Copenhagen Business School

Final Project

Adaptive Regime-Switching Models for EUR/USD Forecasting

A comparative study of baseline, per-regime, and LSTM models using engineered features and historical FX data

Course	:	Machine Learning and Deep Learning CDSCO2004U
Group Member	:	Lou-Felix Thibodeau-Comtois (181550)

Number of pages	:	15 Pages
Number of characters	:	21 839
Submission date	:	2025-08-12
Github Link	:	github.com/loth25ab/FINAL_PROJECT_MLDL

Table of Contents

Abstract

This project studies short-horizon EUR/USD forecasting under changing market conditions. We frame the problem as next-day direction classification y_{t+1} and test whether conditioning on latent market regimes improves out-of-sample performance versus a single global model. The two research questions are: (1) do regime-specific models outperform a global classifier statistically (F1/AUC on rolling out-of-fold data) and economically (Sharpe/CAGR/MDD in $t \rightarrow t+1$ backtests with costs)? and (2) does an LSTM trained on pruned, standardized sequences add incremental value over tree/linear baselines, or does FX noise neutralize that advantage?

Key concepts include regime switching, non-IID time series, leakage-safe evaluation (rolling splits, OOF predictions), and after-cost backtesting. The dataset comprises daily EUR/USD with engineered technical features (returns, volatility, momentum) and light cross-asset context. Methods and tools span HMM-based regime detection (with persistence smoothing), Logistic Regression, XGBoost, and an LSTM baseline, implemented with scikit-learn, hmmlearn, XGBoost, and standard Python data tooling; correlation pruning and optional PCA are used to stabilize inputs.

Results show modest but consistent OOF gains for regime-specific models over a global baseline (e.g., F1 rising from ~0.83 to ~0.85–0.86 with AUC around 0.92–0.93) and more stable behavior across market phases; however, simple $t \rightarrow t+1$ backtests translate those probability gains into only limited economic improvement without careful thresholding and cost control. The LSTM does not deliver incremental ranking power (AUC near random in our setting); any apparent trading benefit comes from its selectivity rather than genuine predictive skill.

We conclude that explicit regime alignment helps, primarily by improving stability rather than delivering dramatic accuracy jumps. Recommendations are to deploy a switching stack (best model per regime) with regime-specific thresholds and sizing, calibrate probabilities, and strengthen regime labeling (e.g., duration-aware HMMs, slow exogenous drivers). Sequence models should be treated as filters unless they demonstrate durable, after-cost gains under the same leakage-safe evaluation.

Keywords: Foreign Exchange (FX), EUR/USD, Market Regime Detection, Time Series Classification, Machine Learning, Deep Learning, Backtesting, LSTM, XGBoost, Logistic Regression.

1. Introduction

The EUR/USD currency pair is the most traded in global foreign exchange (FX) markets, known for high liquidity and rapid reaction to economic, political, and sentiment-driven events. Forecasting its short-term direction is challenging due to the market's non-stationary nature, where relationships between features and outcomes shift over time.

This project evaluates whether training separate models for distinct market conditions can outperform a single global model. The workflow spans data preparation, technical feature engineering, regime detection, and predictive modeling using Logistic Regression, XGBoost, and LSTM.

Models were assessed with rolling time-series cross-validation and validated through backtests simulating a simple long/short strategy. Results provide insights into the benefits and trade-offs of regime-specific approaches for FX prediction, as well as the practical limitations of applying deep learning to noisy financial time series.

2. Motivation and Research Questions

The core motivation is practical: we want signal that holds up once strict time ordering is enforced. In-sample gains are easy; what matters is out-of-fold performance and whether those probabilities translate into reasonable trading behaviour in a simple, rules-based backtest. Our backtests do not include transaction costs, so the equity curves represent an upper bound on what could be implemented.

