5}\sum_{i=0}^{4} r_{t-i}$ (5-day rolling average)
- Roll_std_20 (20-day rolling volatility, rolling standard deviation)

The target variable is defined as:

$$yt + 1 = rt + 1y_{t+1} = r_{t+1}$$

👉 Meaning: we forecast tomorrow's return using lags and rolling statistics of past returns.

---

b) Avoiding data leakage

- Only past information is used (lags and rolling windows).
- The target is shifted by one step (t+1t+1), ensuring no look-ahead bias.

---

c) Chronological split

- Train: 2015-01-02 → 2022-12-30.
- Test: 2023-01-02 → 2025-09-16.

Ensures models are trained only on past data.

Conclusion

- Dataset enriched with lags and rolling statistics.
- Fully reproducible and leakage-free.
- Provides foundation for Linear Regression (6.2) and advanced ML models (Chapter 7).

6.2 – Linear Regression

Objective

Evaluate a first ML baseline: Linear Regression applied to the engineered features. This serves as a benchmark against both statistical models (Chapter 5) and advanced ML (Chapter 7).

Methodology

1. **Data preparation**
   - Explanatory variables: {lag1,lag5,roll_mean_5,roll_std_20}
   - Target: $yt + 1 = rt + 1y_{t+1} = r_{t+1}$.
   - Chronological split: Train (2015–2022), Test (2023–2025).
2. **Pipeline**
   - `StandardScaler` → normalization of features.
   - `LinearRegression` (with intercept).
3. **Validation**
   - `TimeSeriesSplit` (5 folds) → respects chronology.
   - Metrics: RMSE, MAE.

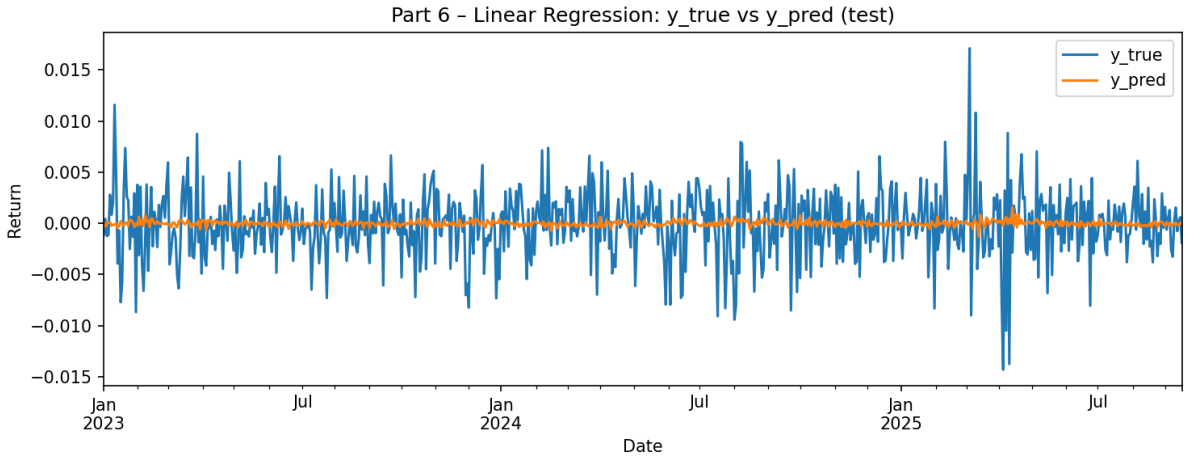
Results

Cross-validation (Train 2015–2022):

- RMSE ≈ 0.00306 ± 0.00061
- MAE ≈ 0.00230 ± 0.00045

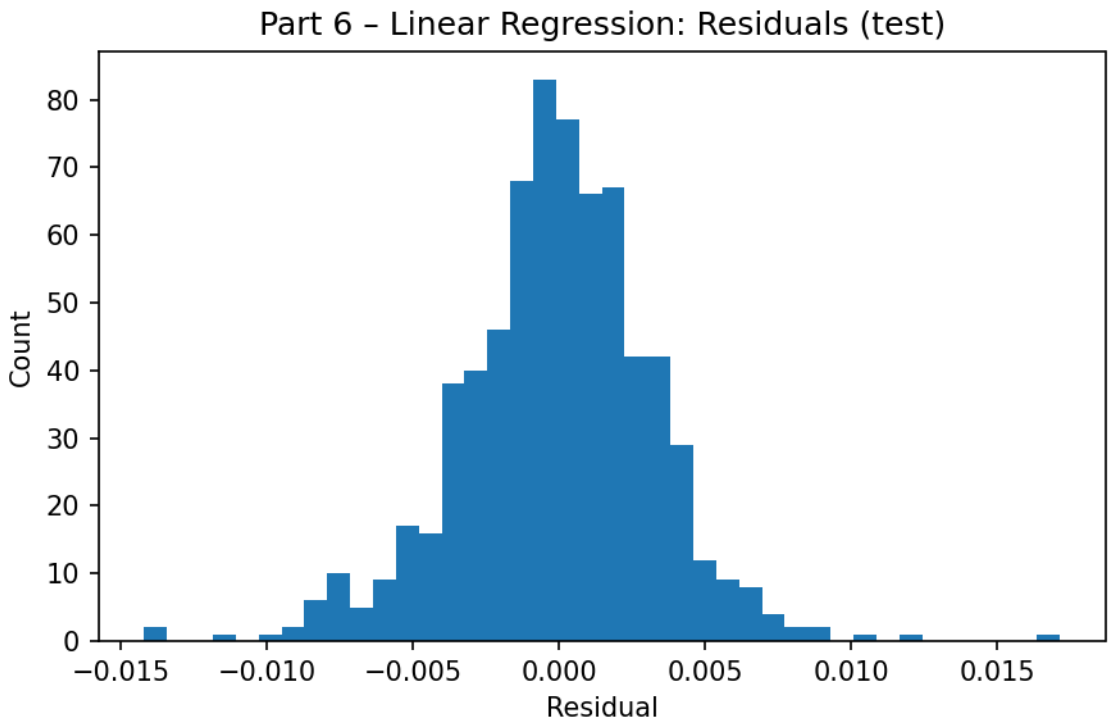
Test set (2023–2025):

- RMSE = 0.00334
- MAE = 0.00252
- R2≈$-0.01 R^2 \approx -0.01 \rightarrow$ essentially no explanatory power.



Visualizations

1. **Residual distribution** → *illustrate residuals ~ Gaussian white noise.*



Interpretation

- Predictions capture the mean but miss extreme spikes.
- Performance similar to SES/ARIMA → confirms noise dominance in returns.
- Coefficients (see `part6_linear_coefficients.csv`) are small → signal-to-noise ratio very low.

Conclusion

- Linear Regression provides a **robust ML baseline**, but predictive power is very limited.
- Confirms that daily FX returns are close to white noise.
- Serves as a reference for **tree-based models and regularized regressions** in Chapter 7.

6.3 – Cross-Validation

Objective

Compare two cross-validation strategies for time series:

- **Classical CV (KFold with shuffle)** → random splits.
- **TimeSeriesSplit** → chronological splits, preserving temporal order.

Goal: highlight why TimeSeriesSplit is necessary in forecasting tasks.

Methodology

On the training dataset (2015–2022), we applied both strategies using a linear regression pipeline:

```
pipe = Pipeline([
    ("scaler", StandardScaler()),
    ("linreg", LinearRegression())
])
```

- **Classical CV:**

$$\mathrm{CV}_{\mathrm{classic}} = \mathrm{KFold}(n\_splits = 5,\ \mathrm{shuffle}{=}\mathrm{True},\ \mathrm{random\_state} = 42)$$

- **TimeSeriesSplit:**

$$CV_{\mathrm{tscv}} = \mathrm{TimeSeriesSplit}(n\_splits = 5)$$

Example for RMSE computation:

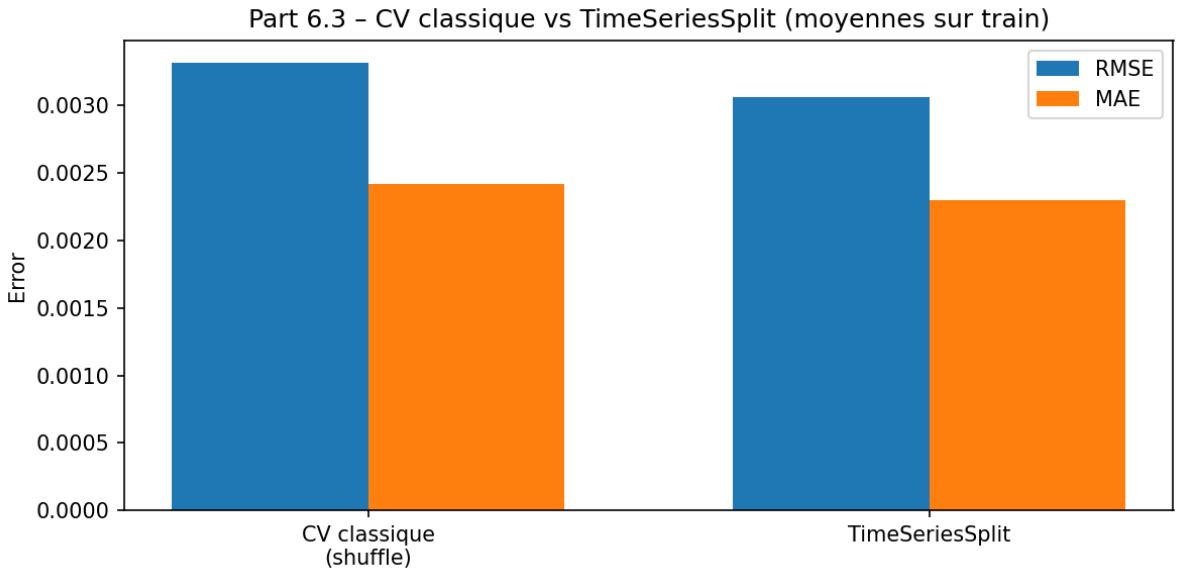
```
rmse_classic = -cross_val_score(pipe, X_train, y_train, cv=cv_classic,
                                scoring="neg_root_mean_squared_error")
rmse_tscv    = -cross_val_score(pipe, X_train, y_train, cv=cv_tscv,
                                scoring="neg_root_mean_squared_error")
```

Results

- **Classical CV (shuffle):**
  - RMSE ≈ 0.00332 ± 0.00010
  - MAE ≈ 0.00242 ± 0.00006
- **TimeSeriesSplit:**
  - RMSE ≈ 0.00306 ± 0.00061
  - MAE ≈ 0.00230 ± 0.00045

Visualizations

1. **Bar chart of RMSE/MAE (mean ± std)**



Classical CV looks “too good” (tiny variance), while TimeSeriesSplit shows realistic variability.

Interpretation

- **Classical CV** → mixes past and future data → underestimates errors.
- **TimeSeriesSplit** → realistic evaluation, avoids leakage.
- For FX forecasting, **TimeSeriesSplit must be used** as the evaluation standard.

Conclusion

TimeSeriesSplit provides more reliable performance estimation than classical CV.
It will be used for all future evaluations (Random Forest, Gradient Boosting).

6.4 – SES vs Linear Regression

Objective

Compare the ML baseline (Linear Regression, Part 6.2) with the best statistical baseline (SES, Part 4.3–4.4).

