// MODULE · MACHINE LEARNING

The model is never the alpha.

Twelve features, six model families, four CV schemes, six overfitting traps, and the fifteen gates that hold the line.

12	06	06	15
FEATURES	MODELS	TRAPS	GATES

NEXURAL · SAGE IDEAS LLC
PUBLISHED 2026 · PERPLEXITY COMPUTER
nexural.com/learn/automation/machine-learning

// table of contents

What you will learn.

Each section is self-contained. If you only have time for one, read section 03 — cross-validation is the single largest source of phantom Sharpe in retail ML.

// section 01

Twelve features. Five families.

The most important decision in ML trading is feature engineering, not model choice. Below: twelve canonical features grouped into five families. Real strategies use 30-200 features — but every one descends from a family here. Master the families first; everything else is a variation.

The five families: price (log returns, z-score), volume (z-score, OBV, order-book imbalance), volatility (realized vol, ATR), momentum (RSI, MACD), and context (cross-asset beta, calendar, macro regime). Each feature carries a formula, the intuition for why it works, and the most common pitfall.

Feature	Family	One-liner
Log returns	PRICE	Foundation feature. Stationary; sums across horizons.
Realized volatility	VOLATILITY	Standard deviation of recent returns; the σ in every Sharpe.
RSI(14)	MOMENTUM	Relative Strength Index — bounded momentum oscillator.
MACD	MOMENTUM	Difference of two EMAs; the classic trend feature.
ATR(14)	VOLATILITY	Average True Range — measures absolute range, not %.
Volume z-score	VOLUME	How extreme is today's volume vs recent history.
OBV	VOLUME	On-Balance Volume — cumulative signed volume.
Z-score of price	PRICE	Mean-reversion feature; (price - rolling mean) / rolling std.
Cross-asset beta	CONTEXT	Rolling beta to a benchmark (SPY, sector ETF, BTC).
Day-of-week / month	CONTEXT	Calendar features — categorical, often forgotten.
Order-book imbalance	VOLUME	Bid size vs ask size — the strongest sub-minute feature.
Macro regime flag	CONTEXT	External state: VIX bucket, yield curve, Fed stance.

// section 01 · features 1-2 of 12

Feature deep dive (1–2)

Log returns

Family. PRICE · One-liner. Foundation feature. Stationary; sums across horizons.

Formula

```
r_t = ln(P_t / P_{t-1})
```

Intuition

Models almost always work in log-return space, not price space. Returns are roughly stationary and additive across horizons; prices are not. Every other feature in this catalog is downstream of returns.

Common pitfall

Mixing raw prices and returns in the same model. Prices have unit roots — coefficients become unstable across regimes and the model silently re-learns scale every time the underlying drifts.

Realized volatility

Family. VOLATILITY · One-liner. Standard deviation of recent returns; the σ in every Sharpe.

Formula

```
σ_t = std(r_{t-N..t-1})
```

Intuition

Volatility clusters — high-vol days follow high-vol days. Even a simple 20-day std predicts next-day vol better than chance. Realized vol is the cheapest, most reliable context feature you can add to any model.

Common pitfall

Using forward-looking windows. σ_t must be computable from t-1 information only. Centering the rolling window on t rather than ending it at t-1 is the most common form of look-ahead in retail vol features.

// section 01 · features 3-4 of 12

Feature deep dive (3–4)

RSI(14)

Family. MOMENTUM · One-liner. Relative Strength Index — bounded momentum oscillator.

Formula

```
RSI = 100 - 100 / (1 + avg_gain / avg_loss)
```

Intuition

Bounded [0, 100]. >70 often signals overbought, <30 oversold — but those thresholds are regime-dependent. Treat as a feature, not a signal. RSI's value is its boundedness, which makes it a clean input to any non-linear model.

Common pitfall

Using RSI alone as a strategy. It works as a feature among many; it does not work as a standalone rule once costs are included. The 'RSI < 30 buy' rule has been arbitraged out of every liquid name.

MACD

Family. MOMENTUM · One-liner. Difference of two EMAs; the classic trend feature.

Formula

```
MACD = EMA(12) - EMA(26); signal = EMA(MACD, 9)
```

Intuition

Captures momentum shifts at the cost of lag. The MACD-signal crossover is the textbook trigger but works poorly without volatility context. Pair with realized vol to filter false signals in choppy regimes.