There is also a learning goal. If market behaviour depends on state, a single model will average away patterns that only exist in certain conditions. Splitting by regime tests that idea directly. Comparing classical models with a sequence model helps separate "more complex" from "actually better" on this dataset. The aim is to be clear about when specialization helps, when it doesn't, and how much performance depends on choices that are robust rather than lucky.

Research Questions:

1. Do regime-specific models for EUR/USD outperform a single global model—both statistically (F1/AUC on rolling OOF) and economically (Sharpe/CAGR/MDD in *t+1* backtests with costs)?
2. Does an LSTM trained on pruned, standardised sequences add incremental performance over tree/linear baselines, or does FX noise neutralise its advantage?

3. Related Work

The idea that markets flip between latent "states" goes back to Hamilton's Markov-switching models and Kim's extensions, which showed that regime shifts explain dynamics better than one-size-fits-all specifications. In asset allocation, surveys by Ang & Timmermann and empirical work by Guidolin & Timmermann find that returns, volatility, and correlations move with regimes, and that conditioning on state can beat single-regime benchmarks.

In FX, Neely and co-authors document that the profitability of technical trading rules is episodic and often tied to policy and intervention. Basically, predictability comes and goes with the regime. On the ML side, Fischer & Krauss report LSTM gains on equities under strict walk-forward tests, but follow-ups in FX show mixed results: deep nets are sensitive to noise, class imbalance, and non-stationarity. Taken together, the literature points to a sensible recipe: detect regimes (k-means/GMM or, when clusters blur, HMMs), build cross-asset technical features, and compare a global baseline with per-regime learners—using sequence models only when the signal justifies the added complexity.

4. Conceptual Framework

This project treats short-horizon EUR/USD prediction as a state-dependent classification problem. Market behaviour is modeled as switching among a small set of latent regimes (e.g., trendy/low-vol vs. choppy/high-vol). The economic idea is simple: the same feature can be informative in one state and useless in another, so a single global mapping is too blunt.

Conceptually, each day t is described by a vector of market descriptors x_t: returns, volatility, momentum (RSI/MACD/ATR), moving-average signals, and cross-asset context (DXY, VIX, 10Y yield, oil, gold) summarized through comparable technical transforms. A regime label $z_t \in \{1, \dots, K\}$ captures the market state as an abstract category; it is not a target in itself but a conditioning variable for prediction.

The prediction task is binary: will EUR/USD go up tomorrow? Formally, the label is $y_{t+1} = 1[r_{t+1} > 0]$, and we learn either a single mapping $f: x_t \mapsto \Pr(\mathrm{y}_{\mathrm{t+1}} = 1)$ or a family of regime-specific mappings $\{f_k\}$ applied when $z_t = k$. Model families are chosen for complementary inductive biases: a linear decision surface (logistic regression) for interpretability and stability; a non-linear ensemble (gradient-boosted trees) for interactions and thresholds; and a sequence model (LSTM) to represent temporal dependencies beyond fixed lags.

Success is defined along two conceptual axes. Statistically, a useful model separates classes out-of-sample (higher F1/AUC). Economically, it must translate probabilistic skill into risk-adjusted returns when converted to trades (higher Sharpe, acceptable drawdowns). The core hypothesis is that conditioning on z_t preserves signal that a global model averages away; sequence models may add value only if the data carry stable temporal structure relative to noise

5. Methodology (Notebook-by-Notebook)

5.1 Notebook 01 — Data Prep & Quick EDA

This first notebook assembles the dataset you'll use everywhere else. It pulls daily OHLCV data for EUR/USD (target) plus the context assets (DXY, VIX, WTI, 10-year yield, Gold, equity benchmark), aligns them on a common calendar, forward-fills only the non-target series, and drops any rows where the target is missing. It then creates the prediction label (next-day direction of EUR/USD from one-day returns) ensuring it uses information available at time *t* only.