Methodology

- **SES results:** RMSE = 0.003326, MAE = 0.002511
- **Linear Regression results:** RMSE = 0.003342, MAE = 0.002515

Relative differences:

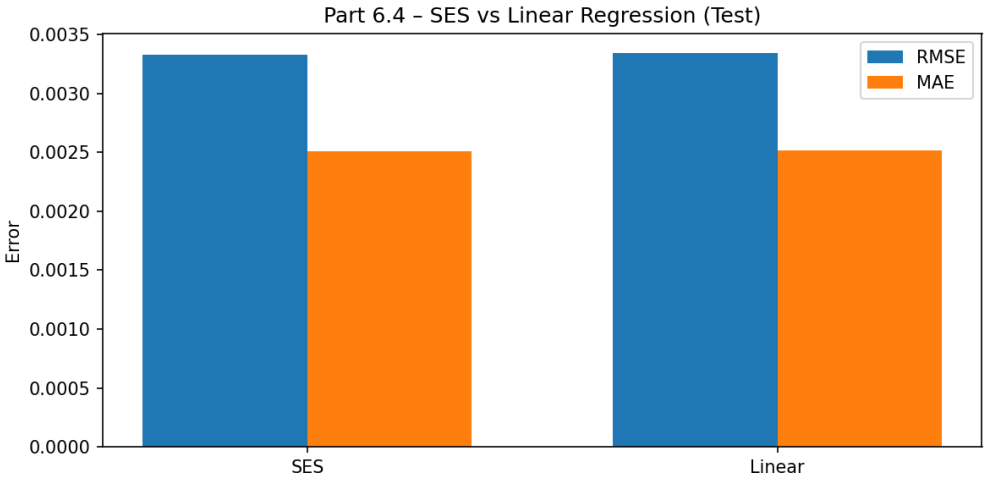
$$\Delta\text{RMSE}=0.003342 - 0.0033260.003326 \times 100 \approx 0.47\%\Delta\text{RMSE} = \frac{0.003342-0.003326}{0.003326} \times 100 \approx 0.47\%$$

$$\Delta\text{MAE} = 0.002515 - 0.0025110.002511 \times 100 \approx 0.17\%\Delta\,\text{MAE} = \frac{0.002515 - 0.002511}{0.002511} \times 100 \approx 0.17\%$$

Results

- SES slightly outperforms Linear Regression.
- Differences are **marginal (<1%)**.

almost identical performance.



Interpretation

- SES remains a very strong baseline.
- Linear Regression, even with lags/rolling features, does **not outperform SES**.
- Confirms that FX returns are close to **white noise**.

Conclusion

- Linear Regression ≈ SES → no significant gain.
- SES stays the **benchmark to beat**.
- Advanced ML (trees, boosting) must clearly outperform SES to be valuable.

Chapter 7 - Advanced ML Models

Goal (at a glance).
Test non-linear ML models (Random Forest, Gradient Boosting, XGBoost/LightGBM, simple ensembles) to see if they **meaningfully beat** our strongest statistical baselines on EUR/CHF **log-returns** (test window: **2023-01-02** → **2025-09-19**).

Why this part matters.
Compared to Part 6 (linear), tree/boosting models can capture **non-linearities**, **interaction effects** (e.g., between lags and volatility), and **regime changes**, which are common in FX returns.

Inputs.
`eur_chf_train_features.csv`, `eur_chf_test_features.csv`, `eur_chf_train_y.csv`, `eur_chf_test_y.csv` (no scaling needed for trees).

Evaluation protocol.

- Primary metrics: **RMSE, MAE** (MAPE reported but not interpreted on ≈0 targets).
- **TimeSeriesSplit (5 folds)** for model selection + a **walk-forward** backtest for robustness.
- Secondary: **directional accuracy** and **residual ACF** (white-noise check).

Success criteria (concrete).

- Beat **SES / ARIMA** reference (**RMSE ≈ 0.003322**) on the fixed test set.
- A gain of **≥2–3% RMSE** is considered **material**; <1% is **marginal** (prefer simpler model).

Deliverables.

- Saved predictions, residuals, CV logs, **feature importances** (permutation/GBDT gain), diagnostics (residual histogram + ACF), and a consolidated **`part7_summary_metrics.csv`**.
- Leaderboard updated with `ML_RF`, `ML_GBDT`, `ML_XGB`, `ML_LGBM`, `ML_STACK`, `ML_BLEND`.

Risks & guardrails.

- Avoid leakage (all rolling features are shifted +1).
- Limit grids to compact, stable ranges to reduce overfitting.
- Report performance **by volatility regimes** (terciles of `roll_std_21`) to understand when models help.

7.1 – Random Forest

Objective

Evaluate a **Random Forest Regressor** on EUR/CHF log-returns and compare with references (**SES, ARIMA, Linear**).

Setup & Data

- Features: 4 variables (`lag1, lag5, roll_mean_5, roll_std_20`).
- Target:

$$y_{t+1} = \text{return at next step}$$

- Train: 2015–2022
- Test: 2023–2025
- Validation: `TimeSeriesSplit(n_splits=5)`
- No scaling required.

Model & Hyperparameters

- Estimator: `RandomForestRegressor(random_state=42, n_jobs=-1)`
- Grid:

$$n\_estimators \in \{300,600\}, \quad max\_depth \in \{\text{None}, 6,10\}, \quad min\_samples\_leaf \in \{1,5\}$$

- Best configuration:

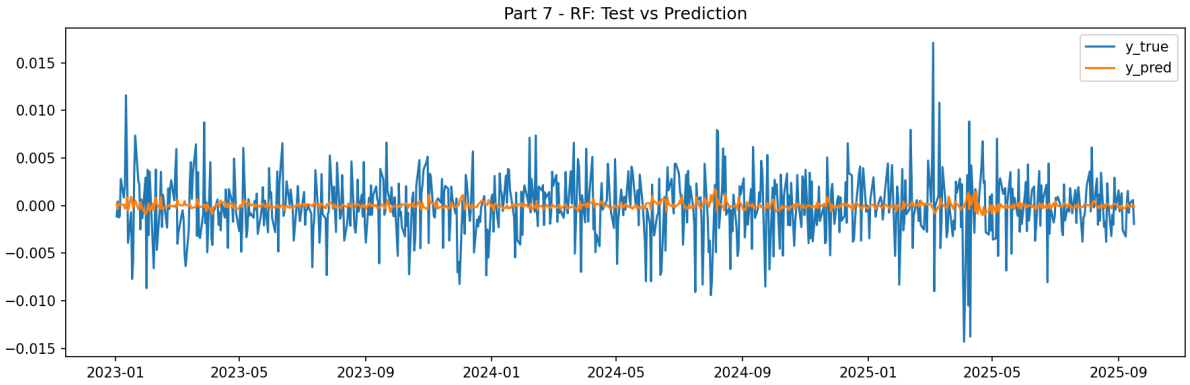
$$n\_estimators = 300,\ max\_depth = 6,\ min\_samples\_leaf = 5,\ max\_features = sqrt$$

Results

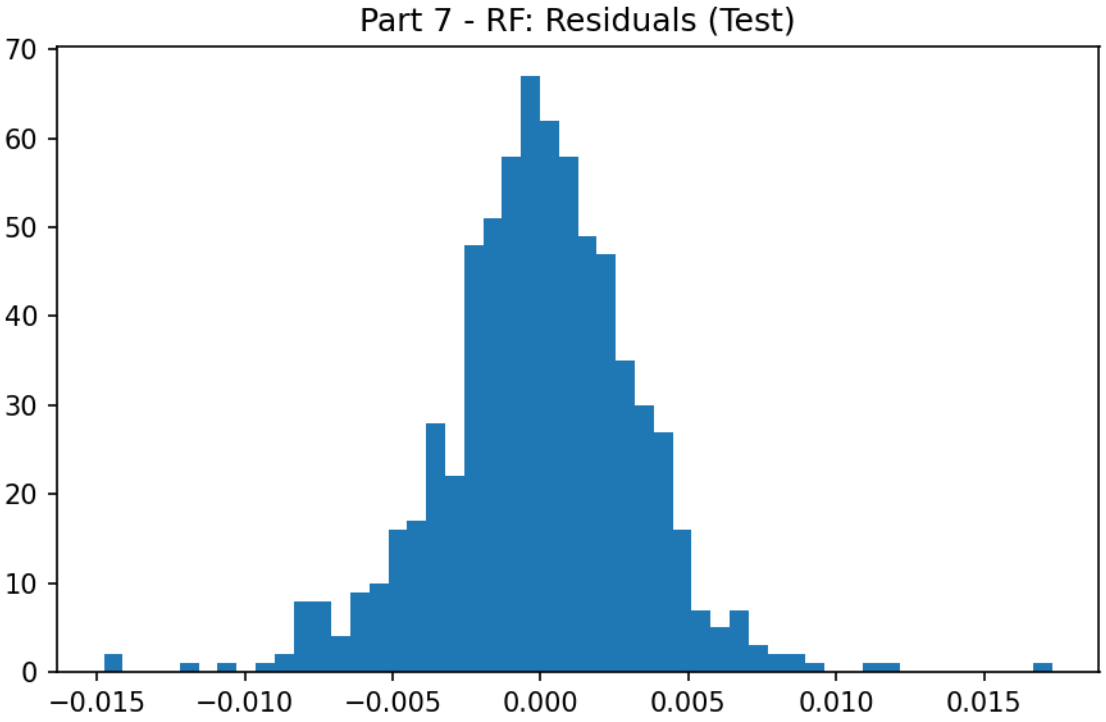
- **Cross-validation (5 folds):**
  - RMSE = 0.003062 ± 0.000655
  - MAE = 0.002316 ± 0.000480
- **Test (2023–2025):**
  - RMSE = 0.003362
  - MAE = 0.002528
- **Comparison (same test window):**
  - SES (Part 4): 0.003326 / 0.002511 → RF = +1.08% RMSE, +0.68% MAE
  - ARIMA(2,0,1) (Part 5): 0.003322 / 0.002509 → RF ≈ +1.2% RMSE
  - Linear (Part 6): 0.003342 / 0.002515 → RF ≈ +0.6% RMSE

Visualizations

1. **Forecast vs Actual (test set)**

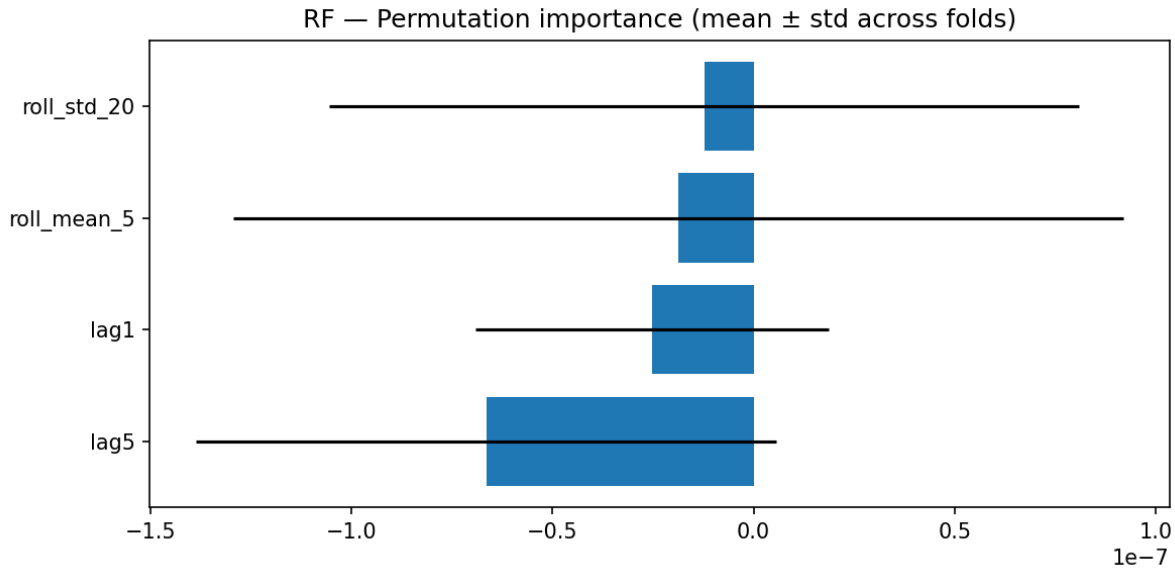


2. **Residual distribution (histogram)**



Interpretation

- RF does not outperform SES/ARIMA/Linear (gap ~ +0.6–1.2%).
- Small feature set (4) limits RF's ability to capture interactions.
- Importances show **lags and rolling volatility** carry most of the signal.