Common pitfall

Constants 12/26/9 are arbitrary — tuning them on the test set is the most common form of overfitting in retail systems. If you change them, change them once and never look back; do not grid-search them on validation.

// section 01 · features 5-6 of 12

Feature deep dive (5–6)

ATR(14)

Family. VOLATILITY · One-liner. Average True Range — measures absolute range, not %.

Formula

```
ATR = avg(max(H-L, |H-C_{t-1}|, |L-C_{t-1}|))
```

Intuition

ATR scales with the absolute price level. Use it for stop sizing and position sizing — not for cross-asset comparison. The single most-used feature for risk-based position sizing in retail systems.

Common pitfall

Forgetting that ATR is in price units, not percent. A $5 ATR means very different things on a $50 vs a $500 stock. Always normalize by price (ATR/close) before feeding to a cross-sectional model.

Volume z-score

Family. VOLUME · One-liner. How extreme is today's volume vs recent history.

Formula

```
z = (V_t - mean(V_{t-N..t-1})) / std(V_{t-N..t-1})
```

Intuition

Unusual volume tends to mark regime breaks. A +3σ volume spike often precedes a multi-day trend continuation or sharp reversal. The volume z-score is the cheapest regime-shift detector retail traders have.

Common pitfall

Not adjusting for day-of-week or known events. Tuesday morning baseline volume differs from Friday close. The right denominator is volume over the same minute of the same day-of-week, not raw daily volume.

// section 01 · features 7-8 of 12

Feature deep dive (7–8)

OBV

Family. VOLUME · One-liner. On-Balance Volume — cumulative signed volume.

Formula

```
OBV_t = OBV_{t-1} + sign(r_t) * V_t
```

Intuition

Captures accumulation/distribution. Divergences between price and OBV are a classic precursor to reversals. Most useful as the input to a ratio or rolling statistic, not as a raw level.

Common pitfall

OBV is non-stationary by construction. Always pass it through a difference or ratio transform before feeding to most ML models — otherwise it dominates feature importance for the wrong reason.

Z-score of price

Family. PRICE · One-liner. Mean-reversion feature; (price - rolling mean) / rolling std.

Formula

```
z = (P_t - mean(P_{t-N..t-1})) / std(P_{t-N..t-1})
```

Intuition

The bedrock of statistical arbitrage. $|z| > 2$ is one of the most replicable retail features for mean-reverting names. Combined with cross-sectional ranking it builds the core of every pair / stat-arb book.

Common pitfall

Choice of N is regime-dependent. A 20-day z-score works in chop and fails in trends; pairing with regime detection is essential. If you only use one N, use it consistently across all symbols — do not pick a different N per name.

// section 01 · features 9-10 of 12

Feature deep dive (9–10)

Cross-asset beta

Family. CONTEXT · One-liner. Rolling beta to a benchmark (SPY, sector ETF, BTC).

Formula

```
β = cov(r_asset, r_bench) / var(r_bench)
```

Intuition

Rolling beta captures regime shifts — a stock can be high-beta in one quarter and defensive the next. Useful as conditioning context, especially for long/short books where you want to neutralize benchmark exposure.

Common pitfall

Using too short a window (beta is noisy at 20 days) or too long (loses regime signal). 60-day is a workable default for daily bars. Always check that |β| has not crossed an unphysical threshold like 5 — that almost always means a data alignment bug.

Day-of-week / month

Family. CONTEXT · One-liner. Calendar features — categorical, often forgotten.

Formula

```
dow = pd.Timestamp(t).dayofweek
```

Intuition

Equity returns have real day-of-week effects (Monday is different). Crypto has weekend liquidity effects. One-hot or sin/cos encode it. Calendar features are nearly free to compute and capture seasonality most models otherwise miss.

Common pitfall

Treating dow as ordinal. 0..4 implies 'Friday is greater than Monday' — almost always meaningless. Use one-hot or cyclical encoding (sin(2π·dow/5), cos(2π·dow/5)). The same applies to month-of-year and minute-of-hour.

// section 01 · features 11-12 of 12

Feature deep dive (11–12)

Order-book imbalance

Family. VOLUME · One-liner. Bid size vs ask size — the strongest sub-minute feature.