Some light EDA at the end of the first notebook help us visualize the dynamics at play:

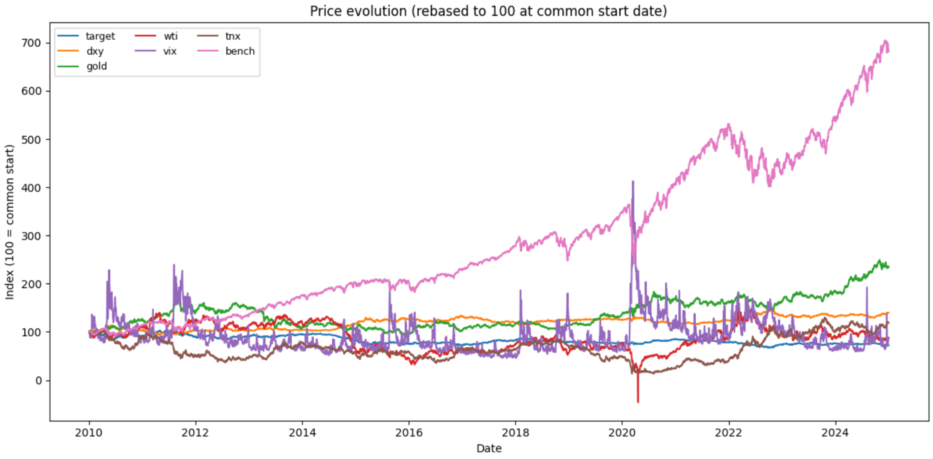

Figure 1: Graph of Daily Closing Prices

The benchmark trends up with sharp COVID swings; VIX spikes episodically; yields fall then rebound; EUR/USD ranges. Could be evidence of multiple regimes.

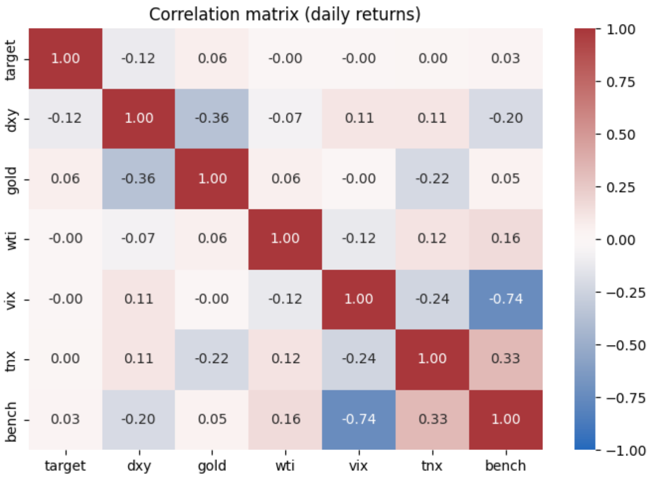

Figure 2: Cross-Assets Correlation Matrix

Links are modest overall. Benchmark vs VIX is strongly negative (−0.74); DXY vs Gold is negative (−0.36); DXY vs EUR/USD is only mildly negative (~−0.12). Cross-asset features add context without being duplicates, and weak linear ties to the target argue for non-linear or regime-aware models.

5.2 Notebook 02 — Features & Regimes

This notebook turns raw prices into signals and gives each day a market "mood." First, it engineers a rich feature set for EUR/USD and the context assets: multi-horizon returns, rolling vol, moving-average/RSI/MACD/ATR cues, Bollinger position, plus rolling cross-asset correlations. It trims the warm-up period so all rolling indicators are fully formed, then saves the clean feature matrix.

Next, it detects regimes using HMM by clustering a regime-oriented slice of the features (returns/vol/momentum). A light PCA is used only to stabilize clustering; the supervised models still see the original features. The result is a timeline of regime IDs merged back into the features.