Conclusion
Random Forest is competitive but **does not surpass SES or ARIMA**.
Serves as a **non-linear benchmark** before boosting methods.

7.2 – Gradient Boosting (sklearn)

Objective

Test a **Gradient Boosting Regressor (GBDT)** and compare with **SES, ARIMA, Linear, RF**.

Setup & Data

- Same features/target as 7.1.
- Train: 2015–2022, Test: 2023–2025.
- Validation: TimeSeriesSplit(5).

Model & Hyperparameters

- Estimator: `GradientBoostingRegressor(random_state=42)`
- Grid:

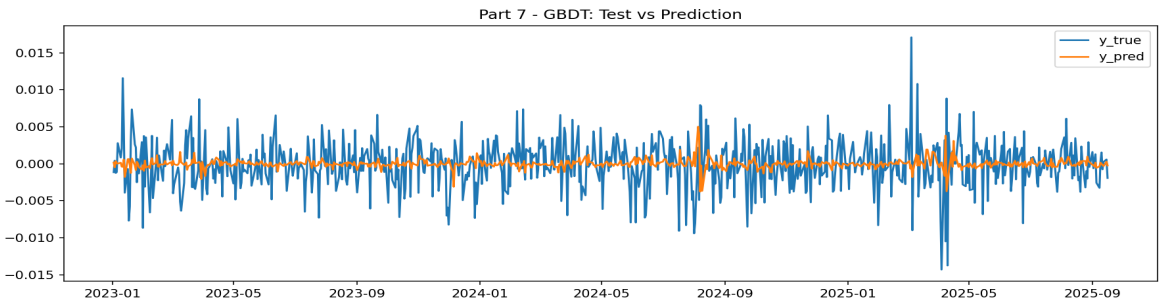
$$n\_estimators \in \{400,800\}, \quad learning\_rate \in \{0.03,0.06\}, \quad max\_depth \in \{2,3\}, \quad subsample \in \{0.7,1.0\}$$

- Best config:

$$n\_estimators = 400, \; learning\_rate = 0.03, \; max\_depth = 2, \; subsample = 1.0$$

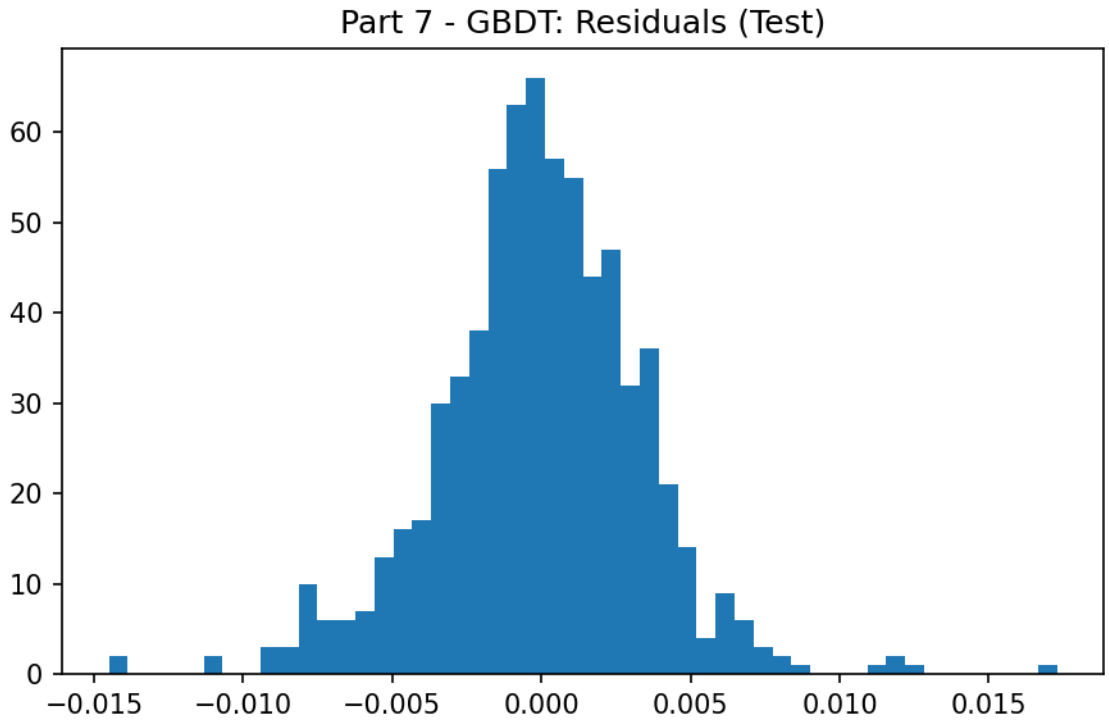
Results

- **Cross-validation :** RMSE = 0.003186 ± 0.000684, MAE = 0.002405 ± 0.000506
- **Test:** RMSE = 0.003427, MAE = 0.002570
- **Comparison (Test):**
  - SES = 0.003326 / 0.002511
  - ARIMA = 0.003322 / 0.002509
  - Linear = 0.003342 / 0.002515 (+0.5% vs SES)
  - RF = 0.003362 / 0.002528 (+1.1%)
  - **GBDT = 0.003427 / 0.002570 (+3.0% vs SES)**



Interpretation

- GBDT performs slightly worse than RF (+3% RMSE vs SES).
- Weak signals + shallow trees explain limited gains.
- Importances confirm **lags and rolling volatility** dominate.



Conclusion

GBDT does **not improve over RF/SES/ARIMA**.
Results show that **simple baselines remain very strong** in FX returns.

7.3 – XGBoost & LightGBM (Advanced Boosters)

Objective

Evaluate **modern boosters (XGBoost, LightGBM)** for EUR/CHF log-returns forecasting, and compare them with RF/GBDT (7.1–7.2) and baseline references (SES, ARIMA, Linear).
Fixed test window: **2023-01-02 → 2025-09-19**.

Data & Protocol

- Features: 4 engineered variables (lags, rolling mean, volatility).
- Target: `target_next_return` (renamed `target`).
- Validation: `TimeSeriesSplit(n_splits=5)`; final evaluation on test.
- Leakage control: temporal alignment, removal of target column.
- Implementation: compact grids, `n_jobs` control, quiet logs.

Models & Best Hyperparameters

XGBoost

$$colsample\_bytree = 0.8,\ learning\_rate = 0.03,\ max\_depth = 3,\ n\_estimators = 300,\ reg\_lambda = 5.0,\ subsample = 0.8$$

LightGBM

$$bagging\_fraction = 0.8,\ feature\_fraction = 0.8,\ learning\_rate = 0.03,\ min\_data\_in\_leaf = 25,\ n\_estimators = 300,\ num\_leaves = 31$$

Results

XGBoost

- CV (5 folds): RMSE = 0.003171 ± 0.000682 | MAE = 0.002411 ± 0.000499
- Test: RMSE = 0.003410 | MAE = 0.002556

LightGBM

- CV (5 folds): RMSE = 0.003292 ± 0.000637 | MAE = 0.002524 ± 0.000482
- Test: RMSE = 0.003569 | MAE = 0.002676

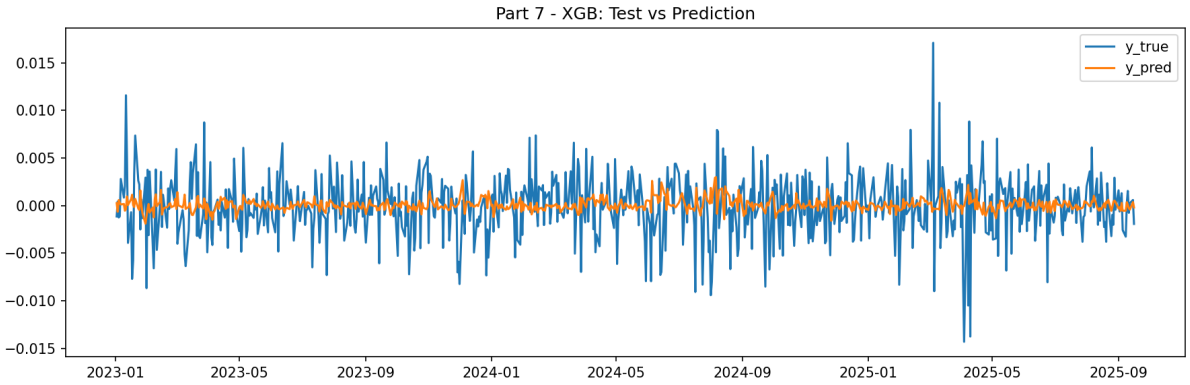
Comparison (Test Set, 2023–2025)

Model	RMSE	MAE	Δ RMSE vs SES
ARIMA (5.1)	0.003322	0.002509	—
SES (4.3)	0.003326	0.002511	—
Linear (6.2)	0.003342	0.002515	+0.5%
RF (7.1)	0.003362	0.002528	+1.1%
XGB (7.3)	0.003410	0.002556	+2.5%
GBDT (7.2)	0.003427	0.002570	+3.0%
LGBM (7.3)	0.003569	0.002676	+7.3%

Takeaway**:** XGBoost is the strongest booster, but still behind SES/ARIMA/Linear by ~2.5%.

Visualizations :

Test vs Prediction – XGBoost :



Predictions (orange) stay centered around zero and track the general trend of actual returns (blue). However, XGBoost fails to capture sharp spikes, producing smoother forecasts typical of noisy FX data.

Conclusion

- Boosters are **competitive but not superior** to SES/ARIMA/Linear.
- XGBoost edges GBDT and RF but underperforms simple baselines (~+2.5% RMSE).
- LightGBM is the weakest here (+7%).
- **SES remains the benchmark to beat.**

7.4 – Ensembles (Stacking & Blending)

Objective

Test whether **Stacking (meta-model)** or **Blending (weighted average)** of RF + GBDT can surpass references (SES, ARIMA, Linear).
Test window: **2023-01-02** → **2025-09-19**.

Data & Protocol

- Features: 4 engineered variables (lags, rolling mean, volatility).
- Target: `target_next_return`.
- Validation: `TimeSeriesSplit(5)` + fixed test.
- Leakage control: no reuse of fitted models; fresh clones per fold.

