

Advanced Programming 2025

Forecasting Monthly Changes in France's 10-Year OAT Yield with Macro Indicators (2000–2025)

Final Project Report

Eva Macchitella
eva.macchitella@unil.ch
Student ID: 24432759

January 8, 2026

Abstract

This paper forecasts the one-month change in France's 10-year government bond yield (OAT) using monthly macro indicators: euro area CPI inflation (YoY), French manufacturing output (YoY), the monthly change in French unemployment, the monthly change in the EA 3-month interbank rate over 2000–2025 plus an autoregressive term. The target is Δy_{t+1} and predictors are lagged by $L \in \{1, 3, 6\}$ to respect the information set available at time t . We use leakage-free preprocessing (past-only lags and train-fold transformations), fixed time splits (train: ≤ 2018 ; validation: 2019–2021; test: ≥ 2022), and walk-forward cross-validation for tuning. We compare a Naïve benchmark, AR(1), multivariate OLS, Ridge, and a shallow Random Forest, and we also test simple moving-average smoothing and winsorization as robustness checks. On validation, OLS performs best among the linear specifications (MAE = 0.0965, MSE = 0.0138, R^2 = 0.1135), while the overall lowest validation MAE is obtained by the tuned Random Forest (CV mean MAE = 0.0962). On the held-out 2022–2025 test window, all models have negative out-of-sample R^2 and the Naïve change forecast remains hard to beat, highlighting the intrinsic difficulty of forecasting monthly yield changes in a regime with large shocks. The main contribution is a transparent and reproducible, time-aware forecasting pipeline with strict no-leakage evaluation and clear evidence on the limits of short-horizon predictability for ΔOAT at monthly frequency.

Keywords: Government bond yield forecasting, Macro-financial predictors, Time-series cross-validation, Ordinary least squares regression, France sovereign yield, Walk-forward cross-validation,

1 Introduction

Background and motivation This project aims to forecast monthly changes in France's 10-year government bond yield (OAT) using key macroeconomic indicators from 2000 to 2025. The OAT reflects the country's long-term borrowing costs and is relevant for fiscal sustainability, especially given France's high public debt. Even modest yield increases can result in substantial additional interest-payment burdens. Moreover, long-term rates are central to asset pricing, monetary policy transmission, and risk management. Short-horizon forecasts of monthly yield changes can therefore inform both policymakers and market participants.

Problem statement: We study the one-month horizon prediction of the OAT change, Δy_{t+1} , using a set of timely macroeconomic drivers observed at t : euro-area CPI inflation (YoY), French

unemployment (monthly change), French manufacturing activity (YoY), the euro-area 3-month interbank rate (monthly change), and an autoregressive term in Δy_t . The objective is to design a parsimonious, reproducible pipeline that avoids look-ahead bias and delivers robust out-of-sample performance relative to a naïve $\Delta y_{t+1} = 0$ benchmark.

Objectives and goals (i) Build a transparent data pipeline (monthly alignment, transformations, supervised table). (ii) Specify and compare baseline and regularized models: Naïve, AR(1), OLS, Ridge, and Random Forest—under time-based splits. (iii) Use walk-forward validation to choose lags and smoothing with moving average, and to tune hyperparameters. (iv) Report test-set performance, uncertainty bands, and diagnostics (residual plots, coefficient paths, feature importances). (v) Focus on an interpretable and reproducible code (single-run script, saved figures & tables).

Report organization. The remainder of the report is organized as follows. Section 2 reviews related work. Section 3 describes the data, transformations, model specifications, and the time-aware validation and selection protocol. Implementation details (code structure, reproducibility, version control, and recommended tests) are summarized in Subsection 3.3. Section 4 presents the out-of-sample results and visual evidence (tables and figures). Section 5 interprets the findings, highlights challenges, and discusses limitations. Section 6 concludes and outlines directions for future work. Appendix A provides additional figures and robustness tables, and Appendix B documents the code repository.

2 Literature Review / Related Work

Previous approaches. Prior euro-area work shows that French OAT yields co-move with the common Bund curve while country-specific premia (risk, liquidity, policy news) drive the OAT–Bund spread. This helps explain why short-horizon predictability of *monthly yield changes* is weak and why simple, stable models are often preferred. See, e.g., [2] on euro-area sovereign spread determinants and [1] on fundamentals and policy effects for spreads (including France).

Algorithms / methodology. Our choices follow material covered in prior macroeconometrics courses and this class.

3 Methodology

3.1 Data Description

Source All series are pulled programmatically from the Federal Reserve Economic Data (FRED) via `pandas_datareader` and aligned to month start (MS). Identifiers: OAT 10Y yield IRLTLT01FRM156N, unemployment rate LRHUTTTTFRM156S, manufacturing production YoY FRAPRMNT001GYSAM, all-items HICP CP0000EZCCM086NEST, and EA 3-month interbank rate IR3TIB01EZM156N. The assembled panel is saved to `data/processed/panel.parquet`.

Size The paper analyzes a monthly panel spanning 1999-01–2025-08 (320 months; eight variables: five raw series and three derived), split chronologically into Train \leq 2018-12 ($n = 225$); Validation 2019-01–2021-12 ($n = 36$); Test \geq 2022-01 ($n = 43$). The effective sample size changes slightly when different lags are applied.