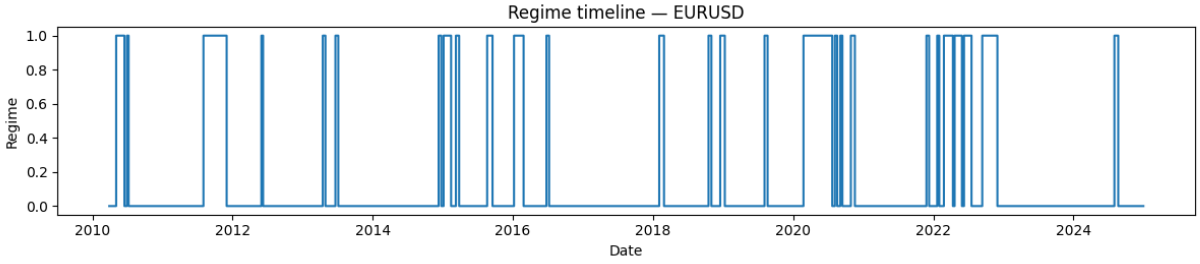

Figure 3: Regime Timeline

Two clusters are detected but the low silhouette score of 0.28 suggest that the clusters are difficult to differentiate. Finally, per-regime pruning is ran: within each regime, it ranks features by predictive importance and keeps only the strongest signals.

5.3 Notebook 03 — Baseline Models

This notebook builds the global, regime-agnostic benchmark we'll compare everything to. It merges the engineered features with the next-day EUR/USD direction label, keeps only numeric columns, and sets up a rolling TimeSeriesSplit so each fold trains on the past and tests on the future.

Two straightforward classifiers are trained:

- Logistic Regression (linear reference, with standardization).
- XGBoost (non-linear trees to capture interactions).

For each fold it stores out-of-fold (OOF) probabilities, then aggregates Accuracy, Precision, Recall, F1, and AUC. The model with the best F1 (tie-break by AUC) is refit on the entire dataset to produce a deployable baseline.

5.4 Notebook 04 — Per-Regime Models

Here we stop treating the market as one blob. Using the regime labels from NB-02 and the pruned feature sets per regime, we:

1. split the data by regime_id;
2. for each slice, train the same two families as before (LogReg and XGBoost) with rolling TimeSeriesSplit;
3. pick the winner per regime by F1 (tie-break AUC) and refit it on the full slice;
4. stitch the out-of-fold probabilities back into one timeline so we can compare head-to-head with the global baseline.

Guardrails stay tight: no peeking across time, scalers fit on train folds only, and we skip regimes with too few rows to be stable.

5.5 Notebook 05 — RNN Model (LSTM Sequence)

This notebook switches from single-day snapshots to short sequences. From the engineered feature table we form rolling windows (for example, the past 60 days up to time t) to predict the EUR/USD direction at *t+1*. Features are standardized with scalers fit only on training folds, and the warm-up period is trimmed so every window is complete. Training uses the same rolling TimeSeriesSplit as earlier notebooks so validation always sits in the future. A compact stacked LSTM with dropout and mild L2 is fit in each fold with class weights, early stopping, and a learning-rate schedule. Out-of-fold probabilities are saved by date for fair backtesting, and the model is finally refit on the full history for completeness. The aim is to test whether sequence information adds robust signal beyond the global and per-regime baselines under identical leakage-safe evaluation.

5.6 Notebook 06 — Backtest & Evaluation

This final notebook turns model probabilities into a tradable path and measures whether any of the approaches earned their keep. It loads the out-of-fold predictions from the baseline (NB-03), the stitched per-regime models (NB-04), and the LSTM (NB-05), aligns them to the EUR/USD return series from the aligned table, and enforces $t \rightarrow t+1$ execution so today's probability becomes tomorrow's position.

Signals are built from a simple threshold (default 0.5). Positions are binary long/short, rebalanced daily, and returns are the product of yesterday's signal with today's EUR/USD percentage move. The notebook then computes and reports the usual trading statistics on the full OOF timeline: annualized Sharpe, CAGR, volatility, maximum drawdown, hit rate, turnover, and trade count. Equity curves are plotted side by side, with optional drawdown and rolling-Sharpe views to compare stability. No transaction costs are applied here, so results are an upper bound.