Formula

```
OBI = (bid_size - ask_size) / (bid_size + ask_size)
```

Intuition

Persistent imbalance predicts short-horizon price moves. The relationship breaks down past 30-60 seconds for most names. The strongest predictor available at the millisecond scale and the weakest at the daily scale.

Common pitfall

Latency. OBI computed on stale L2 is worse than no feature. Do not use it unless your data path is sub-100ms. Backtesting OBI on snapshot data sampled every minute will produce a wildly optimistic Sharpe that vanishes in production.

Macro regime flag

Family. CONTEXT · One-liner. External state: VIX bucket, yield curve, Fed stance.

Formula

```
regime = bucket(VIX_t, [15, 25, 35])
```

Intuition

Macro features rarely have within-day predictive power but they condition every other feature. Train one model per regime if you have data, or pass the regime flag as a feature so a single model can learn regime-conditional weights.

Common pitfall

Look-ahead via revised macro series. Use as-of release dates, never revised values, when constructing the history. The FRED API returns the latest revision by default; you must explicitly ask for vintage data.

// interlude · label engineering

The label is half the model.

Features get all the attention. Labels — the y you are predicting — quietly decide whether your model has any chance of working. The wrong label can make a signal-rich feature set look like noise; the right label can turn a thin feature set into a tradeable strategy.

Three label families

Fixed-horizon return. r_{t+N} for some N (e.g. 5 days). Simple, leaks no information beyond N. Weak because the same path can hit large gains then unwind to zero by day N — the model never sees the gain.

Triple-barrier (de Prado). Set an upper barrier, a lower barrier, and a time limit. The label is whichever barrier is hit first. Captures path-dependent payoff the way a real stop-loss / take-profit strategy would. The right default for trading ML in 2026.

Meta-labeling. First model predicts direction; second model predicts whether to size up the first model's call. Splits the problem of what from how much — the second model gets much cleaner features to learn from.

Common pitfall

Using overlapping labels (every bar gets a 5-day-forward label) and treating them as independent samples in CV. They are not — they share information across rows. This is what the 'purge' in purged CV is built to fix.

// section 02

Six model families.

Six honest model archetypes. Below: a comparison table on four axes (interpretability, training cost, latency, data hunger). Then per-model pages with when it's right, when it's wrong, and the honest take. The most expensive mistake retail ML traders make is skipping the linear baseline — start there, always.

Honest principle: always ship the simplest model that hits your target. Linear → RF → GBM → anything else. If you cannot beat the previous step by ≥ 10% out-of-sample Sharpe, do not graduate.

Model	Family	Interpret	Train cost	Latency	Data hunger
Linear regression / Logistic	LINEAR	High	Low	Low	Low
Random Forest	TREES	Mid	Low	Mid	Mid
Gradient Boosting (XGBoost / LightGBM)	TREES	Mid	Mid	Mid	Mid
Neural Net (MLP)	NEURAL	Low	Mid	Low	High
Transformer / Attention	NEURAL	Low	High	Mid	High
Ridge / Elastic Net	LINEAR	High	Low	Low	Low

// section 02 · model 1 of 6

Linear regression / Logistic

INTERPRET	TRAIN COST	LATENCY	DATA HUNGER
High	Low	Low	Low

When it is right

Small data (<10k rows). Strong linear prior. Need to explain coefficients to compliance.

When it is wrong

Non-linear interactions matter. Features have heavy tails or regime shifts.

Honest take

An honest linear baseline beats 60% of ML models in retail trading once costs are paid. Always start here. If a tree-based model can't beat it by ≥ 10% Sharpe out-of-sample, you don't need ML — you need better features or a different problem.

// section 02 · model 2 of 6

Random Forest

INTERPRET	TRAIN COST	LATENCY	DATA HUNGER
Mid	Low	Mid	Mid

When it is right

Tabular features. Mixed types. You want a strong baseline without tuning. Variance reduction via bagging.

When it is wrong

High-frequency latency budget. Sequential or temporal dependence is the dominant signal.

Honest take

The most under-rated retail model. Boring, accurate, robust to outliers. If GBM beats it by < 5% AUC, ship the RF — fewer hyperparameters to overfit, lighter ops burden, and feature importances that compliance can read.

