End-to-End AI-Enhanced Regime-Aware Hybrid Alpha Strategy Pipeline

[17-Step Workflow]

By Dr. Ray Islam
Adjunct Professor, IST
Geroge Mason University, Fairfax, VA, USA
01/16/2026

[Code and Other Contents: https://github.com/ray-islam/End-to-End-AI-Enhanced-Regime-Aware-Hybrid-Alpha-Strategy]

Overall Objective

An end-to-end hybrid trading strategy prototype that integrates market-based **alpha signals** (trend/momentum + mean-reversion) with FinBERT news sentiment, uses regime-aware gating to combine signals, applies realistic constraints and execution costs, and produces backtest results and reporting artifacts. This pipeline is designed to evolve from a single-ticker MVP (AAPL) into a scalable multi-asset system.

Overall Goals

• Create reproducible data ingestion and feature pipelines (raw > cleaned > features > feature store).

• Generate three alpha sleeves (trend, reversion, sentiment) and combine them using regime-aware gating.

• Translate alpha into tradable weights with constraints, position sizing, and execution-cost assumptions.

• Backtest weekly rebalancing long-only strategy and produce metrics + tear sheet + logs.

• Lay the groundwork for ML ranker/gating and walk-forward evaluation as future iterations.

Scripts: https://github.com/ray-islam/End-to-End-AI-Enhanced-Regime-Aware-Hybrid-Alpha-Strategy

Run Order

1. script_01_define_and_ingest.py (Steps 1–5)
2. script_02_market_features.py (Step 6)
3. script_03_sentiment_features.py (Steps 7–9)
4. script_04_alphas_regime_gating.py (Steps 10–12)

5. script_05_portfolio_execution.py (Steps 13–15)
6. script_06_backtest_report_prod.py (Steps 16–17)

17 Steps Checklist (with Script Mapping)

Step	What you do	Script	Primary Output Artifact
1	Define scope, universe, objective, and assumptions	Script 1	data/processed/strategy_spec.json
2	Ingest OHLCV daily bars (Polygon)	Script 1	data/raw/<TICKER>_ohlcv_daily_raw.csv
3	Ingest company news feed (Polygon reference news)	Script 1	data/raw/<TICKER>_news_raw.csv
4	Market data validation & cleaning (OHLCV sanity checks)	Script 1	data/processed/<TICKER>_ohlcv_daily_clean.csv
5	News validation & cleaning (dedup, normalize dates, drop invalid)	Script 1	data/processed/<TICKER>_news_clean.csv
6	Feature engineering: returns panel (returns, vol, liquidity, ranges)	Script 2	data/features/<TICKER>_returns_panel.parquet
7	FinBERT sentiment scoring on news text	Script 3	data/sentiment/<TICKER>_news_scored.csv
8	Aggregate sentiment to daily features (mean/sum/count	Script 3	data/sentiment/<TICKER>_sentiment_daily.csv

	)		
9	Build feature store (merge market features + sentiment features)	Script 3	data/features/<TICKER>_feature_store.parquet
10	Construct alpha sleeves: trend, mean-reversion, sentiment (standardized)	Script 4	data/features/<TICKER>_alphas_regime_gated.parquet
11	Regime detection (e.g., volatility-based risk_on / risk_off)	Script 4	regime labels in the same parquet
12	Gating/ensemble weighting: combine sleeves based on regime	Script 4	alpha_combined in the same parquet
13	Constraints & risk controls (liquidity, exposure caps, turnover caps)	Script 5	liquidity_ok + constrained targets
14	Position sizing (sigmoid + volatility targeting; clip to bounds)	Script 5	target_weight_raw
15	Execution model (cost bps incl. vol bump; used during backtest)	Script 5	exec_cost_bps
16	Backtesting: weekly rebalance, turnover cap, apply costs; walk-forward scaffold	Script 6	data/backtests/<TICKER>_backtest.csv

17	Reporting + production hygiene: tear sheet PDF, metrics JSON, logs, kill switch	Script 6	data/reports/<TICKER>_tear_sheet.pdf + logs

Notes / Tips

• If running inside Jupyter/IPython, do NOT use __file__. Use Path.cwd() or a resolve_base_dir() helper.

• If you see an 'xformers not installed' message, it is a warning (performance), not a failure.

• If pandas raises an aggregation TypeError, use the tuple-aggregation syntax: sent_mean=('sentiment_score','mean').

• Set environment variable KILL_SWITCH=1 to safely stop execution in Scripts 5–6 (production hygiene).

Expected Outputs

strategy_spec.json

Purpose: Stores the strategy definition (ticker, date range, objective).

How to read: Open in any text editor. Verify ticker/start/end match your intended run.

*_ohlcv_daily_raw.csv / *_ohlcv_daily_clean.csv

Purpose: Raw and cleaned daily market data used for feature engineering/backtest.

How to read: Open in Excel/Pandas. Confirm date continuity, close>0, volume>=0, and no duplicates.

*_news_raw.csv / *_news_clean.csv

Purpose: Raw and cleaned news articles pulled from Polygon.