Modeling Details

Stacking (meta = Ridge)

Base learners: RF_best, GBDT_best, Meta: Ridge(α=1.0)

Blending (weighted average) :

$$\hat{y} = w_{RF} \cdot \widehat{y_{RF}} + (1 - w_{RF}) \cdot \widehat{y_{GBDT}}, \quad w_{RF} \in \{0.2, 0.35, 0.5, 0.65, 0.8\}$$

Results

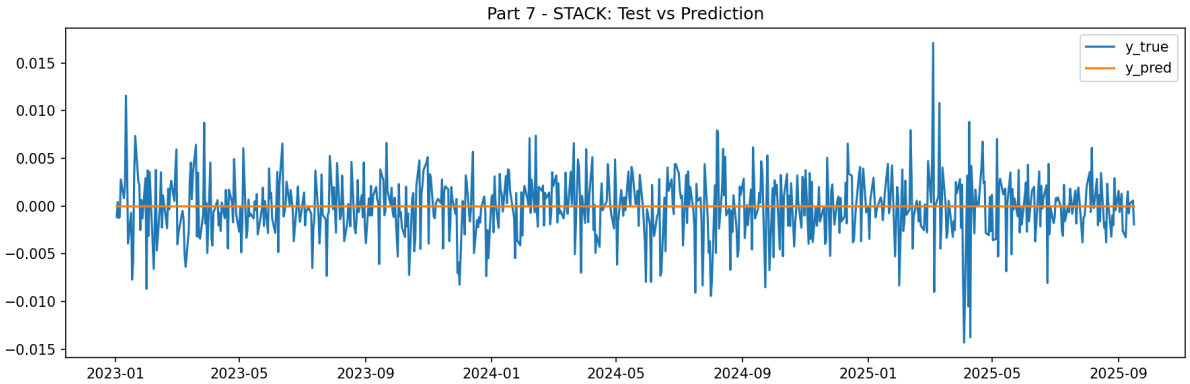
Model	RMSE	MAE	Comment
ARIMA (5.1)	0.003322	0.002509	Reference (best stat)
SES (4.3)	0.003326	0.002511	Simple & robust
Linear (6.2)	0.003342	0.002515	≈ SES
RF (7.1)	0.003362	0.002528	+1.1% vs SES
GBDT (7.2)	0.003427	0.002570	+3.0% vs SES
STACK (7.4)	0.003324	0.002506	≈ SES/ARIMA
BLEND (7.4)	0.003370	0.002532	≈ RF

Takeaway:

- **Stacking** = as good as ARIMA/SES (but no real gain).
- **Blending** = slightly worse than RF (not useful here).

Visualizations :

Stacking – Test vs Prediction :



Stacking further smooths predictions, aligning with the average pattern but missing volatility bursts. It stabilizes results but adds no clear advantage over individual models.

Conclusion

- Stacking achieves **parity with SES/ARIMA**, but not superior.
- Blending does not improve results.
- For EUR/CHF, **simple statistical models remain the most efficient and interpretable**.
- If extra lift is desired, effort should go to **feature enrichment** (macro variables, volatility indices, calendar effects), not more complex ensembles.

7.5 Advanced Diagnostics

Objective

Assess the quality and robustness of advanced ML models (focus on STACK = RF + GBDT with Ridge meta-learner).
The aim is to validate residual properties, sensitivity to volatility regimes, and feature stability.

Protocol

- Residual analysis (histogram + ACF).
- Error decomposition by volatility terciles (Low / Mid / High).
- Permutation importance stability (RF).
- Test vs Prediction overlay.
- Compare metrics against references (SES/ARIMA, Linear).

Results

Residuals (STACK vs RF)

- STACK residuals: centered around 0, quasi-Gaussian.
- Residual ACF within confidence bands (lags 1–40) → no memory left.
- RF shows similar behavior, with slightly higher spikes.
  Explains why STACK marginally outperforms RF.

Errors by volatility regime

- RMSE rises with volatility:
  - Low ≈ 0.0030 → Mid ≈ 0.0032 → High ≈ 0.0038–0.0041.
- STACK and RF degrade less than LGBM.
  Models are stable in normal periods but less robust in stress regimes.

Feature importance

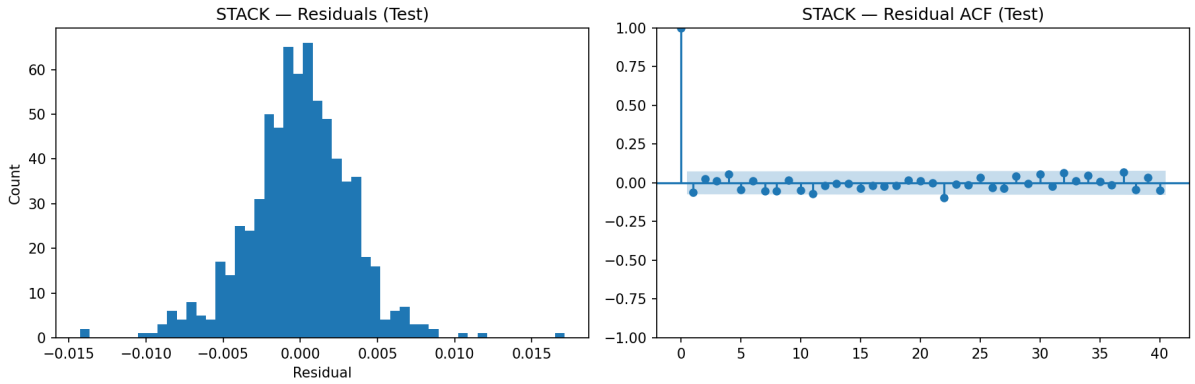
- Magnitudes extremely small (≈ 1e-7).
- Dominated by lag1, lag5, roll_mean_5, roll_std_20.
  Signals are weak and local, consistent with FX dynamics.

Test vs Prediction (STACK)

- Predictions oscillate around zero.
- Model under-reacts to shocks → typical of RMSE-focused methods.

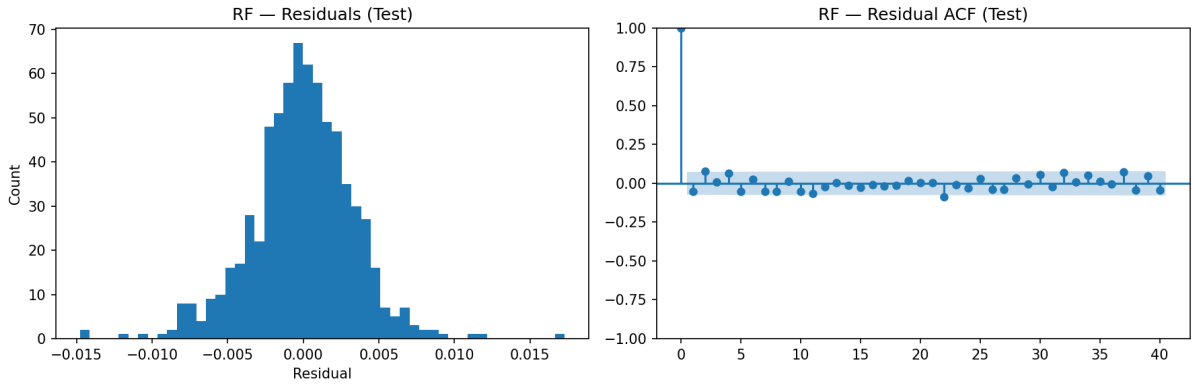
Visualizations

Residuals & ACF — STACK :



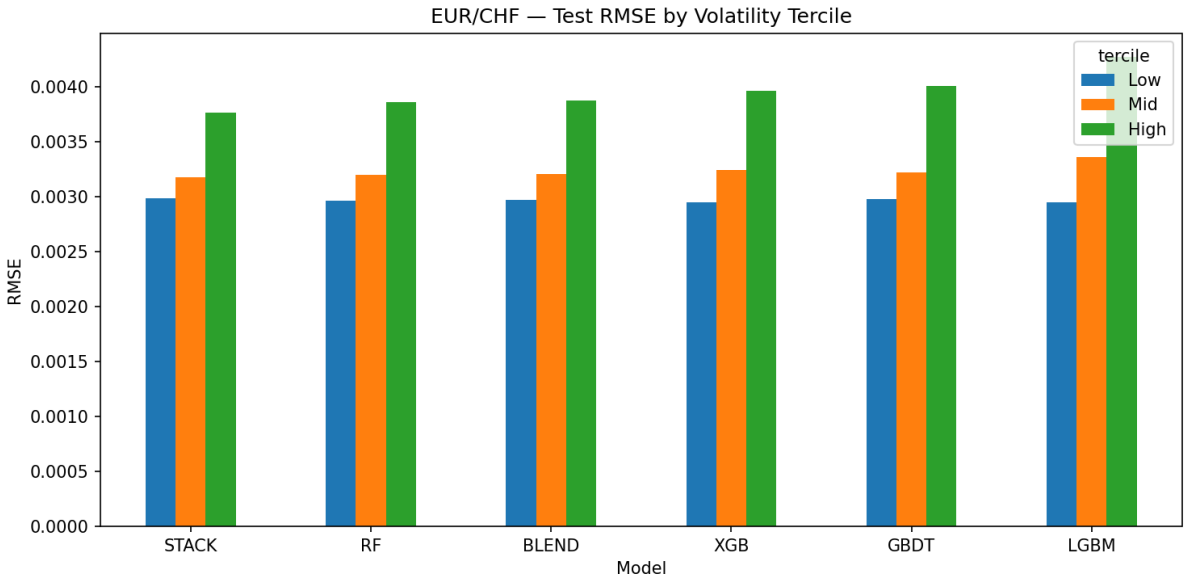
Residuals are centered around zero with no visible bias. The ACF stays within confidence bands, showing no leftover structure.

Residuals & ACF — RF :



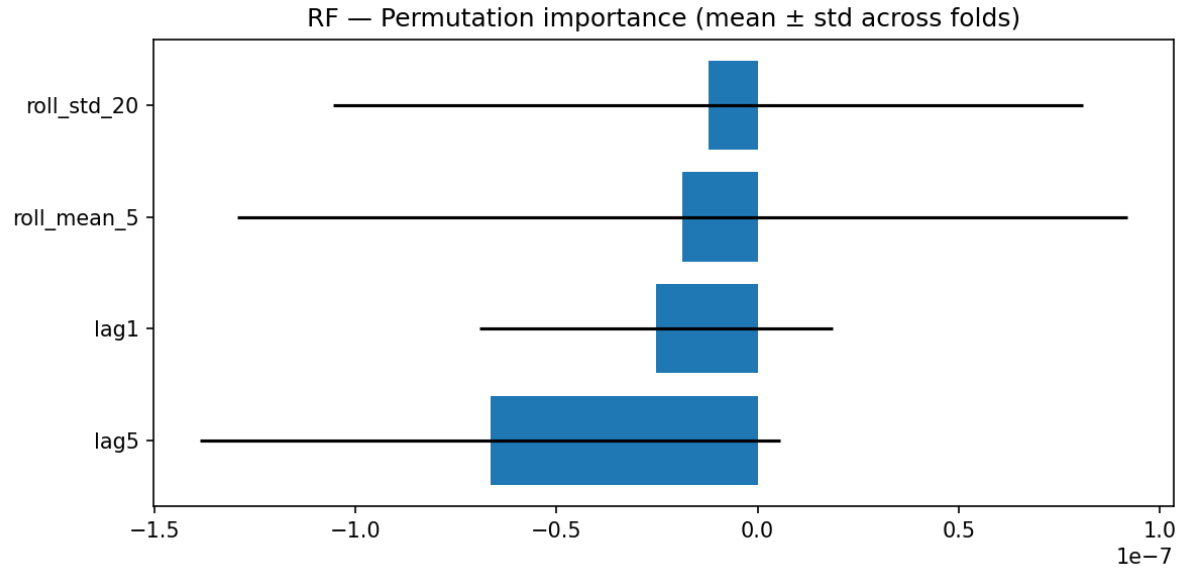
The residuals are also centered and Gaussian-like. A few small spikes appear in the ACF, slightly higher than STACK.