// section 02 · model 3 of 6

Gradient Boosting (XGBoost / LightGBM)

INTERPRET	TRAIN COST	LATENCY	DATA HUNGER
Mid	Mid	Mid	Mid

When it is right

Tabular data with enough volume to support tuning. Need state-of-the-art on structured features.

When it is wrong

Very small data (<1k rows) — overfits aggressively. Online learning required.

Honest take

The retail-quant default for good reason. Treat early-stopping rounds and max-depth like your life depends on them. A tuned GBM on bad features still loses to a linear model on good features — feature engineering is the lever, not algorithm choice.

// section 02 · model 4 of 6

Neural Net (MLP)

INTERPRET	TRAIN COST	LATENCY	DATA HUNGER
Low	Mid	Low	High

When it is right

Smooth non-linear relationships with abundant data. Need cheap inference at scale.

When it is wrong

Tabular data with <100k rows. Inputs have wildly different scales without normalization.

Honest take

Almost never the right first choice for retail tabular trading. The papers that show MLPs winning use millions of rows and careful tuning — neither of which you have. Use as the 'last 5% Sharpe' step, not the foundation.

// section 02 · model 5 of 6

Transformer / Attention

INTERPRET	TRAIN COST	LATENCY	DATA HUNGER
Low	High	Mid	High

When it is right

Sequence problems with rich token vocabularies — news, alt-data text, multi-asset attention across hundreds of names.

When it is wrong

Single-name tabular features. Sub-1s inference latency. Limited GPU.

Honest take

Mostly oversold in retail trading. The 'transformer for time series' literature is mixed at best — GBMs match or beat them on tabular features. If you need a transformer, you usually need an alt-data team first.

// section 02 · model 6 of 6

Ridge / Elastic Net

INTERPRET	TRAIN COST	LATENCY	DATA HUNGER
High	Low	Low	Low

When it is right

Many correlated features (>50). Need automatic feature selection (L1) or stable coefficients (L2).

When it is wrong

Truly non-linear signal. Need feature interactions.

Honest take

The right baseline when feature engineering produced 100+ correlated columns. Use ElasticNetCV with a 0.5 mix as a starting point and let it pick features for you. Underrated as a coefficient-stable replacement for plain linear regression.

// section 03

Cross-validation.

Four CV schemes — only two are honest for time-series ML, and one is the default scikit-learn ships with. Below: the diagram for each scheme, the verdict, and why. If you take only one thing from this PDF: never use random k-fold on time-series.

The legend: TRAIN = model fits on this block. TEST = held-out evaluation block. PURGE = removed to prevent label overlap. EMBARGO = quarantine after test to clear information leakage. — = unused in this fold.

// cv scheme · kfold

Random k-fold

WRONG

One-liner. Shuffle rows. Train on 4 chunks, test on 1. Repeat 5 times.

Diagram (→ time →)

```
Fold 1: TEST  TRAIN TRAIN TRAIN TRAIN
Fold 2: TRAIN TEST  TRAIN TRAIN TRAIN
Fold 3: TRAIN TRAIN TEST  TRAIN TRAIN
Fold 4: TRAIN TRAIN TRAIN TEST  TRAIN
Fold 5: TRAIN TRAIN TRAIN TRAIN TEST
```

Detail

This is the scikit-learn default and the number-one source of phantom Sharpe in retail ML. Random folds leak future bars into the training set — the model can see tomorrow during training and 'predicts' it during testing. Backtests built on k-fold CV regularly inflate true Sharpe by 2-4×.

// cv scheme · walk

Walk-forward

OK BASELINE

One-liner. Train on past N, test on next M. Slide. Repeat until end.

Diagram (→ time →)

```
Fold 1: TRAIN TEST  ---   ---   ---
Fold 2: TRAIN TRAIN TEST  ---   ---
Fold 3: TRAIN TRAIN TRAIN TEST  ---
Fold 4: TRAIN TRAIN TRAIN TRAIN TEST
```

Detail

The minimum honest scheme. Each fold only uses past data to predict the future. Sharpe estimates from walk-forward are noisy but unbiased. Use this as your baseline; everything below is more sophisticated, not strictly better.

// cv scheme · purged

Purged k-fold + embargo

BEST FOR DAILY

One-liner. k-fold with a buffer zone (purge) and a quarantine after each test (embargo).