How to read: Check published_utc/date fields. Ensure titles exist. Verify duplicates removed.

*_returns_panel.parquet

Purpose: Market feature panel containing returns/volatility/liquidity measures.

How to read: Load with pandas.read_parquet(). Inspect columns: ret_1d, vol_21d, adv_dollars_20d, fwd_ret_5d.

***_news_scored.csv**

Purpose: News articles with FinBERT sentiment labels/scores.

How to read: Review sentiment_label and sentiment_score. Spot-check a few titles vs scores for reasonableness.

***_sentiment_daily.csv**

Purpose: Daily aggregated sentiment features (mean/sum/count).

How to read: Look for non-zero sent_count days; compare sent_mean vs major news dates.

***_feature_store.parquet**

Purpose: Master dataset for modeling: market features + sentiment features aligned by date.

How to read: Ensure no look-ahead (sentiment uses same-day news). Confirm no NaNs in key fields.

***_alphas_regime_gated.parquet**

Purpose: Alpha sleeves + regime labels + combined alpha after gating.

How to read: Plot alpha_trend/alpha_reversion/alpha_sentiment and regime. Confirm alpha_combined changes with regime.

***_targets_with_costs.parquet**

Purpose: Portfolio targets with constraints + execution-cost parameters.

How to read: Inspect target_weight_raw (0..1), liquidity_ok flag, and exec_cost_bps distribution.

***_backtest.csv**

Purpose: Day-by-day backtest results including NAV, position weights, turnover, costs.

How to read: Plot nav over time; check turnover spikes align with rebalance dates; verify trade_cost is applied.

*_metrics.json

Purpose: Summary performance metrics.

How to read: Open JSON; compare annual_return, annual_vol, sharpe, max_drawdown; validate costs/turnover.

*_tear_sheet.pdf

Purpose: Visual tear sheet report (equity curve, drawdown, rolling sharpe, weights).

How to read: Confirm equity curve consistency, drawdown realism, and rolling sharpe stability. Use this as your review artifact.

logs/strategy_run.log

Purpose: Execution log for debugging and auditing.

How to read: Search for WARN/ERROR. Confirm script stages ran fully and outputs were saved.