Characteristics The dataset consists of monthly macro-financial time series. The French 10-year OAT and the euro-area 3-month interbank rate are expressed in percent, while the response Δy_t is a month-to-month change in percentage points. The unemployment rate is seasonally adjusted and later enters in first differences, manufacturing production is reported as %YoY, and the HICP index is converted to %YoY inflation.

Feature and target construction We forecast the next-month change in the OAT, Δy_{t+1} , rather than its level. Predictors at time t are: (i) HICP inflation %YoY, (ii) the monthly change in the EA 3-month interbank rate Δr_t , (iii) the monthly change in the unemployment rate Δu_t , (iv) manufacturing %YoY, (v) an AR term Δy_t . To avoid look-ahead, all predictors are lagged by $L = 1, 3, 6$ months. We transform HICP to %YoY to remove seasonality, and we difference unemployment and the policy-rate proxy. We use EA 3-month interbank and manufacturing production rather than a refinancing rate or quaterly GDP to keep a true monthly frequency and avoid averaging.

Stationarity Augmented Dickey–Fuller tests indicate that unemployment and the policy-rate proxy are $I(1)$ in levels; they enter as first differences Δu_{t-L} , Δr_{t-L} . Variables already in %YoY (HICP, manufacturing) are treated as stationary and simply lagged. The target Δy_{t+1} is stationary by construction because it is a first difference of a rate.

Data quality Missingness comes from lag creation, rows with incomplete lags are dropped (`dropna()`, no imputation).

COVID-19 (2020–2021). We keep the COVID-19 period in the sample to preserve the real-time chronology. As a robustness check, we tested train-only winsorization to reduce the influence of extreme observations.

3.2 Approach

Algorithms (and why)

We analyze five models that increase in flexibility while keeping a transparent core:

- **Naïve** ($\widehat{\Delta y}_{t+1} = 0$): random-walk-in-levels baseline.
- **AR(1) in changes**: $\Delta y_{t+1} = c + \phi \Delta y_t + \varepsilon_{t+1}$; tests short-run persistence.
- **OLS (linear, interpretable)**: adds a small macro block to Δy_t to test incremental predictive content.
- **Ridge (L2-regularized OLS)**: same linear form but shrinks coefficients to mitigate multicollinearity and variance.
- **Random Forest (shallow)**: non-linear model with feature importances for robustness checks.

Preprocessing

All series are harmonized to monthly frequency (month-start). Economic transforms follow Section 3 (Data Description): HICP \rightarrow %YoY; unemployment and EA 3M interbank enter as first differences; manufacturing is %YoY. We form the supervised panel by (i) lagging all macro predictors by $L = 1, 3, 6$ months to respect the information set at t , (ii) including the AR term Δy_t , and (iii) defining the target as next-month change Δy_{t+1} . We use a strictly time-ordered split: *Train* \leq 2018-12, *Validation* 2019-01–2021-12, *Test* \geq 2022-01. As a robustness check, we

optionally winsorize *features only* at the 1st/99th percentiles using *train-only* cutoffs and apply them unchanged to validation/test. In the main specification, winsorization is turned off. No imputation; rows with incomplete lags are dropped. Standardization is applied *only* inside the Ridge pipeline (fit on Train, applied to Val/Test). AR(1), OLS, and RF use raw features.

Model specification

Let

$$x_t = (\text{HICP YoY}_{t-L}, \Delta r_{t-L}, \Delta u_{t-L}, \text{Manuf YoY}_{t-L}, \Delta y_t), \quad \text{target: } \Delta y_{t+1}.$$

OLS: $\Delta y_{t+1} = \beta_0 + \beta^\top x_t + \varepsilon_{t+1}$, estimated on Train; coefficients are interpreted on standardized features (for comparability) when plotted.

Ridge: minimize

$$\frac{1}{n} \sum_{t \in \mathcal{T}} (\Delta y_{t+1} - \beta_0 - \beta^\top x_t)^2 + \lambda \|\beta\|_2^2,$$

with $\lambda \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$; features are standardized inside the pipeline.

Random Forest: tuned over a small grid $\{n_est \in \{300, 600\}, \max_depth \in \{3, 5, 7\}, \min_leaf \in \{3, 5, 8\}, \max_features = \text{"sqrt"}\}$.

Validation and selection

Lag length, optional moving-average smoothing, and winsorization are treated as hyperparameters: $L \in \{1, 3, 6\}$, MA window $k \in \{\text{none}, 3\}$, winsorization $\in \{\text{off}, \text{train-only at } [1\%, 99\%]\}$. We select these by lowest Validation MAE under a walk-forward scheme (2019–2021), then refit the chosen setting on Train+Val before the single Test evaluation.

Evaluation metrics and diagnostics

Primary metrics are **MAE**, **MSE**, and R^2 on Validation and Test; the model ranking uses Validation MAE. We complement these with (i) test-period overlays (actual vs. predictions for all models), (ii) calibration scatter plots with identity and fitted lines, (iii) residual diagnostics (line, ACF, QQ) for the validation winner, (iv) standardized OLS coefficients and RF feature importances for interpretability, and (v) a simple homoskedastic 95% band for OLS using Train+Val residual variance.