Diagram (→ time →)

```
Fold 1: TEST  PURGE EMBRG TRAIN TRAIN
Fold 2: TRAIN TEST  PURGE EMBRG TRAIN
Fold 3: TRAIN PURGE TEST  PURGE EMBRG
Fold 4: TRAIN TRAIN PURGE TEST  PURGE
Fold 5: TRAIN TRAIN TRAIN PURGE TEST
```

Detail

From López de Prado's Advances in Financial Machine Learning. The 'purge' removes training rows whose labels overlap the test window; the 'embargo' prevents the next fold's training set from starting until information leakage has cleared. The gold standard for daily-bar ML once your labels span multiple bars (e.g., triple-barrier, n-day forward return).

// cv scheme · cpcv

Combinatorial purged CV

BEST FOR SELECTION

One-liner. Test on multiple combinations of held-out folds — many backtest paths.

Diagram (→ time →)

```
Path 1: TEST  TRAIN TEST  PURGE TRAIN
Path 2: TEST  PURGE TRAIN TEST  TRAIN
Path 3: TRAIN TEST  PURGE TEST  TRAIN
Path 4: PURGE TRAIN TEST  TRAIN TEST
```

Detail

Generates many out-of-sample paths instead of one. Critical when selecting between strategies: a single walk-forward Sharpe can be lucky, but the distribution of paths from CPCV reveals the strategy's true variance. Slow; reserve for the final 2-3 candidates.

// section 04

Six overfitting traps.

Every ML-trading paper that does not pre-register hypotheses, use purged CV, and report deflated Sharpe is — statistically — making one of these six mistakes. Each trap below has the one-liner, how to detect it, how to fix it, and a concrete real example.

These traps are ordered by frequency in retail systems. The top three (look-ahead, snooping, selection) account for the majority of phantom Sharpe in published strategies; the bottom three are subtler but no less destructive.

// trap · lookahead

Look-ahead bias

One-liner. Using information that was not knowable at the time of the trade.

Detection

Compute every feature at time t using only data filed_at <= t. Audit by deliberately shifting the universe back by one bar and confirming Sharpe drops.

Fix

Always join on as-of timestamps (pd.merge_asof). Store every alt-data source with its 'filed_at' or 'release_time' separate from the timestamp it refers to.

Real example

Joining quarterly earnings on period-end date instead of report date: 'as of 2024-03-31' contains a number first published 2024-04-20. That is three weeks of look-ahead that vanishes on live trading.

// trap · snooping

Test-set snooping

One-liner. Tuning hyperparameters on the same data you report Sharpe from.

Detection

Compare your reported out-of-sample Sharpe against the median Sharpe across 100 randomly-perturbed validation splits. If the gap is > 30%, you have snooped.

Fix

Three sets: train / validation / blind test. The blind test is opened exactly once, after model selection is complete, and the result goes to the report regardless.

Real example

Trying 50 hyperparameter combos, picking the one with the best test Sharpe, and reporting that Sharpe as 'out of sample'. You have effectively trained on the test set.

// trap · selection

Selection bias

One-liner. Filtering the universe in a way that depends on outcomes.

Detection

Repeat the backtest with the universe selection date shifted forward. If Sharpe changes by > 20%, your selection is leaking.

Fix

Universe membership must be point-in-time. Use index-membership history (CRSP, Norgate) instead of 'current S&P 500'.

Real example

Backtesting on 'today's S&P 500' over 2010-2024 — every survivor's quality is conditioned on the next 14 years. Real-time the index had additions and deletions you would have traded.

// trap · phacking

p-hacking / multiple hypothesis

One-liner. Trying 1000 strategies, reporting the best. The best is noise.

Detection

Track every strategy you test. Compute the Bonferroni-corrected threshold: Sharpe must beat the median Sharpe + 2 × σ across all attempts.

Fix

Pre-register the strategy hypothesis before backtesting. Apply the Deflated Sharpe Ratio (López de Prado) when reporting.

Real example

Sharpe 2.1 on the best of 50 grid-search combinations sounds great until you compute the deflated SR — once corrected for 50 trials, the true effect is closer to 0.4.

// trap · regime

Regime curve-fitting

One-liner. Optimizing for the regime that happened to dominate your sample.