Two quick stress tests are included to understand robustness. First, a threshold sweep shows how performance changes when you demand higher confidence before taking a trade. Second, a simple cost stress test applies a small per-trade spread to gauge how sensitive each strategy is to frictions; per-regime models usually turn over less and should degrade more gracefully than a high-churn baseline. The notebook finishes by exporting standardized summaries (tables and curves) for the report so results from all three streams can be cited consistently.

6. Results

6.1 Baseline Model Selection (NB 03)

The following table presents the results upon which we base the baseline model selection:

	Accuracy	Precision	Recall	F1	AUC
Logistic Regression	0.804	0.834	0.751	0.790	0.881
XGBoost	0.836	0.857	0.800	0.827	0.924

Table 1: Baseline Models Performance

XGB outperforms Logistic Regression across all metrics, with a notable lead in F1 and AUC. Logistic Regression remains competitive but shows lower recall and overall predictive power. XGB will be the baseline model to beat which won't be easy considering the high performance.

6.2 Per-Regime Model Selection (NB 04)

The results for the per-regimes models are shown in Table 2:

Regime	*Model*	*Acc*	*Prec*	*Rec*	*F1*	*AUC*
0	rf	0.853	0.870	0.824	0.846	0.927
0	xgb	0.841	0.862	0.807	0.833	0.923
1	xgb	0.862	0.859	0.863	0.861	0.925
1	rf	0.864	0.871	0.851	0.861	0.919

Table 2: Per-Regime Models Performance

Regime 0 (more turbulent): Random Forest wins. F1 0.846, AUC 0.927 (vs XGB F1 0.833, AUC 0.923). Higher recall (0.824) with strong precision (0.870) suggests RF handles noisy, non-linear structure slightly better here.

Regime 1 (calmer/trending): XGBoost and RF are neck-and-neck, both F1 0.861. Tie-break by AUC favors XGB (0.925 vs 0.919), so we select XGB for this regime.

Both regime models beat the global baseline (F1 0.827). RF will be deploy for Regime 0 and XGB for Regime 1, then stitch their OOF probabilities for backtesting.

6.3 LSTM Perfomance (NB 05)

Even after multiple rounds of tuning, LSTM underwhelms; AUC 0.522 ≈ random, acc 0.508, and it basically flags everything as "up" (rec 0.945, prec 0.500), which inflates F1 0.654 without real ranking skill. LSTM's probabilities aren't informative out-of-sample. It's likely overfit and biased toward the majority class. Possibly a quick threshold sweep, and probability calibration (isotonic/Platt) might help. If that doesn't move AUC, perhaps harder pruning, shorter windows, smaller network and more regularization/weight decay, or per-regime LSTMs might do the trick. Unfortunately, due to time constraints, LSTM will move on to the backtesting stage as is.

6.4 Backtesting Results (NB 06)

Finally, the following backtesting results serves as conclusion to the project code:

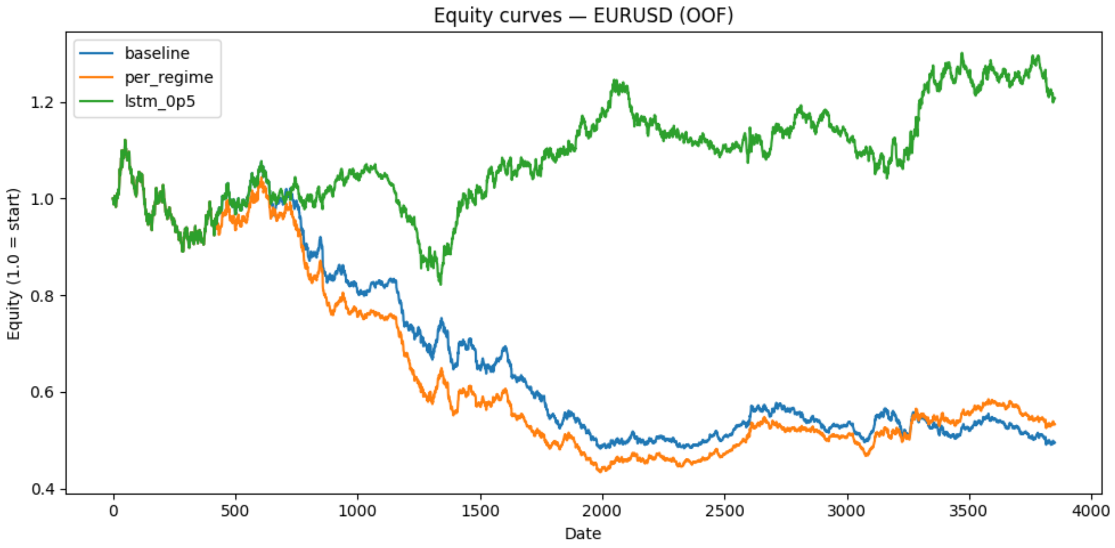

Figure 4: Models Backtesting Performances

	CAGR	***Sharpe***	***Vol***	***MDD***	***Hit***	***Turn***	***Trades***
LSTM	**0.012**	**0.188**	**0.085**	**-0.267**	**0.499**	**9.619**	**73.0**
Per-Regime	**-0.040**	**-0.438**	**0.086**	**-0.613**	**0.477**	**212.476**	**1623.0**
Baseline	**-0.045**	**-0.493**	**0.086**	**-0.570**	**0.476**	**218.758**	**1671.0**

Table 3: Models Backtesting Statistics

Baseline & Per-regime both lose money out-of-sample: CAGR –4.5% / –4.0%, Sharpe –0.49 / –0.44, hit ≈ 47–48%, and ~1,600–1,670 trades. That looks like over-trading on weak edges—equity drifts down, big drawdowns (–57% / –61%).

LSTM (0.5 threshold) is the only one that stays afloat: CAGR +1.2%, Sharpe 0.19, max DD –27%, and just 73 trades (turnover ~10 vs ~200+). The equity curve trends up because the model is highly selective—it sits out most days and trades only when confident. Hit rate is ~50%, so the edge likely comes from *when* it chooses to be exposed (average win > average loss) rather than superior classification skill.

7. Discussion

7.1 Answers to the Research Questions

RQ1 – Do regime-specific models beat a single global model (statistically and economically)? Statistically, yes but only modestly. The per-regime learners improve F1 from 0.827 (global XGB) to 0.846–0.861 while keeping AUC in the 0.919–0.927 range. That suggests conditioning on regime does help the classifier separate classes a bit better. Economically, the answer is no under our simple t→t+1 backtest: both baseline and per_regime strategies are negative (CAGR −4.5% vs −4.0%, Sharpe −0.49 vs −0.44) with very high turnover (~1.6k trades). The weak translation from better probabilities to PnL likely reflects three issues:

1. low regime separability (silhouette ~0.28),
2. a naïve threshold of 0.5 that over-trades small edges
3. no transaction costs, which would worsen both lines.

RQ2 – Does an LSTM add incremental performance over tree/linear baselines? On pure classification metrics, no. The LSTM's OOF AUC ≈ 0.52 is close to random and its F1=0.65 is inflated by very high recall (0.95) with precision ≈ 0.50, meaning it tends to call "up" frequently, not rank well. Economically, however, the LSTM with a fixed 0.5 threshold is the only strategy that avoids capital decay (CAGR ~+1.2%, Sharpe ~0.19) because it trades very selectively (73 trades total) and thereby sidesteps many losing days. In short: the network does not add robust predictive skill, but its sparsity acts as a crude trade filter that helps a simple strategy.