3.3 Implementation

Stack (minimal). The project is implemented in Python (3.11) using `pandas/numpy` for data handling, `pandas_datareader` for FRED access, `statsmodels` for time-series diagnostics, `scikit-learn` for model estimation and cross-validation, and `matplotlib` for figures.

Code organization (one-pass pipeline). The codebase is structured to mirror the forecasting workflow: `src/data_loader.py` builds the monthly panel and transformations, `src/models.py` fits AR(1), OLS, and performs walk-forward tuning for Ridge and Random Forest, `src/graphs_and_metrics.py` computes metrics and produces plots, and `main.py` orchestrates the end-to-end run and exports all artifacts.

Performance and (optional) parallelism. The pipeline is lightweight at monthly frequency, so end-to-end runs are fast enough to support repeated experiments. Random Forest training is the only potentially heavy step, and it can exploit parallelism across trees via `n_jobs` in `scikit-learn` (no dedicated scalability benchmark is reported).

Reproducibility and maintenance. Dependencies are pinned in `environment.yml` and a single command reproduces the full workflow:

```
1 python main.py
```

All outputs are written to `results/figures/` and `results/tables/`, ensuring the report does not rely on manual editing.

Version control and updates. Development is tracked with git commits that separate data ingestion, modeling, and visualization changes, and the report can be tied to a tagged release commit. When a new month becomes available, the pipeline can be rerun to append data, rebuild the supervised table, and refit models using the same selection rule. Walk-forward validation provides a natural monitoring tool: deteriorating rolling performance flags regime change and motivates re-specification.

Minimal testing (recommended). To guard against regressions, the most valuable unit tests target leakage-sensitive components: (i) lag creation (no future values in features), (ii) split boundaries (Train/Validation/Test cutoffs), and (iii) deterministic artifact generation (fixed seeds for Random Forest).

4 Results

Experimental setup (compact).

Table 1: Experimental setup and tuned components

Item	Setting
Target	Next-month change in the 10Y OAT yield, Δy_{t+1} (pp).
Predictors	CPI YoY, Δ EA 3M interbank, Δ unemployment, manufacturing YoY (all lagged by $L = 3$), and the AR term Δy_t at time t .
Sample split	Train \leq 2018-12 ($n = 225$), Validation 2019-01–2021-12 ($n = 36$), Test \geq 2022-01 ($n = 43$).
Smoothing	MA(3) applied to Δ unemployment, Δ interbank, and manufacturing; CPI YoY and Δy_t unsmoothed.
Models	Naïve, AR(1), OLS; Ridge and Random Forest tuned by walk-forward CV.
Tuning rule	Hyperparameters selected by lowest validation MAE using expanding-window CV; the Test set is evaluated once after re-fitting on Train+Validation.
Winsorization	Disabled in the main run; treated as an optional robustness check (train-only cutoffs) in separate experiments.

Evaluation metrics. We report MAE (primary), MSE, and R^2 on Validation (for model choice) and on the held-out Test set (final performance).

4.1 Performance Evaluation

Selection rule and reported models. Models are selected by the pre-declared rule of lowest validation MAE (2019–2021). Under this criterion, the tuned Random Forest is the validation winner. Because interpretability is also an objective, we report a transparent linear specification (OLS) alongside two benchmarks (Naïve and AR(1)).

Benchmarks (Naïve and AR(1)). The Naïve forecast $\widehat{\Delta y}_{t+1} = 0$ yields Validation MSE = 0.0157, MAE = 0.1067, and $R^2 = -0.0057$, and Test MSE = 0.0561, MAE = 0.1705, and $R^2 = -0.1028$. The AR(1) in changes, $\Delta y_{t+1} = c + \phi \Delta y_t + \varepsilon_{t+1}$, improves on validation (MSE = 0.0144, MAE = 0.0979, $R^2 = 0.0784$), but does not generalize to the test window (MSE = 0.0571, MAE = 0.1755, $R^2 = -0.1229$).

OLS (main interpretable specification). OLS uses five predictors: CPI YoY, Δ EA 3M interbank, Δ unemployment, manufacturing YoY (all lagged by $L = 3$), and the autoregressive term Δy_t . On validation, OLS achieves MSE = 0.0138, MAE = 0.0965, and $R^2 = 0.1135$, which improves on both benchmarks. On the test window, OLS deteriorates (MSE = 0.0674, MAE = 0.2063, $R^2 = -0.3251$), underperforming the Naïve baseline.

Regularization and nonlinearity (Ridge and Random Forest). Ridge is tuned by expanding-window (walk-forward) cross-validation on Train+Validation and attains mean CV MAE = 0.1002 and mean CV MSE = 0.0164. After refitting on Train+Validation, Ridge reduces test error relative to OLS (Test MSE = 0.0636 vs. 0.0674), but it remains worse than Naïve (Test MSE = 0.0561). The tuned Random Forest achieves mean CV MAE = 0.0962 and mean CV MSE = 0.0166, and on the test window it yields MAE = 0.1895 and MSE = 0.0615. Thus RF improves on OLS and Ridge on Test, but it still does not beat the Naïve benchmark. Impurity-based feature importance is reported for context and typically ranks the autoregressive term Δy_t among the most informative predictors.