Detection

Backtest separately on each regime bucket (low / mid / high VIX, or rising / falling rates). If Sharpe is positive in only one bucket, you are not robust.

Fix

Stratified backtest: hold out one full regime as test. If the strategy only works in 2017-style low-vol environments, the marketing should say that.

Real example

A short-volatility strategy trained on 2010-2019 looked excellent until March 2020. The model was fit to a single regime and had no protection against the regime ending.

// trap · hyper

Hyperparameter over-tuning

One-liner. Burning your degrees of freedom on n_estimators and max_depth.

Detection

Plot validation Sharpe vs. each hyperparameter. Sharp peaks (one good value, neighbors collapse) signal overfitting; broad plateaus signal real signal.

Fix

Coarse grid first, fine grid only if the coarse one shows a plateau. Cap total combinations at sqrt(n_train) as a rule of thumb.

Real example

An XGBoost where Sharpe = 2.4 at max_depth=7, but max_depth=6 gives 0.9 and max_depth=8 gives 0.7. That peak is a single noise sample, not a real optimum.

// interlude · deflated sharpe ratio

Worked example: Deflated Sharpe Ratio.

You backtested 50 hyperparameter combinations of an XGBoost on US equities. The best combo posted an out-of-sample Sharpe of 2.1. Sounds publishable. Then you apply the deflation correction.

The math (Bailey & López de Prado, 2014)

```
# Deflated Sharpe Ratio (DSR)
# SR_hat       = observed Sharpe of best trial
# N            = number of trials (50)
# T            = number of OOS observations (e.g. 252 trading days)
# skew, kurt   = third / fourth moments of the strategy's returns

# Expected max Sharpe under the null (no skill, N trials):
#   E[SR_max | null] ≈ (2 * ln(N)) ** 0.5  *  std_dev_of_trial_sharpes

# DSR is the probability that SR_hat exceeds E[SR_max | null]
# adjusting for T, skew, and kurt.
```

Numerical sketch

Assume std of trial Sharpes ≈ 0.8 across the 50 trials. Then E[SR_max | null] ≈ sqrt(2 · ln(50)) · 0.8 ≈ 2.50. The reported 2.1 is below the null expectation — there is no evidence of skill. The DSR rounds to roughly 0.10, meaning a 10% probability the strategy is real.

What this means

A 2.1 Sharpe on the best of 50 trials is statistically indistinguishable from luck. To clear the deflated threshold you need either (a) fewer trials, (b) more OOS data, or (c) a higher observed Sharpe. Pre-registering the strategy hypothesis is the cheapest way to shrink N from 50 to 1.

// section 05

Production-readiness gates.

Fifteen gates across five categories — data, model, validation, deployment, monitoring. Production-ready means all fifteen ticked AND documented. Skipping any one of them is how retail ML loses money.

ID	Category	Gate
01	DATA	Training data is point-in-time, with delisted symbols included
02	DATA	Every feature has a documented filed_at timestamp
03	DATA	Train/val/test split has temporal gap and embargo
04	MODEL	Linear baseline established; ML beats it by ≥ 10% OOS Sharpe
05	MODEL	Feature importance reviewed; no single feature > 40%
06	MODEL	Hyperparameters from coarse grid; plateau (not peak) selected
07	VALIDATION	Purged + embargoed CV with multiple seeds
08	VALIDATION	Deflated Sharpe Ratio computed for the reported result
09	VALIDATION	Strategy works in ≥ 2 regime buckets (vol, rate, drawdown)
10	DEPLOYMENT	Inference path is deterministic and version-pinned
11	DEPLOYMENT	Position sizing capped; kill-switch tested in production
12	DEPLOYMENT	Live latency < 50% of feature decay horizon
13	MONITORING	Live PnL tracked vs backtest expectation; alert on > 2σ deviation
14	MONITORING	Feature drift monitored; PSI > 0.25 triggers retrain
15	MONITORING	Quarterly review with retire/retrain/refit decision

// gates · data

Data

01. Training data is point-in-time, with delisted symbols included

Universe membership uses historical (not current) index files. Delisted tickers retained with their final price. Verify by re-running the universe query with a date 1 year in the past and checking that the count differs.