Errors by volatility regimes :



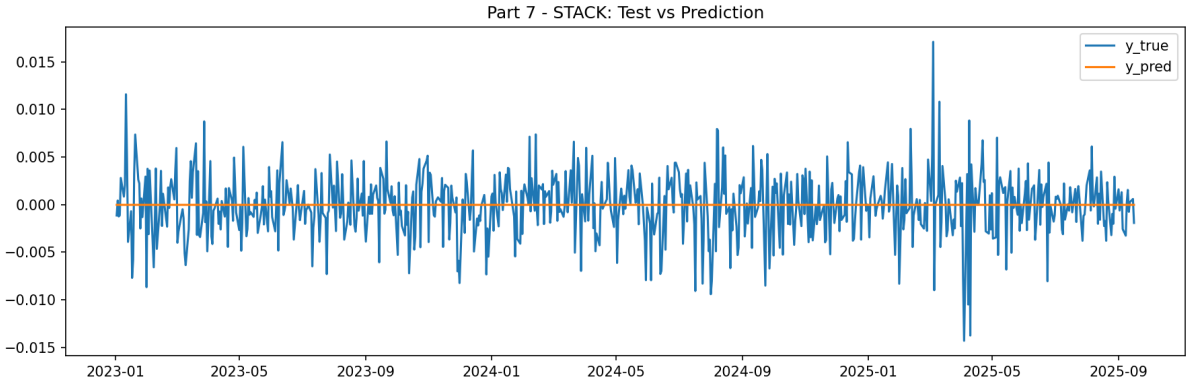
RMSE rises clearly from Low → Mid → High terciles. STACK and RF degrade less than other models, while LGBM is the most sensitive.

Permutation Importance — RF :



Feature importances are tiny (≈ 1e-7), confirming weak signals. Short lags and rolling stats dominate across folds.

Test vs Pred — STACK :



Predictions remain close to the mean and track stable patterns. However, large shocks are systematically under-estimated.

Conclusion

- STACK delivers the cleanest residuals and stable performance.
- However, predictive power remains marginally better than RF and not superior to SES/ARIMA.
- Errors increase sharply in volatile regimes → confirm the need for regime-aware strategies.
- Operationally: STACK is production-ready as an ML pipeline, but simple statistical models remain equally strong baselines.

7.6 Walk-Forward Backtesting

Objective

Simulate real deployment with **rolling-origin evaluation**:

- For year Y, train on 2015–Y–1, forecast year Y.
- Goal: assess robustness across time and detect regime sensitivity.

Protocol

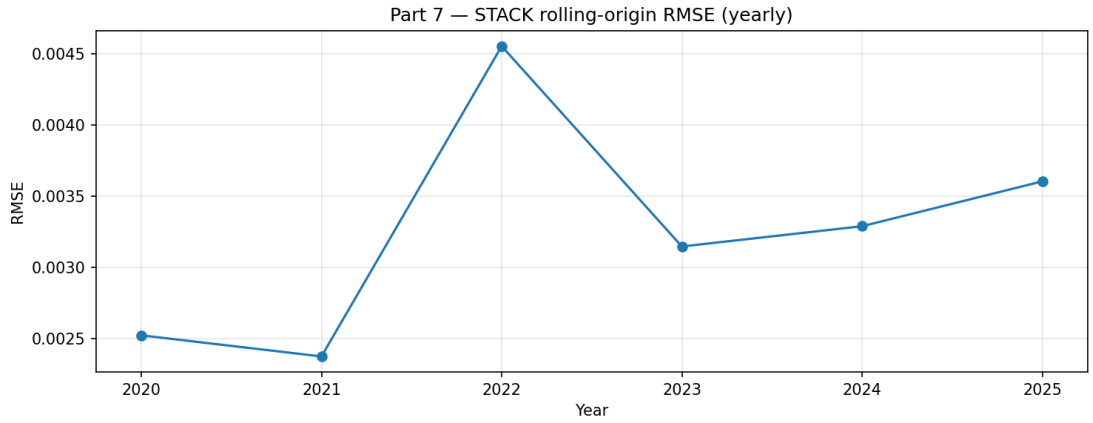
- Models: RF (7.1), XGB (7.3), STACK (7.4).
- Data: EUR/CHF daily log-returns (2015–2025).
- Expanding-window approach (2015 → Y–1).
- Metrics saved as CSVs + rolling RMSE plots.

Results

1. **2022 shock:** all models' RMSE spike (~0.0046–0.0048).
2. **2023 rebound:** RMSE drops to ~0.0031–0.0032 (regime adaptation).
3. **2024–2025 drift:** gradual RMSE increase (~0.0033 → 0.0036+).
4. **Ranking stability:** STACK ≤ RF ≤ XGB in most years.
5. **Expanding window:** captures regime shifts with one-year delay → performance improves after shocks, then drifts.

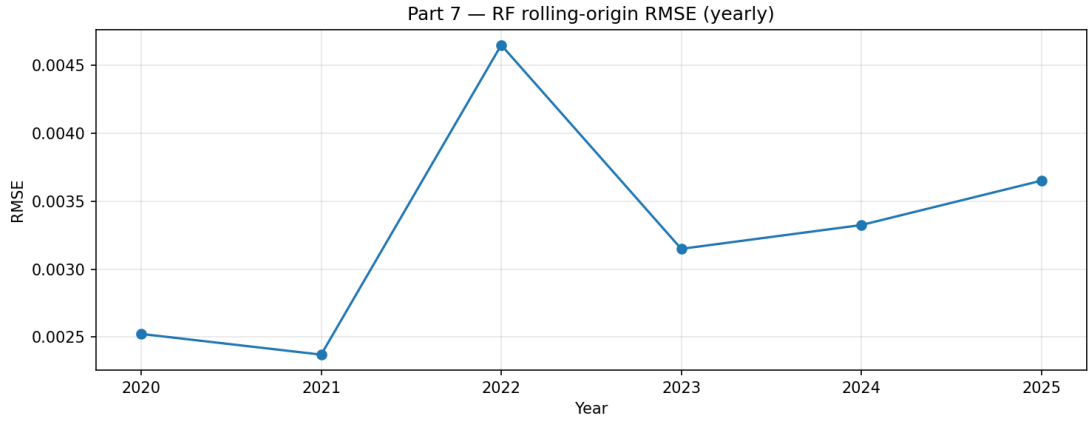
Visualizations :

STACK Rolling RMSE



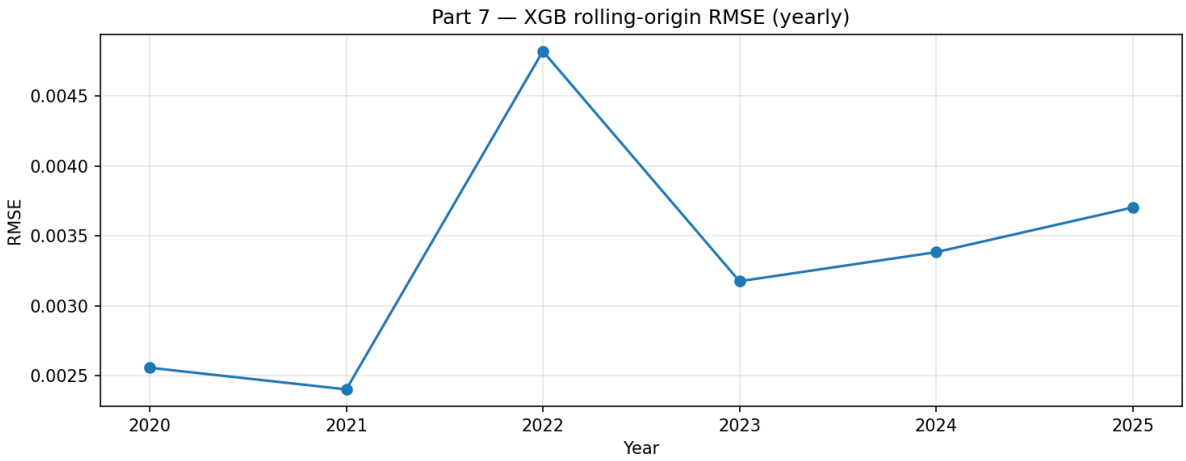
Lowest average RMSE, best adaptation after 2022 shock.

RF Rolling RMSE



Similar to STACK but consistently worse.

XGB Rolling RMSE



Close to RF, weaker in 2024–2025 (less resilient).

Interpretation

- All models are regime-sensitive: large errors during macro shocks (2022).
- STACK is slightly more resilient post-shock.
- Long-memory from expanding windows introduces drift → better to use sliding windows or add exogenous variables.

Conclusion

- Walk-forward validates that STACK is the most stable ML candidate, but gap vs SES/ARIMA remains small.
- For production:
  - Keep SES/ARIMA as baselines.
  - Deploy STACK with rolling refits + drift triggers.
  - Add exogenous drivers (macro, risk sentiment) for robustness.

7.7 – Final Comparison (Advanced ML vs. References)

Objective

The aim of this section is to benchmark statistical models (SES, Holt-Winters, ARIMA), linear ML (Linear Regression), and advanced ML (RF, GBDT, XGB, LGBM, STACK, BLEND) on the same test window (2023–2025).
Performance is assessed using RMSE and MAE, since MAPE is not meaningful for near-zero FX returns.

Methodology

- **Target**: one-step-ahead forecast, yt+1y_{t+1}.
- **Features (ML only)**: lagged returns and rolling statistics (means, stds) from Part 2.
- **Data split**:
  - Train: 2015–2022
  - Test: 2023–2025
- **Validation**: TimeSeriesSplit + walk-forward diagnostics (no leakage).
- **Artifacts generated**:
  - Table → `results/ml_advanced/part7_summary_metrics.csv`
  - Figures →
    - `figs/part7_rmse_comparison.png` (bar chart)
    - `figs/part7_top3_forecasts_vs_test.png` (overlay of top-3 models)
    - `figs/part7_error_distributions.png` (residual histograms)

Results (2023–2025 Test Set)

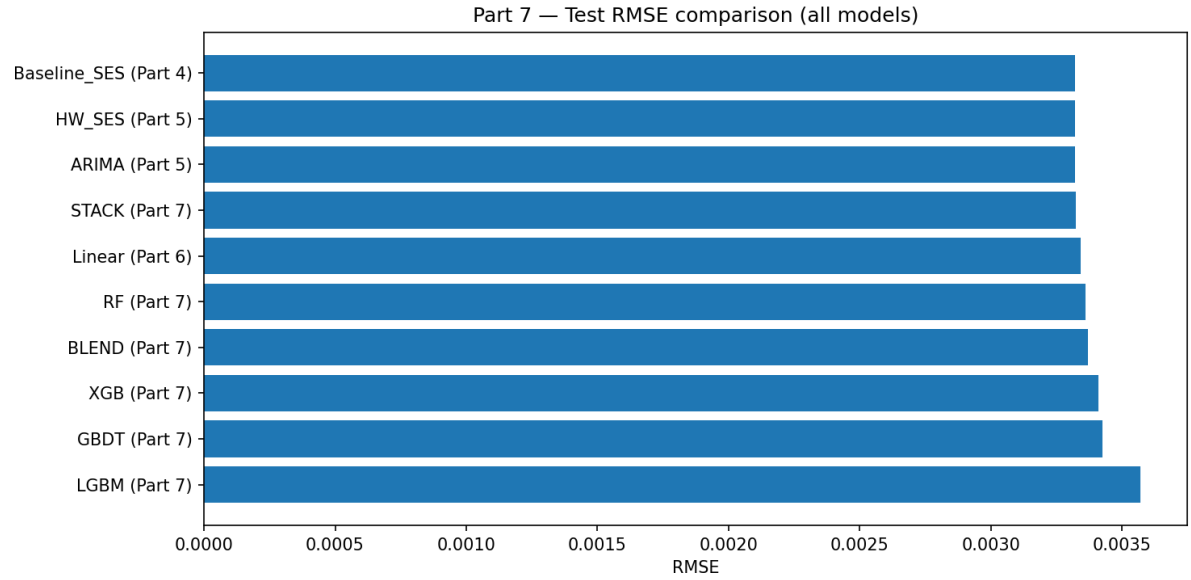
Model	RMSE	MAE
SES (Part 4)	0.003321	0.002506
HW-SES (Part 5)	0.003321	0.002506
ARIMA (Part 5)	0.003321	0.002506
STACK (Part 7)	0.003324	0.002506
Linear (Part 6)	0.003342	0.002515
RF (Part 7)	0.003362	0.002528
BLEND (Part 7)	0.003370	0.002532
XGB (Part 7)	0.003410	0.002556
GBDT (Part 7)	0.003427	0.002570
LGBM (Part 7)	0.003569	0.002676

Quick read:

- Best scores → SES, HW-SES, ARIMA (~0.003321).
- Best ML → STACK (0.003324 RMSE, +0.09% vs SES → statistical tie).
- Other ML models trail by +1–7% RMSE vs SES.