Main conclusion across regimes. Validation (2019–2021) shows modest predictability, with AR(1) and OLS outperforming the Naïve forecast. In contrast, the post-2022 test window is harder: all models have negative R^2 , and the Naïve baseline is the most difficult to beat. This regime dependence is consistent with the known difficulty of forecasting monthly yield changes and with the structural shift after 2021 (inflation surge and rapid tightening).

Evaluation protocol (no leakage). Hyperparameters are tuned using expanding-window validation, where each validation slice starts strictly after its training cutoff. After tuning, models are refit once on Train+Validation and evaluated a single time on the held-out Test window (2022–2025).

Uncertainty visualization. Figure 7 visualizes forecast uncertainty for the OLS specification on the Test period by adding a symmetric error band around the point forecast. Let $\widehat{\sigma}$ be the Train+Validation RMSE,

$$\widehat{\sigma} = \sqrt{\frac{1}{N} \sum_{t \in \text{Train} \cup \text{Val}} (y_t - \widehat{y}_t)^2},$$

and we plot $\widehat{y}_t \pm 1.96 \widehat{\sigma}$ on the Test window. The band is a descriptive “typical error” envelope (homoskedastic normal approximation) rather than a formal predictive interval.

Design choices (robustness). Winsorization is not used in the main run and did not materially improve out-of-sample accuracy in robustness experiments. The main specification uses a macro lag $L = 3$ and MA(3) smoothing for selected noisy predictors, while CPI YoY and Δy_t are left unsmoothed. Alternative lags and preprocessing variants were tested, but they did not improve the final test performance.

For transparency, we provide a complete log of robustness checks in `robustnesscheck/oat_metrics_.`csv, which reports MSE/MAE (and R^2 where applicable) across alternative specifications (e.g., lag $L \in \{1, 3, 6\}$ and moving-average smoothing $MA \in \{1, 3\}$), in addition to the final model results discussed in the main text.

4.2 Visualizations

Figure overview (compact).

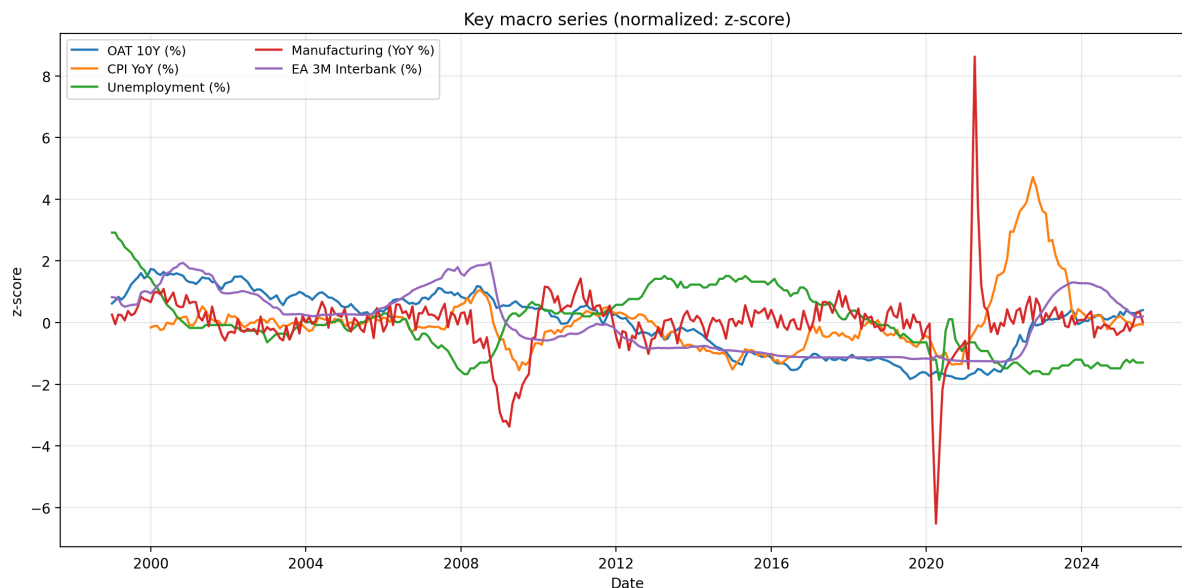


Figure 1: Macro context (normalized): France OAT 10Y, Euro area CPI YoY, France unemployment, France manufacturing YoY, and EA 3M interbank after z -scoring. Purpose: quick visual check of regimes and co-movements on a comparable scale.