02. Every feature has a documented filed_at timestamp

Fundamentals, news, alt-data all carry the moment they became knowable — never the period they describe. The feature schema includes filed_at as a non-null column.

03. Train/val/test split has temporal gap and embargo

No row from after test_start appears in training. A purge window equal to the label horizon separates them. The embargo prevents the next training set from absorbing test-period information.

// gates · model

Model

04. Linear baseline established; ML beats it by ≥ 10% OOS Sharpe

Without a beaten linear baseline you have no evidence ML adds value. Document the baseline metrics and the date; rerun the baseline whenever the feature set changes.

05. Feature importance reviewed; no single feature > 40%

Heavy concentration on one feature usually signals leakage. Audit the top-3 features for look-ahead. If a single feature dominates, ablate it and check that the model degrades gracefully — not collapses.

06. Hyperparameters from coarse grid; plateau (not peak) selected

Picked a plateau region of hyperparameter space, not a single best point. Saved the full grid for audit. A plateau is robust; a peak is overfit.

// gates · validation

Validation

07. Purged + embargoed CV with multiple seeds

At least 5 seeds with the de Prado scheme. Sharpe distribution width reported, not just the median. The seed-to-seed variance is itself a statistic.

08. Deflated Sharpe Ratio computed for the reported result

DSR > 0.4 (depending on number of trials). The deflation correction is applied for every backtest you ran — not just the final one.

09. Strategy works in ≥ 2 regime buckets (vol, rate, drawdown)

Bucket the test window by macro regime. Sharpe is positive in ≥ 2 of 3 buckets, not concentrated in one. A strategy that only works in low-VIX is not 'robust' — it is regime-dependent and must be marketed as such.

// gates · deployment

Deployment

10. Inference path is deterministic and version-pinned

model_hash + feature_hash + lib_hash recorded with every prediction. A live decision is reproducible six months later. Pin every Python and CUDA version; pin model weights by SHA256, not by filename.

11. Position sizing capped; kill-switch tested in production

Max position size; daily VaR limit; kill-switch verified on a paper-trading run before any live capital. The kill-switch must be tested at least quarterly — assume it has rotted otherwise.

12. Live latency < 50% of feature decay horizon

If the alpha decays in 5 minutes, your data-to-order latency is under 2.5 minutes including signal computation. Measure end-to-end, including broker ack — not just model inference.

// gates · monitoring

Monitoring

13. Live PnL tracked vs backtest expectation; alert on > 2□ deviation

Daily PnL plotted against the backtest distribution. Page when the live trajectory exits the 95% band for 3+ days. The first 30 days of live PnL are the most informative evidence you will ever get.

14. Feature drift monitored; PSI > 0.25 triggers retrain

Population Stability Index between training and live feature distributions reviewed weekly. PSI > 0.10 is a warning; > 0.25 is a forced retrain.

15. Quarterly review with retire/retrain/refit decision

Standing meeting. Each model gets one of three labels per quarter. No model lives forever by default — the burden of proof is on continuation, not retirement.

// section 05 · operating cycle

The four-quarter operating cycle.

Once a model is live, it enters a four-quarter operating cycle. Each quarter has a specific verdict-meeting agenda. No model lives forever by default — the burden of proof is on continuation, not retirement.

Quarter	Action	Detail
Q1 · Drift	DRIFT	Feature drift (PSI) and live PnL distribution checked vs backtest. PSI > 0.10 warns; > 0.25 schedules a refit. Live PnL outside the 95% backtest band for 3+ consecutive weeks escalates.
Q2 · Refit	REFIT	If drift triggered, refit on rolling 3-year window. Re-run the entire validation and gate suite. Refit does not change features or hyperparameters — only the trained weights.
Q3 · Retrain	RETRAIN	If refit did not restore performance, expand the feature set. Re-run grid search (coarse only). New label/feature schema means new model version; old model goes to shadow.
Q4 · Retire	RETIRE	If retrain did not work, retire. Document the post-mortem: which feature broke, what changed in the market, what we will not try again. The graveyard is the most valuable artifact you produce.

Honest principle: the cycle has four exits but only one entrance. Once a model is live, drift / refit / retrain / retire is the only loop. Skipping the loop — letting a model run untouched for a year — is how good systems quietly stop working.

// appendix · glossary

Twenty-four terms, defined honestly.