7.2 Implications and Practical Applications

The small but consistent F1/AUC gains from regime conditioning indicate there is state dependence worth exploiting. Yet probability quality does not translate to profitability in this case. For traders, two practical adjustments follow:

- Translate probabilities with care. Calibrate scores (Platt/isotonic), use regime-specific thresholds (and possibly confidence bands), and control turnover explicitly. The per-regime models likely need higher entry bars and lower/zero position when uncertainty is high.

- Use sequence models as filters, not forecasters. Given the LSTM's poor AUC but decent trading curve, a pragmatic role is exposure gating: only take trades when a sequence model and a tree/linear model agree, or let the LSTM veto low-confidence days to reduce churn.

Operationally, the HMM quality matters. Better regime labeling (more informative inputs, refined hyperparameters, or allowing three states) should strengthen both pruning and per-regime training. Finally, all deployment decisions should be re-tested with transaction costs and slippage, rolling re-training, and out-of-time holdouts.

7.3 Limitations and Ethical Considerations

Several limits temper these results. The study uses a single asset (EUR/USD) at daily frequency, so conclusions may not generalize to intraday or other FX pairs. yfinance data and close-to-close returns ignore microstructure effects. Backtests exclude costs and assume perfect *t→t+1* execution; both likely overstate performance. Regime separation is weak (silhouette ~0.28), which caps the upside of specialization. Feature engineering and per-regime pruning risk multiple-testing bias despite rolling CV. The LSTM search was necessarily narrow; different windowing, regularization, or per-regime RNNs might behave differently.

Ethically, regime-aware systems can encourage leverage and pro-cyclical behavior if left unchecked; risk limits, position caps, and scenario tests should be enforced by design. Data provenance and transformations must be transparent to avoid hidden leakage, and model reporting should foreground out-of-fold and after-cost results rather than cherry-picked windows. Finally, prefer simpler, auditable models when performance is close—clarity and control matter as much as a few basis points on a backtest.

8. Conclusion & Future Work

This project set out to see whether markets “speak in regimes” and whether models that listen perform better. On the statistical side, regime-specific learners did edge out a single global model (F1 ≈ 0.85–0.86 vs. 0.83; AUC ≈ 0.92+ across the board). On the trading side, that lift did not translate into profits under a simple t→t+1 strategy: both baseline and per-regime backtests lost money with high turnover. The LSTM did not add ranking skill (AUC ~0.52) yet delivered the only positive equity curve by being extremely selective—useful as an exposure filter rather than a forecaster. Net-net: regimes matter, but converting slightly better probabilities into robust PnL requires sharper regime labeling and a more careful probability-to-trade mapping.

In terms of future works, a natural next step would be to tighten regime detection with duration-aware models and modest exogenous drivers so state changes are steadier and more explainable. Keep the predictive stack simple but more utility-aligned: tune thresholds and sizing per regime with costs in the loop, and validate on purged walk-forward tests. Try a lightweight attention or mixture-of-experts model only if it demonstrably improves after-cost results; otherwise favor the current tree/linear setup. Finally, harden the pipeline for daily runs and sanity-check portability on a second asset to confirm the gains aren’t EUR/USD-specific.

References

Appel, G. (2005). *Technical analysis: Power tools for active investors*. Financial Times Prentice Hall.

Bollinger, J. (2002). *Bollinger on Bollinger Bands*. McGraw–Hill.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2), 357–384.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065), 20150202.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (pp. 4765–4774).

McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51–56).

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.

Van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2), 22–30.

Wilder, J. W. (1978). *New concepts in technical trading systems*. Trend Research.

Aroussi, R. (2019). *yfinance* [Computer software]. GitHub. (Yahoo! Finance market data downloader used for data acquisition.)

hmmlearn Developers. (2019). *hmmlearn* [Computer software]. (Hidden Markov Model library used for regime labeling.)