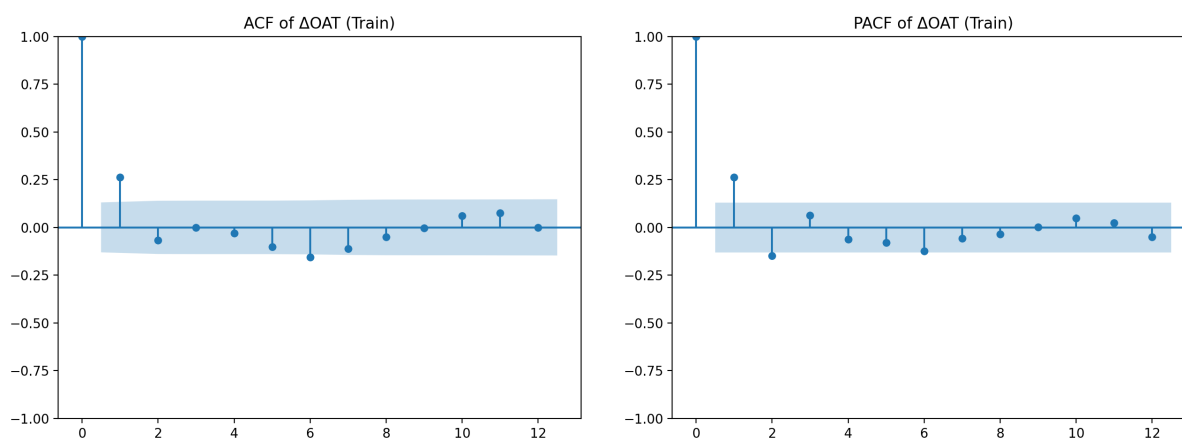


Figure 2: Train-time structure of ΔOAT : ACF(12) (left) and PACF(12) (right). The ACF/-PACF summarize serial dependence in monthly OAT changes: only very small spikes appear at short lags, implying weak autocorrelation. This supports using at most a simple AR term and explains why one-step-ahead predictability is limited

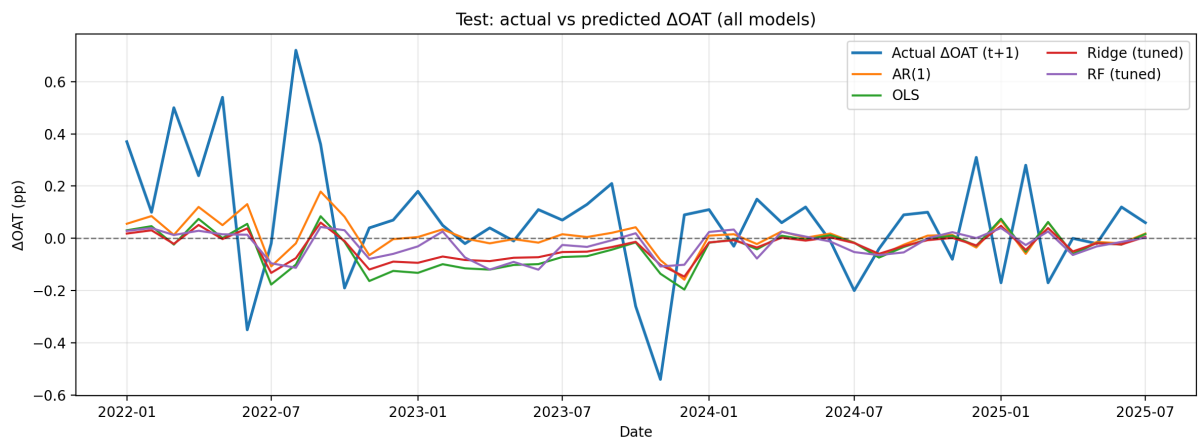


Figure 3: This overlay compares each model's test forecasts to the realized ΔOAT and shows whether models track turning points and volatility. The forecasts stay close to zero and miss large shocks, which visually explains the negative test R^2 .