Alpha decay	How quickly a signal stops working after discovery.
AUC	Area under ROC; classifier rank metric.
Backtest	Simulated trading on historical data.
Bagging	Bootstrap aggregating; train on resamples, average predictions.
Bonferroni	Multiple-test correction; divide alpha by N tests.
CPCV	Combinatorial Purged Cross-Validation (de Prado).
Deflated Sharpe	Sharpe adjusted for number of trials and moments of returns.
Drift (PSI)	Population Stability Index measuring distribution shift.
Embargo	Quarantine period after test fold to prevent information leakage.
Feature store	Versioned, cached store of computed features.
GBM	Gradient Boosted Machines (XGBoost, LightGBM, CatBoost).
K-fold (random)	CV scheme that shuffles rows; WRONG for time-series.
Label	The y you are predicting; outcome of the trade.
Look-ahead	Using information not knowable at time t.
MLP	Multi-layer perceptron; a feedforward neural net.
Meta-labeling	Second model predicts whether to trust the first.
Out-of-sample	Data not used during training; honest evaluation set.
p-hacking	Trying many strategies; reporting only the best.
Purge	Removing training rows whose labels overlap the test window.
Selection bias	Universe filter that depends on outcomes.
Sharpe ratio	Excess return / volatility; annualized.
Triple-barrier	Label scheme: first of upper / lower / time barrier.
Walk-forward	CV that always trains on past, tests on future.
XGBoost	Open-source gradient-boosting library; retail-quant default.

// section 06

Honest ML-trading checklist.

Before you ship an ML strategy into production, tick each item below. If any one fails, the backtests downstream are not yet trustworthy.

[ ] 01	Linear baseline beaten by ≥ 10% out-of-sample Sharpe.
[ ] 02	Purged k-fold + embargo CV with multiple seeds reported.
[ ] 03	Deflated Sharpe Ratio computed and disclosed.
[ ] 04	All features carry filed_at timestamps; no period-end joins.
[ ] 05	Universe is point-in-time, including delisted symbols.
[ ] 06	Hyperparameters from coarse grid; plateau selected (not peak).
[ ] 07	Strategy works in ≥ 2 regime buckets (vol, rate, drawdown).
[ ] 08	Inference path is deterministic and version-pinned.
[ ] 09	Position sizing capped; kill-switch tested in production.
[ ] 10	Live PnL monitored vs backtest expectation; 2σ-deviation alerts wired.
[ ] 11	Feature drift (PSI) reviewed weekly.
[ ] 12	Quarterly retire/retrain/refit review on the calendar.

Twelve boxes. If you cannot tick all twelve, document which you cannot tick AND why — that is the honest record.

// further reading

Sources and further reading.

[1] López de Prado — Advances in Financial Machine Learning (Wiley 2018).

https://www.wiley.com/en-us/Advances+in+Financial+Machine+Learning-p-9781119482086

[2] López de Prado — The Deflated Sharpe Ratio.

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2460551

[3] scikit-learn — Cross-validation strategies for time series.

https://scikit-learn.org/stable/modules/cross_validation.html#time-series-split

[4] XGBoost — Tunable parameters and early stopping.

https://xgboost.readthedocs.io/en/stable/parameter.html

[5] LightGBM — Microsoft gradient-boosting framework.

https://lightgbm.readthedocs.io/en/stable/

[6] Bailey & López de Prado — The Probability of Backtest Overfitting.

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2326253

[7] CRSP — Center for Research in Security Prices (PIT universes).

https://www.crsp.org/products/research-products/crsp-us-stock-databases

[8] Norgate Data — Survivorship-free index history.

https://norgatedata.com

[9] pandas — merge_asof for as-of joins.

https://pandas.pydata.org/docs/reference/api/pandas.merge_asof.html

[10] PSI / Population Stability Index — drift detection primer.

https://www.listendata.com/2015/05/population-stability-index.html

[11] Bonferroni correction — multiple hypothesis testing.

https://en.wikipedia.org/wiki/Bonferroni_correction

[12] Triple-barrier labeling — Lopez de Prado technique.

https://towardsdatascience.com/financial-machine-learning-part-0-bars-labels-and-stationarity-a2e5fc1b71b2

Continue reading → Next module: Operations