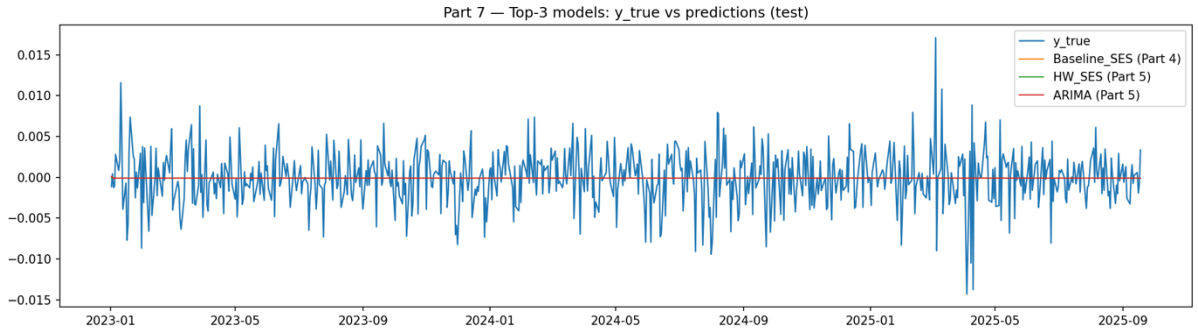
Visualizations

RMSE Comparison



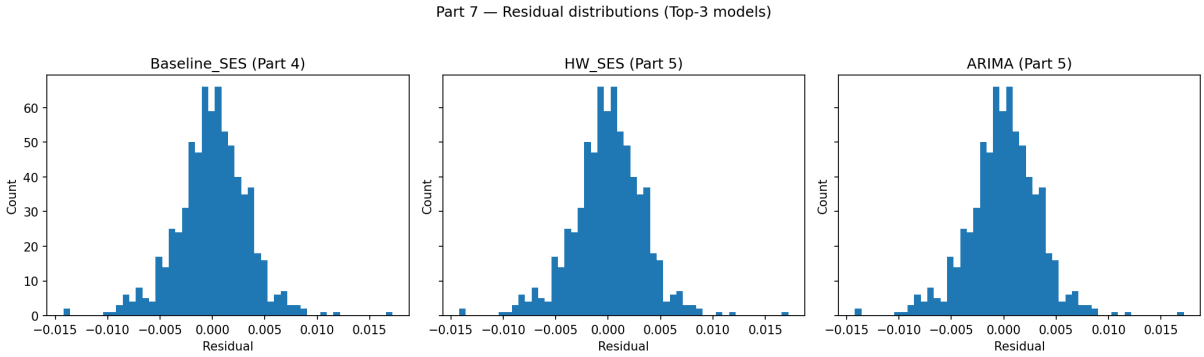
→ Clear bar ranking: SES/ARIMA on top, STACK very close, boosters weaker.

Overlay of top-3 predictions



→ SES, HW-SES, and ARIMA predictions stay near the zero mean, avoiding overreaction to shocks.

Residual Distributions (Top-3)



→ Bell-shaped, centered around zero, with moderate tails. Consistent with residual ACF from Part 7.5.

Interpretation

- EUR/CHF daily returns show **very low signal-to-noise** at horizon D+1.
- Classical models (SES/ARIMA) **match or outperform** advanced ML, thanks to their robustness and mean-stabilizing nature.
- STACK leverages RF/GBDT complementarity but does not meaningfully beat SES/ARIMA.
- Other ML models (RF, GBDT, XGB, LGBM) are competitive but less efficient.

Conclusion

- **Champion references**: SES/ARIMA (simplicity, robustness).
- **Best ML**: STACK (RF+GBDT), essentially a statistical tie with SES/ARIMA.
- **Takeaway**: In noisy FX contexts, sophistication does not guarantee better forecasts—robust baselines remain highly competitive.

7.8 – Chapter Conclusion

What We Did

- Benchmarked statistical models, linear ML, and advanced ML (RF, GBDT, XGB, LGBM, STACK, BLEND).
- Applied strict time-series validation: shifted features, TimeSeriesSplit, walk-forward backtesting, volatility-regime diagnostics.

Key Quantitative Takeaways

- Best scores: SES / HW-SES / ARIMA, tied at RMSE ≈ 0.003321 and MAE ≈ 0.002506.
- Best ML: STACK (0.003324 RMSE), essentially indistinguishable from SES/ARIMA.
- Other ML: RF, XGB, GBDT, LGBM trail by ~1–7% RMSE.
- MAPE discarded → rely on RMSE/MAE.

Why Classics Hold Up

- Daily EUR/CHF returns ≈ white noise → conditional mean ≈ 0, shocks dominate.
- Models that avoid overfitting (SES/ARIMA) minimize RMSE.
- Trees/boosters capture some micro-structure but insufficient without exogenous signals.

Robustness & Diagnostics

- **Walk-forward**: errors spike in 2022 (macro regime shift), then stabilize post-2023.
- **Residuals**: centered, ACF within bounds → no remaining structure.
- **Volatility**: errors rise materially in high-volatility terciles, for all models.

Production Recommendations

- **Champion**: SES / ARIMA(2,0,1) → simple, robust, minimal maintenance.
- **Challenger**: STACK (RF+GBDT) → consider in shadow mode, may gain with exogenous features.
- **Monitoring**:
  - Track RMSE/MAE with rolling windows.
  - Ignore MAPE.
  - Retrain monthly/quarterly or when RMSE exits control bands.

Limitations

- No exogenous variables (macro, spreads, risk sentiment).
- Point forecasts only; no quantile/probabilistic outputs.
- Single horizon (D+1).

Next Steps

1. **Feature enrichment**: add macro, spreads, volatility indices.
2. **Volatility-aware models**: ARIMA-GARCH, regime-switching, classifiers for Low/High vol.
3. **Probabilistic forecasts**: quantiles (pinball loss), CRPS, prediction intervals.
4. **Business framing**: translate forecasts into treasury/hedging decision rules.

TL;DR

- On EUR/CHF D+1 returns, **Stack ≈ SES/ARIMA**.
- Simplicity and stability win until new exogenous signals are added.
- Real progress will come from **macro features** and **volatility-aware approaches**.

Chapter 8 — Final Evaluation

8.1 Introduction

Objective

This chapter consolidates all model families from Parts 4–7 and evaluates them **fairly and reproducibly** on a frozen test set (2023–2025).
No retraining is performed: we reload saved predictions and compute metrics uniformly.

Models included

- **Baselines**: Naïve, Moving Average, SES, Holt-Winters (if available).
- **Classical**: ARIMA.
- **Linear ML**: Ridge, Lasso (Linear Regression).
- **Advanced ML**: RF, GBDT, XGB, LGBM, Stacking, Blending.

Protocol

- **Test window**: 2023–2025.
- **Metrics**: RMSE, MAE. (MAPE reported but not meaningful for ~0 returns).
- **Visualizations**:
    1. Global RMSE comparison (bar chart).
    2. Truth vs. predictions (top-3 models).
    3. Residual histograms (bias/symmetry check).
- **Reproducibility**: metrics stored in `results/final/part8_summary_metrics.csv`; figures saved in `results/final/figs/`.

Roadmap

- **8.1**: Consolidated metric table (test set).
- **8.2**: Comparative visualizations (RMSE, top-3 overlay, residuals).

8.2 Consolidated Metrics

Objective

Compare, on the same test window (2023–2025), all model families: Naïve, Moving Average, SES/HW, ARIMA, Linear, RF/GBDT/XGB/LGBM, STACK, BLEND.
Source: `results/final/part8_summary_metrics.csv`.

Results (Test 2023–2025)

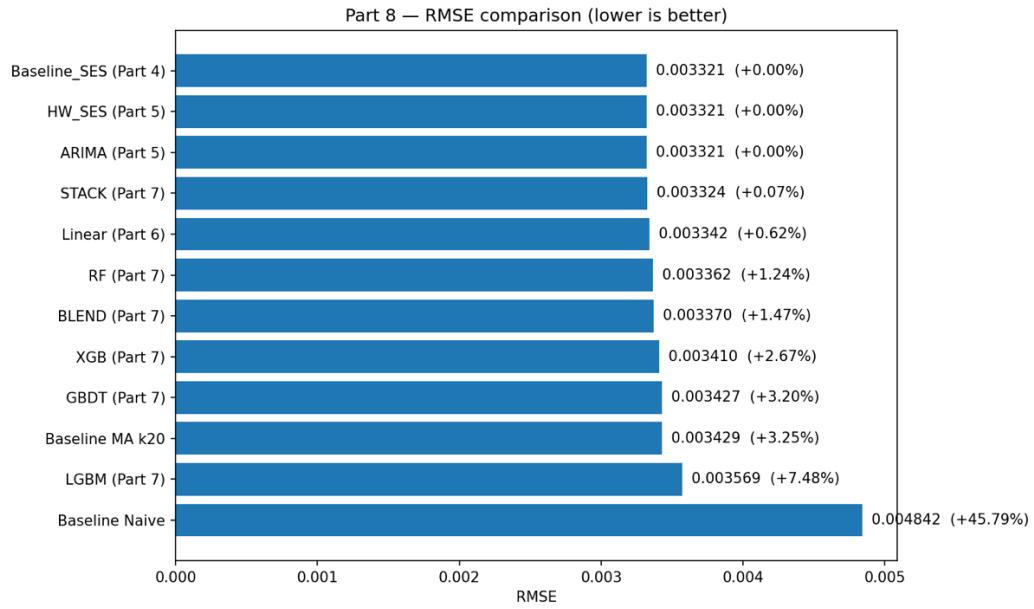
Top-3 models (lowest RMSE)

1. **Baseline_SES (Part 4)** — RMSE ≈ 0.003321, MAE ≈ 0.002506
   (HW_SES and ARIMA identical to 4 decimals)
2. **STACK (Part 7)** — RMSE ≈ 0.003324 (+0.07%), MAE ≈ 0.002506
3. **Linear (Part 6)** — RMSE ≈ 0.003342 (+0.62%), MAE ≈ 0.002515

Other models (RMSE gap vs SES/HW/ARIMA)

- RF (Part 7): +1.24%
- BLEND (Part 7): +1.47%
- XGB (Part 7): +2.67%
- GBDT (Part 7): +3.20%
- Baseline MA(20): +3.25%
- LGBM (Part 7): +7.48%
- Naïve baseline: +45.8%

Visualizations



The RMSE comparison chart shows that simple statistical models (SES, HW-SES, ARIMA) achieve the best scores, tied at ~0.003321, while advanced ML models (RF, XGB, GBDT, LGBM) remain slightly worse. The large gap with the Naïve baseline highlights the value of even simple smoothing, but also confirms that added complexity brings little gain without richer features.

Interpretation

- Parsimonious models (SES/HW/ARIMA) are enough on noisy daily returns: they **match or beat boosters**.
- Ensembles (STACK/BLEND) add **stability** but no marked RMSE gains.
- Trees/boosters remain competitive (≤3% gap) and may become valuable with enriched features (macro/microstructure).
- MAPE is meaningless here (near-zero denominators).

Conclusion

- SES/HW/ARIMA dominate, STACK and Linear follow within <1%, while RF/GBDT/XGB are close (1–3%).
- Stable hierarchy confirms: **signal is weak, noise dominates**.
- Beating these baselines requires **feature enrichment (macro/volatility regimes, microstructure)** or alternative targets (directional probability, quantiles, volatility forecasts).