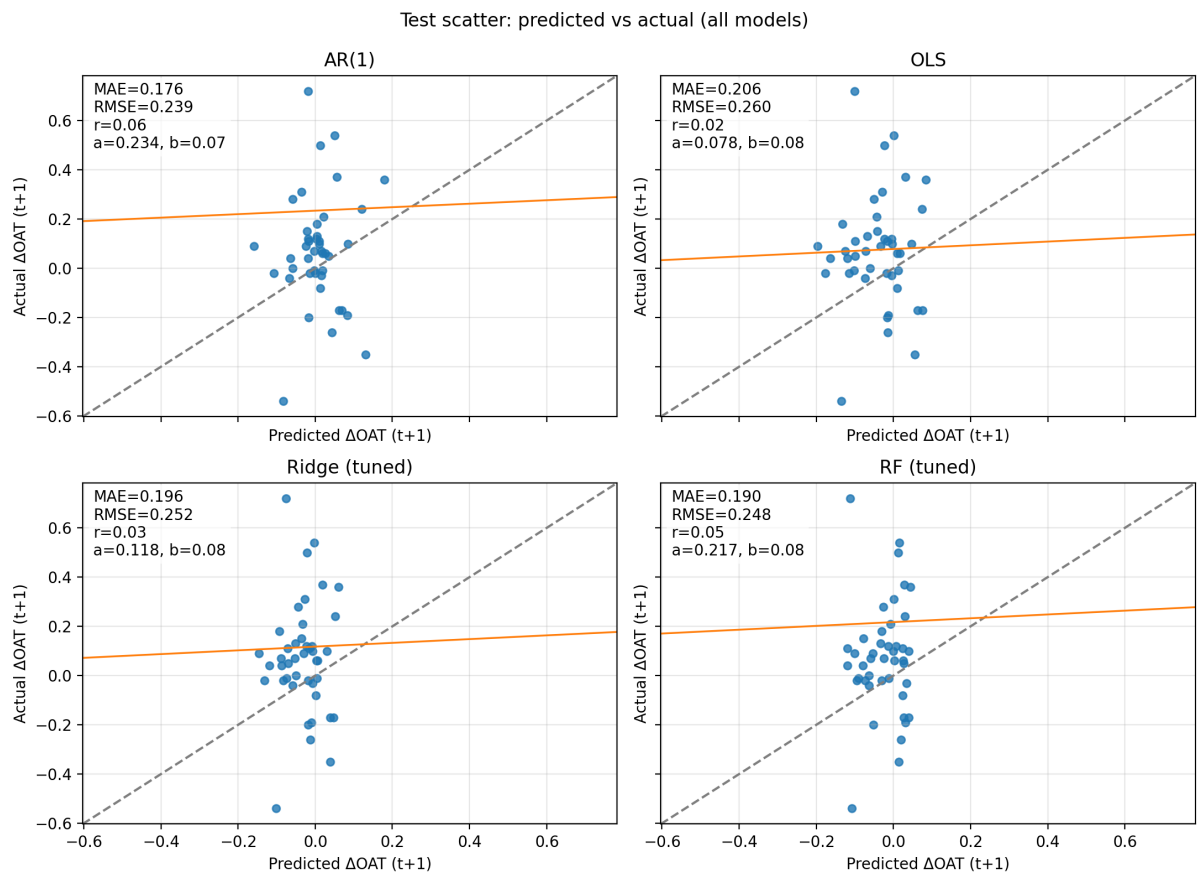


Figure 4: Each panel plots predicted versus actual ΔOAT with the identity line $y = x$, where perfect forecasts would lie on the line. The calibration line $y = a + bx$ is nearly flat and correlation is low, indicating under-dispersed predictions and weak association with outcomes.

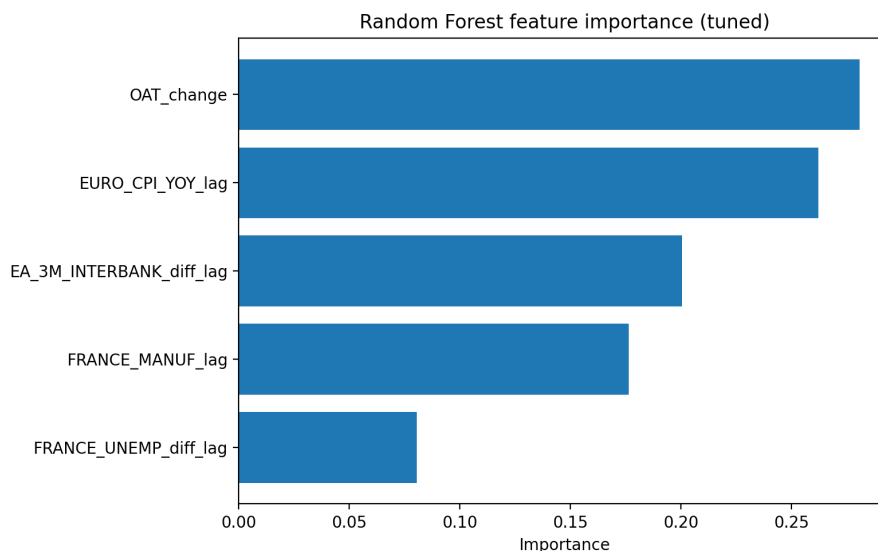


Figure 5: This bar chart ranks predictors by their impurity-based importance in the tuned Random Forest, measuring how much splits on each feature reduce error inside the trees. The autoregressive term Δy_t is most important, while macro predictors provide only secondary signal at the one-month horizon.

5 Discussion

- What worked well? The main strength is leakage control: predictors are lagged so that information at month t only uses data available by the end of t , and the target is ΔOAT_{t+1} . After experimenting with a single validation slice, I implemented walk-forward validation: training always ends before the validation window begins, the validation window rolls forward in time, and hyper-parameters are selected by their average out-of-sample error across these rolling slices. This design substantially reduces the risk of inadvertently over-fitting to one favorable period and mirrors how a forecaster would operate in real time. Model choice and tuning rely on time-based splits and walk-forward validation, which reduces overfitting to one period and mimics real forecasting conditions. Transformations are transparent and reproducible (HICP to YoY, differencing for interbank and unemployment), and residual diagnostics (time plot, ACF, QQ, Ljung-Box) provide basic checks that little predictable structure remains at a monthly horizon.
- What were the challenges? The fundamental difficulty is the target itself, forecasting monthly yield *changes* is intrinsically hard: most movements are driven by shocks and news not captured by low-frequency macro variables. The post-2022 regime (inflation surge and rapid tightening) is structurally different from earlier years, so relationships estimated on 2000–2018 do not generalize well to 2022–2025. This helps explain why validation improvements for AR(1) and OLS (small positive R^2) do not translate to the test set, where all models exhibit negative out-of-sample R^2 and the Naïve forecast is difficult to beat.
- Limitations of your approach The feature set is intentionally small and monthly, excluding term-structure factors, market-based expectations, risk measures, and surprise variables that often explain short-horizon rate moves. Linear models assume constant parameters across long samples, while the Random Forest is kept shallow to avoid overfitting in a small dataset. The analysis uses revised macro series rather than real-time vintages, so data revisions are ignored.

6 Conclusion and Future Work

6.1 Summary

This project implements a reproducible, leakage-safe forecasting pipeline for monthly ΔOAT with a deployment-style evaluation. Series are aligned to monthly frequency and transformed transparently (YoY inflation, first differences, lagged predictors) to respect information timing. We compare Naïve and AR(1) baselines to OLS, Ridge, and a shallow Random Forest, select hyperparameters via walk-forward validation, and refit on Train+Validation before a single held-out Test evaluation. Validation diagnostics suggest little remaining serial correlation, and forecast plots include heuristic uncertainty bands from Train+Validation residual dispersion. Empirically, models show only modest skill on validation and fail to beat the Naïve benchmark on the 2022–2025 test window, yielding negative out-of-sample R^2 . Overall, the contribution is a clean experimental design and an honest assessment of limited predictability for one-month yield changes, especially across post-2021 regime shifts.

6.2 Future Directions

Suggest potential improvements or extensions:

- **Richer predictors and adaptive models.** A first improvement is to enrich the information set with term-structure factors such as level, slope, and curvature. It would also be useful to include expectation-based variables and surprise measures constructed from realized releases relative to forecasts. Finally, adding risk proxies such as volatility and credit spreads could better capture the shocks that move long rates at short horizons. To handle regime changes, the model could be estimated in an adaptive way, for example with rolling windows, time-varying parameters, or strongly regularized boosting methods.
- **More diagnostic evaluation.** A second improvement is to report performance in a rolling out-of-sample way to show when the model helps and when it fails. An ablation study can then quantify the marginal contribution of each predictor by removing them one at a time. In addition, formal forecast comparison tests such as Diebold–Mariano can be used to assess whether differences relative to the Naïve benchmark are statistically meaningful.
- **Practical use.** Even when point forecasts are weak, the pipeline delivers realistic error scales and uncertainty bands for scenario analysis and monitoring. As new data arrive, re-running the same walk-forward evaluation can flag performance deterioration and motivate feature or model updates.

A Additional Figures

Table 2: Model selection rule and winning specification

Variant (OLS only)	Val. MAE	Val. MSE	Val. R^2	Test MSE
L=3, MA=None, Winsor=Off	0.097	0.014	0.077	0.061
L=3, MA=3, Winsor=Off	0.097	0.014	0.113	0.067
L=3, MA=None, Winsor=1–99	0.101	0.015	0.060	0.057
Benchmarks (for reference)				
Naïve ($\Delta y_{t+1}=0$)	0.107	0.016	−0.006	0.056
AR(1) on ΔOAT	0.098	0.014	0.078	0.057

Selection metric: lowest Validation MAE on 2019–2021; R^2 is reported for context. Test (2022–2025) reported once, no re-tuning.

Table 3: Diagnostics for design choices (OLS only)

(a) Lag choice (MA=None, Winsor=Off)

Lag L	Val. MAE	Val. MSE	Test MSE
1	0.099	0.015	0.061
3	0.097	0.014	0.061
6	0.103	0.015	0.057

(b) Effect of smoothing at $L = 3$ (MA on ΔUnemp , $\Delta\text{Interbank}$, Manuf)

MA window k	Val. MAE	Val. MSE	Test MSE
None	0.097	0.014	0.061
3	0.097	0.014	0.067

(c) Effect of winsorization at $L = 3$ (MA=None)

Winsor	Val. MAE	Val. MSE	Test MSE
Off	0.097	0.014	0.061
1–99%	0.101	0.015	0.057

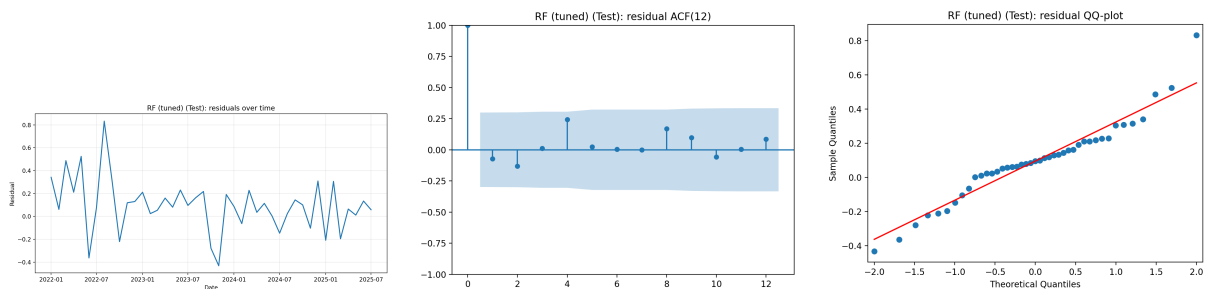


Figure 6: Residual diagnostics for the validation winner on test: residuals over time (left), residual ACF(12) (middle), QQ-plot (right). Checks bias, leftover dependence, and tail behavior.