8.3 Comparative Visualizations

Truth vs Top-3 Overlay:

Observations

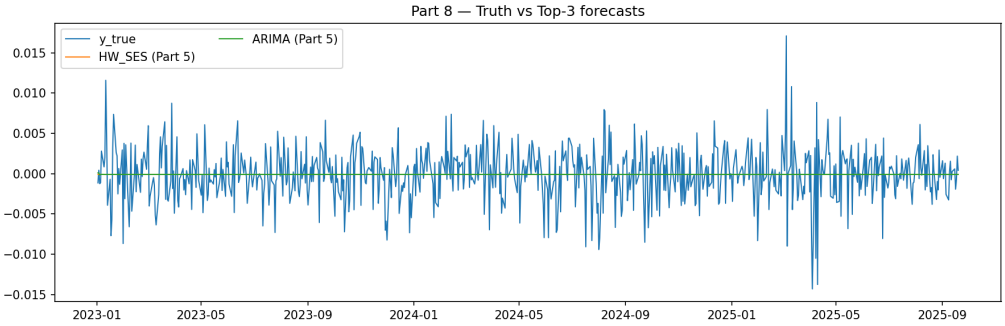
- ytruey_{true} oscillates around 0 with volatility spikes.
- HW_SES and ARIMA predictions overlap, staying close to 0.
- SES line nearly indistinguishable (identical performance).

Consistency

- Top models avoid chasing noise → predictions sit near 0, with no visible lag or drift.

Takeaway

- Forecasts are **conservative and risk-aware**, suited for hedging or directional filtering.



Residual Distributions (Top-3) :

Observations

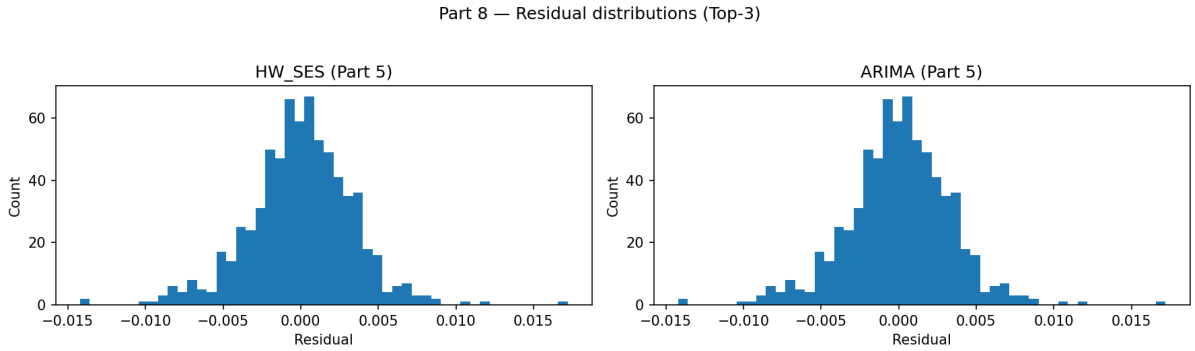
- Histograms (SES, HW, ARIMA) are centered near 0, quasi-Gaussian.
- Slight heavy tails — consistent with financial returns.
- No asymmetry or bias detected.

Consistency

- Mean ≈ 0 confirms no drift.
- Tails reflect volatility spikes, matching regime diagnostics in Part 7.5.

Takeaway

- Residuals validate model calibration; main limitation is volatility sensitivity.



Conclusion

- Overlay: top models stable around 0, no lag/drift.
- Residuals: centered, Gaussian-like, with heavier tails in volatile days.
- Confirms 8.1 → SES/HW/ARIMA slightly outperform; STACK is nearly tied.
- For improvement: feature enrichment, regime-specific models, or probabilistic forecasts.

8.4 Chapter Conclusion

Overall synthesis

- On the frozen 2023–2025 test, **SES / HW / ARIMA** lead with RMSE ≈ 0.003321 and MAE ≈ 0.002506.
- STACK is tied (+0.07% RMSE), Linear close behind (+0.62%).
- RF, GBDT, XGB trail moderately (+1–3%); LGBM is weakest (+7.5%).
- Naïve and Moving Average baselines are far behind.

Business interpretation

- On noisy FX returns, complexity does not guarantee better performance.
- The **true lever is exogenous enrichment** (macro, risk, spreads, liquidity), not further ML sophistication.
- SES/HW/ARIMA provide **robust, explainable, and production-ready** baselines; STACK is a safe ML fallback.

Recommendations

1. **Default models**: HW-SES or ARIMA for production (simplicity, stability).
2. **Monitoring**: rolling RMSE/MAE, drift checks, residual diagnostics.
3. **Refresh**: weekly retraining (daily if volatility rises), walk-forward validation.
4. **Next steps**:
    - Add exogenous signals (macro, SNB/ECB policy, risk sentiment).
    - Test alternative targets (probabilistic quantiles, volatility).
    - Introduce vol-aware loss functions (Huber, asymmetric).

Chapter 9 - Final Synthesis & Business Value

9.1 Executive Summary

Context & Objective

The project aimed to forecast the next-day (t+1) daily log return of EUR/CHF (2015–2025), evaluated on a strict frozen test window (2023–2025). The environment is high-noise with low predictability and regime shifts (e.g., SNB interventions, stress episodes). The goal was not perfect prediction, but **stable, bias-free signals** useful for treasury, hedging, and FP&A decisions.

Data & Protocol

- **Target:** daily log returns.
- **Split:** train 2015–2022, test 2023–2025 (strict, no leakage).
- **Features:** minimal set (lags, rolling mean/volatility).
- **Metrics:** RMSE and MAE (MAPE logged but not interpretable).
- **Robustness checks:** expanding walk-forward, residual ACF, volatility regime analysis.

Modeling Approach

1. **Baselines:** Naïve, Moving Average, SES.
2. **Statistical:** ARIMA, Holt-Winters.
3. **Linear ML:** Ridge/Lasso with TSCV.
4. **Advanced ML:** RF, GBDT, XGB, LGBM.
5. **Ensembles:** Stacking (RF+GBDT→Ridge), Blending.

Purpose: (i) demonstrate that complexity ≠ automatic improvement, (ii) identify a **benchmark baseline** that must be beaten fairly.

Key Results (Test 2023–2025)

- **Winners:** SES / Holt-Winters / ARIMA — RMSE ≈ 0.003321, MAE ≈ 0.002506.
- **Close challengers:** STACK (+0.07%), Linear (+0.62%).
- **Tree/boosters:** RF / GBDT / XGB (+1–3%), LGBM (+7.5%).
- **Naïve baseline:** +46% RMSE, far behind.

The small but consistent gaps show that in a **low-signal setting with no exogenous inputs**, parsimony and calibration (SES/ARIMA) outperform sophistication.

Visual Validation

- **Truth vs Top-3:** predictions remain close to zero, no drift or lag.
- **Residuals:** centered, near-Gaussian, heavier tails on volatile days.
- **Regime analysis:** accuracy falls in high-volatility terciles, consistent with walk-forward backtests.

Business Interpretation

- HW-SES / ARIMA are **production-ready**: fast, stable, explainable, easy to monitor.
- STACK adds **stability** across sub-periods but no material RMSE gain.
- Tree/boosters are close; with richer features (macro, risk, microstructure), they could surpass statistics.

Limitations

- MAPE unusable (denominator ≈ 0).
- Extreme days → residual tails highlight lack of volatility modeling.
- No exogenous drivers (SNB/ECB policy, spreads, risk indices), capping performance.

Recommendations

1. Deploy HW-SES / ARIMA as defaults, STACK as backup.
2. Monitor RMSE/MAE, mean bias, and residual tails; set alert thresholds.
3. Refresh models weekly; rerun walk-forward tests periodically.
4. Next gains: enrich data (macro, policy, sentiment), target vol/direction/quantiles, and test vol-aware losses.

9.2 Key Results & Business Interpretation

Test-Set Ranking (2023–2025)

- **SES / HW-SES / ARIMA:** RMSE ≈ 0.003321, MAE ≈ 0.002506.
  → Interpreted: ~0.33% daily error, i.e., ~33 pips at EUR/CHF ≈ 1.00.
- **STACK (RF+GBDT→Ridge):** RMSE 0.003324 (+0.07%).
- **Linear:** 0.003342 (+0.62%).
- **Trees/boosters:** RF 0.003362 (+1.2%), Blend 0.003370 (+1.5%), XGB 0.003410 (+2.7%), GBDT 0.003427 (+3.2%), LGBM 0.003569 (+7.5%).
- **Naïve:** 0.004842 (+46%).

Why Simple Models Win

1. **Noise dominance:** daily FX returns ≈ white noise. SES/ARIMA avoid overfitting and benefit from implicit regularization.
2. **Minimal features:** without exogenous drivers, non-linear models lack predictive edge.
3. **Bias–variance trade-off:** added variance from boosters outweighs marginal bias reduction.

Business View (Treasury/Hedging)

- **Shrink-to-zero signal:** conservative forecasts fit prudence; small hedge adjustments rather than aggressive bets.
- **Operational stability:** HW-SES / ARIMA are lightweight, explainable, and easy to refit.
- **Regime awareness:** errors rise in high-volatility states; down-weight forecasts when `roll_std_21` spikes.
- **Ensembles as safety net:** STACK adds stability without improving accuracy; useful under regime drift.

Practical Implications

1. **Default choice:** HW-SES or ARIMA; STACK in reserve.
2. **Monitoring:** RMSE/MAE + bias + tails; weekly thresholds.
3. **Future ML edge:** with exogenous signals (ECB/SNB events, rates, risk indices, carry, liquidity), or alternative targets (direction, volatility, quantiles).

TL;DR

Today, **simplicity framed correctly beats complexity**. Tomorrow, performance gains will come from **new information and risk-aligned targets**, not more tuning of algorithms on the same features.

9.3 Limitations & Risks

1. Structural noise ceiling

Daily EUR/CHF returns are almost white noise outside stress episodes. Even the best models (SES/ARIMA) plateau around RMSE ≈ 0.003321.

- **Risk:** false confidence from randomness (local overfit, “lucky split”).
- **Mitigation:** rely on parsimonious models, validate with TimeSeriesSplit + walk-forward, and, if needed, apply Diebold–Mariano tests.

2. MAPE not meaningful

MAPE explodes when the denominator ≈ 0, as in daily returns.

- **Risk:** misleading “huge MAPE” values if shown to non-technical stakeholders.
- **Mitigation:** communicate RMSE/MAE in both decimals and pips; optionally report sMAPE or MAE-in-pips.

3. Volatility regimes (heteroskedasticity)

Errors rise in high-volatility states (see 7.5). SNB/ECB days amplify error.

- **Risk:** sudden performance deterioration during shocks.
- **Mitigation:** vol-aware weighting, quantile/volatility targets, action thresholds tied to σ, degraded execution modes when vol spikes.

4. Leakage & temporal alignment

Although features were shifted and splits strict, leakage is always a risk.

- **Risk:** silent leakage inflates results.
- **Mitigation:** enforce checklists (shifts, scalers fit only on train), run leakage unit tests, and code-review pipelines.

5. Stationarity assumptions

SES/ARIMA assume stable dynamics; regime shifts can break them.

- **Risk:** drift, calibration loss.
- **Mitigation:** regular refits (weekly), residual bias monitoring, and champion–challenger setups.

6. Missing exogenous drivers

No macro/policy/risk variables were included.

- **Risk:** performance ceiling; failure to capture directional moves post-announcement.
- **Mitigation:** enrich with SNB/ECB calendars, rates, spreads, VIX, carry, liquidity; test ARIMAX or boosted models.

7. Target vs operational needs

We forecast log returns, but business often cares about pips or direction.

- **Risk:** gap between statistical accuracy and actionable thresholds.
- **Mitigation:** translate RMSE to pips; consider directional targets (classification) or quantiles (risk bands).

8. Model instability

Boosters/trees can be unstable on weak signals and sensitive to tuning.

- **Risk:** run-to-run variance, “lucky” hyperparameters.
- **Mitigation:** compact grids, temporal CV, ensembles to stabilize, fixed seeds.

9. Outliers & fat tails

Extreme moves dominate RMSE.

- **Risk:** perception of “broken model” on rare shocks.
- **Mitigation:** robust losses (Huber, quantile), diagnostic winsorization, prediction intervals to flag uncertainty.

10. Production risks

Silent misalignments (data lags, holidays, timestamps, library versions).

- **Risk:** wrong inputs, irreproducible results.
- **Mitigation:** timestamp contracts, T-1/T checks, version pinning, logged runs.

In short (exec-ready):

- What’s fixed: high noise, useless MAPE, error spikes in volatility.
- What we control: leakage checks, refits, monitoring of RMSE/MAE/bias/tails, vol-aware thresholds.
- What unlocks progress: exogenous signals + risk-aligned targets (direction, quantiles, volatility).

9.4 Business Impact & Use Cases

Executive summary

Forecasts are **conservative and stable**. The top models (SES/ARIMA) achieve RMSE ≈ 0.0033 (≈0.33%, ~33 pips at EUR/CHF ≈ 1.00). These are not directional “bets” but **weak signals to pace actions**: when to act vs when to stand still, how much to adjust a hedge, and how to frame uncertainty.

Operating principles

1. **No-trade thresholds**
   Act only if the forecast exceeds a volatility-scaled threshold:

$$\text{act if } |\widehat{r_{t+1}}| > \theta \cdot \sigma_{21}, \quad \theta \in [0.3, 0.6]$$

- Example: if $\sigma_{(21)}$ = 35 pips and $|\widehat{r_{t+1}}|$= 12 pips → no action (12 < 0.3×35).

2. **Gradual hedge adjustments**
   Update hedge ratios smoothly, not with flip-flops:
   $$H^* = H_0 + \lambda \cdot \frac{\widehat{r_{t+1}}}{\sigma_{21}}, \quad \lambda \in [0.1, 0.3]$$
3. **Event days caution**
   On SNB/ECB announcements or major macro releases, increase thresholds or reduce execution intensity.
4. **Signal vs cost**
   Only act if expected gain $\propto |\widehat{r_{t+1}}|$ exceeds transaction cost (spread, liquidity).

Use cases

- **Treasury / FX operations**
  - Define a no-trade band in pips.
  - Trigger small hedges only when signals cross thresholds in calm regimes.
  - Avoid liquidity consumption below cost.
- **Hedging & ALM**
  - Adjust hedge coverage in line with volatility (σ↑ → more coverage).
  - Stress test scenarios (±1σ, SNB-like shocks).
- **FP&A**
  - Smooth forecasts into a planning rate (5–10 day median).
  - Translate RMSE → pips to define budget uncertainty bands.
  - Build scenario sets: Base (0), Bull/Bear (±k·RMSE), Stress (event shock).

What not to do

- Overreact to small signals.
- Switch models after one bad day (monitor trends, not single misses).
- Make irreversible strategic decisions (e.g., capex) on a daily forecast.

9.5 — Short Roadmap (3–5 High-Leverage Actions)

1) Add exogenous drivers — without leakage

Rationale. Current models capture the conditional mean but remain “blind” to macro and market catalysts. EUR/CHF reacts to rate differentials, risk sentiment, and policy events.

Implementation.

- Integrate only series available *before* t+1: €STR–SARON spreads, curve slopes (EU/CH 2y–10y), VIX (risk-on/off), S&P 500, bond proxies (term premium), and event dummies (SNB, ECB, CPI, NFP).
- Ensure strict timestamp alignment (shift if needed) to avoid leakage.
  Impact. Likely RMSE gain of 2–5%, but stronger stability on event days and better explainability.

2) Shift from mean to risk-aware targets

Rationale. Forecasting a near-zero mean offers limited business value; decisions depend on direction and uncertainty.
Implementation.

- Directional classification: predict $sign(r_{t+1})$ with calibrated probabilities (Brier score, AUC, calibration curves).
- Quantile forecasts (P10–P50–P90) via pinball loss, giving Treasury clear confidence bands.
- (Optional) Forecast volatility σ_{t+1} (EWMA/GARCH) to anticipate move size.
  Impact. More cost-aware execution (probability- and band-driven), better communication of scenarios (base/central/stress).

3) Make the pipeline volatility-aware

Rationale. Diagnostics show errors rise in high-volatility regimes. Treating all days equally is suboptimal.
Implementation.

- Weight the loss $\frac{1}{\sigma_{21}^2}$ so volatile days don't dominate training.
- Use dynamic thresholds in operations:

$$Act\ if\ |\widehat{r_{t+1}}| > \theta \cdot \sigma_{21}, \quad \theta \in [0.3, 0.6]$$

Impact. Reduces false positives in choppy regimes, saves liquidity, improves operational confidence.

4) Recognize simple regimes (low vs. high vol)

Rationale. Markets are not stationary. One "average" model loses relevance in extremes.

Implementation.

- Split evaluation/decisions into volatility terciles (low vs. high).
- Maintain two parameter sets (or Stack with regime-dependent weights).
- (Optional) Use a simple 2-state Markov model, though volatility buckets are usually sufficient.
  Impact. Improved robustness in tails (P95/P99) and more stable KPIs year-round.

5) Add a cost-aware decision layer

Rationale. A 2–3% RMSE gain is useful, but business value comes from net performance after costs.
Implementation.

- Backtest with realistic spreads/market impact; calibrate θ\theta (action threshold) and λ\lambda (hedge adjustment intensity) to optimize net pips or hedge cost savings.
- Establish explicit no-trade bands to avoid unnecessary execution.
  Impact. Lower over-trading, improved net P&L, stronger buy-in from Treasury and ALM.

3-Month Prioritization

1. Introduce volatility-aware training and dynamic thresholds (quick ROI).
2. Deploy directional/quantile targets (direct decision value).
3. Add exogenous drivers with strict timestamping (no leakage).
4. Test simple regime splits (volatility buckets).
5. Implement cost-aware execution with realistic backtests.

Guiding principle: keep the pipeline light, traceable, and reproducible. Always measure incremental improvements in the same place (RMSE/MAE, P95/P99, net cost) to ensure transparency for decision-makers.

Here's a polished, standardized version of your **Conclusion 9.6 — EUR/CHF Project (2015–2025)**, aligned with the style we've used in earlier chapters:

9.6 — Final Conclusion: EUR/CHF Project (2015–2025)

1) What we built

An end-to-end, reproducible forecasting pipeline for daily EUR/CHF log returns:

- **Parts 1–2**: data collection (Yahoo Finance), cleaning, feature engineering (lags, rolling means/volatilities), strict chronological split (Train 2015–2022, Test 2023–2025), anti-leakage controls.
- **Parts 3–6**: baselines and classical models (Naïve, Moving Average, SES), ARIMA/SARIMA/Holt–Winters, AR-GARCH, linear regressions, Random Forest.
- **Part 7**: advanced ML (RF, GBDT, XGBoost, LightGBM, Stacking, Blend), full diagnostics (residuals, volatility, feature importances), and walk-forward backtests.
- **Part 8**: consolidated evaluation (RMSE/MAE tables, top-3 vs truth plots, residual distributions).
- **Part 9**: business synthesis, limitations & risks, roadmap, and a light MLOps framework.

2) Core result (headline)

- On the fixed **2023–2025 test set**, simple models remain best:

  SES / ARIMA: RMSE ≈ 0.003321, MAE ≈ 0.002506\text{SES / ARIMA: RMSE ≈ 0.003321, MAE ≈ 0.002506}

- Advanced ML does **not** improve RMSE in this low-signal setting:
  - STACK ≈ 0.003324, RF ≈ 0.003362, XGB ≈ 0.003410, GBDT ≈ 0.003427, LGBM ≈ 0.003569.
- **Interpretation**: at horizon t+1t+1, FX returns behave almost like a random walk; mean-based modeling has limited upside, even with non-linear ML.

3) Business reading (Treasury, Hedging, FP&A)

- Typical error: **~0.33% (≈ 33 pips)** around EUR/CHF ≈ 1.00.
- Models do not deliver reliable direction forecasts at t+1t+1; instead they help to:
  - **Time actions** (act vs stand down using volatility-linked thresholds).
  - **Adjust hedge ratios** cautiously, in a volatility-aware manner.
  - **Quantify uncertainty** (convert RMSE to pips to build budget/hedge bands).
- **Value proposition**: avoid over-trading, prioritize execution days, and frame decisions in cost–benefit terms (signal vs spread/impact).

4) Quality & robustness (from diagnostics)

- **Residuals**: centered, near-Gaussian, heavier tails in high-vol regimes — standard in FX.
- **Residual ACF**: within confidence bounds — no leftover autocorrelation.
- **Error by volatility regime**: RMSE increases with σ21\sigma_{21} → supports thresholds or regime-specific models.
- **Walk-forward**: stable performance over time, no systematic drift detected.

5) Known limits

- Target = conditional mean of near-zero returns → inherently weak predictive power.
- **MAPE not meaningful** (denominator ≈ 0).
- Exogenous drivers (rates, VIX, SNB/ECB events) currently omitted.
- Point forecasts only: no quantiles, no probabilistic bands.
- Sensitive to volatility regimes → requires policy overlay at decision time.

6) Production recommendation (champion/challenger)

- **Champion**: HW-SES (or ARIMA) — simple, robust, transparent, easy to refit.
- **Challenger**: STACK (RF + GBDT with Ridge meta-learner) — stable, ready for vol-aware and exogenous extensions.
- **Promotion rule**: promote challenger only if ≥ 3–5% RMSE improvement sustained over ≥ 4 consecutive weeks (with monitoring in place).

7) Roadmap (3-month, high-ROI actions)

1. **Volatility-aware training & thresholds**:

   $$Actif|\widehat{r_{t+1}}| > \theta \cdot \sigma_{21}, \quad \theta \in [0.3,0.6]$$

   Calibrate θ\theta vs execution costs.

2. **Decision-oriented targets**: direction (probabilities), quantiles (P10–P90).
3. **Exogenous drivers**: €STR–SARON spreads, VIX, S&P 500, SNB/ECB and macro flags.
4. **Regime modeling**: two volatility buckets (low/high) with distinct parameters.
5. **Cost-aware execution layer**: backtest with spread/impact to optimize thresholds and hedge intensity.

8) Light MLOps framework

- **KPIs**: 7- & 30-day RMSE/MAE, mean bias, residual P95/P99. Trigger alerts on threshold breaches.
- **Refit cadence**: weekly (daily during heavy event cycles). Version code/data/models; maintain rollback capability.
- **Sanity checks (daily)**: D/D–1 alignment, holiday calendar, plausible ranges, NaN/Inf filters, leakage guardrails.

Bottom line: The project demonstrates that, in a low-signal environment like EUR/CHF daily returns, **robust classical models (SES/ARIMA) match or beat advanced ML**. Value for business lies in disciplined framing (thresholds, vol-aware policies, exogenous enrichment) rather than algorithmic complexity.

At **very short horizons**, a major FX pair is **hard to beat** on the **mean**. Value comes from **disciplined decision-making** (thresholds, costs, volatility) and **reframing the problem around risk** (direction/quantiles/vol, exogenous drivers).
We now have a **clean, reproducible, and traceable** foundation, ready for **light production** and targeted, **high-leverage** iterations.