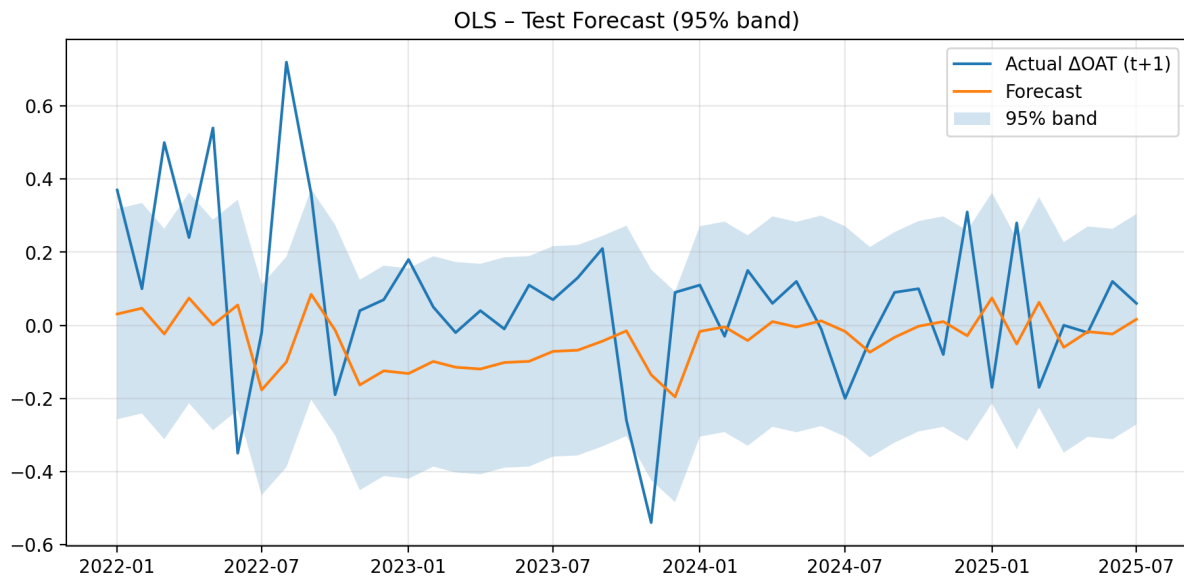


Figure 7: OLS test forecast with $\pm 1.96 \hat{\sigma}$ band (homoskedastic normal approximation; $\hat{\sigma}$ from Train+Validation residuals).

AI use I used ChatGPT and DeepSeek as support tools for code development (debugging and structuring the pipeline), report drafting (clarifying explanations and improving phrasing), and preliminary literature search (identifying relevant references).

B Code Repository

GitHub Repository: <https://github.com/yourusername/project-repo>

Repository structure

The repository follows a small, modular layout:

- **main.py:** entry point that runs the full pipeline end-to-end (download/build panel → feature table → train/tune → evaluate → export figures/tables).
- **src/data_loader.py:** data acquisition from FRED, monthly alignment, variable transformations (YoY inflation, first differences), lag creation, and export of the processed panel (data/processed/panel.parquet).
- **src/models.py:** model estimation (Naïve, AR(1), OLS) and walk-forward tuning/refit for Ridge and Random Forest.
- **src/graphs_and_metrics.py:** metrics (MAE/MSE/ R^2) and all plotting utilities used in the report.
- **data/:** raw download and processed dataset (data/processed/panel.parquet).
- **results/figures/** and **results/tables/:** exported PNG figures and tables used in the report.
- **environment.yml:** pinned Conda environment for reproducibility.
- **robustnesscheck/:** previous metrics downloads from robustness checks (Winsor on and off, Lag and MA testing)

Installation instructions

Create and activate the Conda environment:

```
1 conda env create -f environment.yml
2 conda activate oat-forecast
```

How to reproduce results

Running the full experiment (including data download, preprocessing, model estimation, evaluation, and artifact export) is done with:

```
1 python main.py
```

The script writes all outputs to **results/figures/** and **results/tables/**, and raw data to **data/raw/**. The LaTeX report loads figures directly from **results/figures/**. Re-running **main.py** with the same pinned environment and fixed random seeds reproduces the same tables and figures.

References

- [1] António Afonso, Michalis P. Arghyrou, and Alexandros Kantonikas. *The determinants of sovereign bond yield spreads in the EMU*. ECB Working Paper 1781. ECB Working Paper No. 1781. European Central Bank, 2015. URL: <https://www.ecb.europa.eu/pub/research/working-papers/2015/html/index.en.html>.
- [2] Roberto A. De Santis. *The euro area sovereign debt crisis: safe haven, credit rating agencies and the spread of the fever from Greece, Ireland and Portugal*. ECB Working Paper 1419. ECB Working Paper No. 1419. European Central Bank, 2012. URL: <https://www.ecb.europa.eu/pub/research/working-papers/html/index.en.html>.