Results

Performance Summary

total_return: 0.0350

annual_return: 0.0354

annual_vol: 0.0532

sharpe: 0.6804

max_drawdown: -0.0582

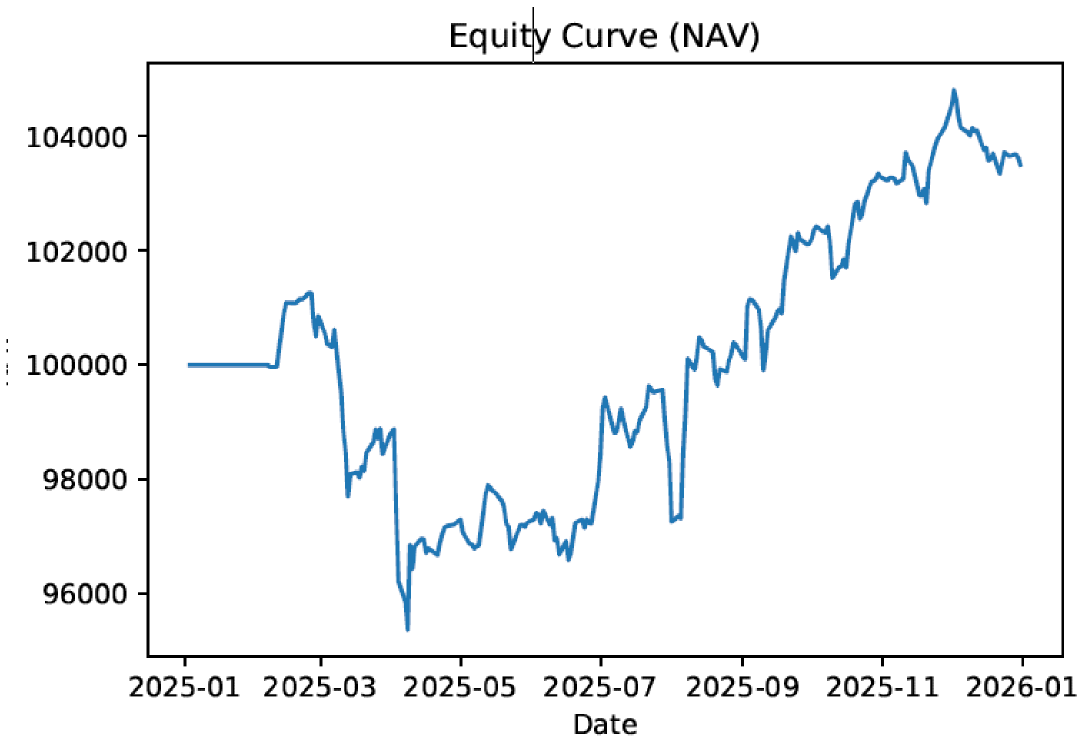
avg_turnover: 0.0072

total_cost: 358.8096

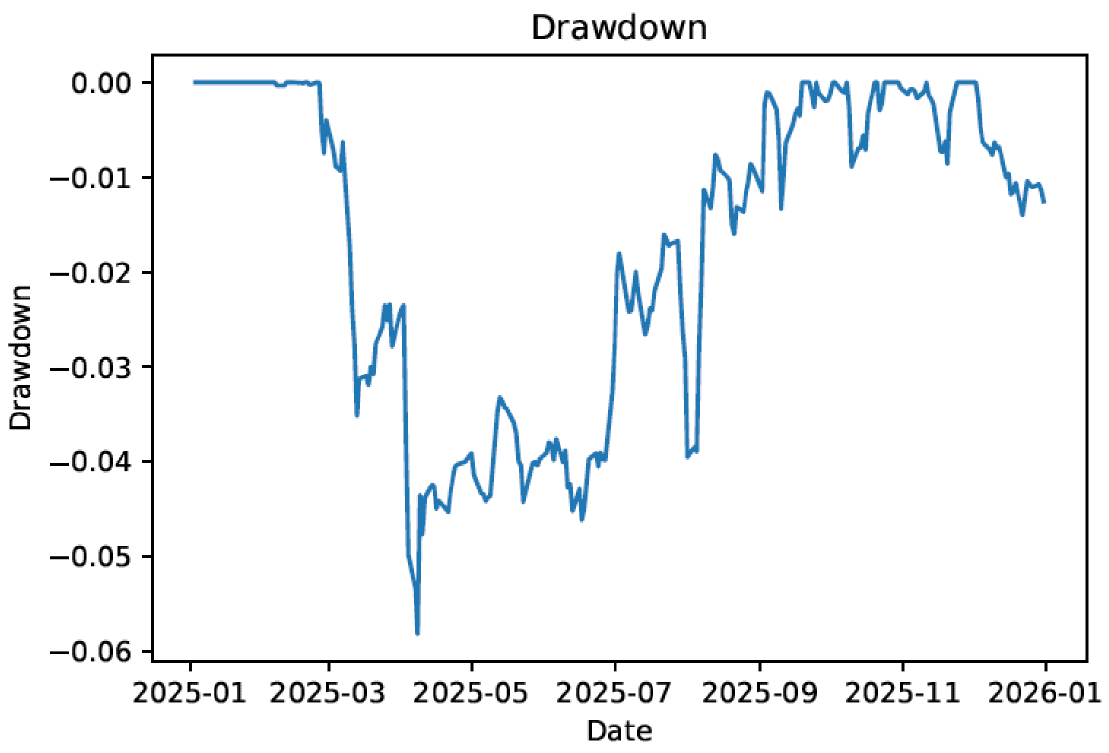
final_nav: 103499.2125

win_rate: 0.4819

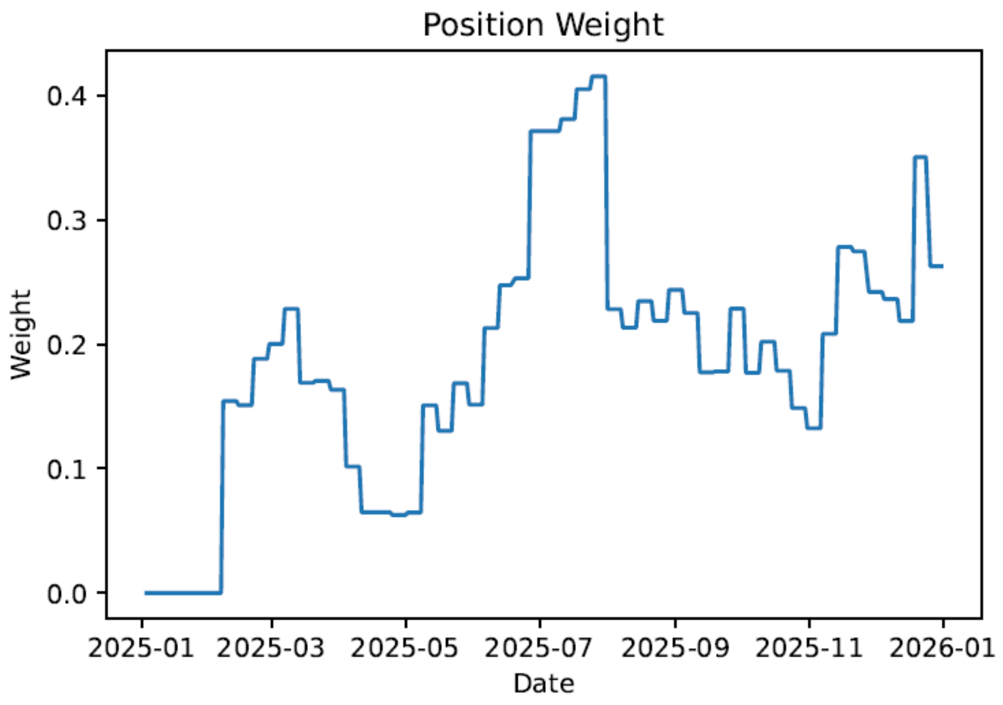
avg_daily_return: 0.0001

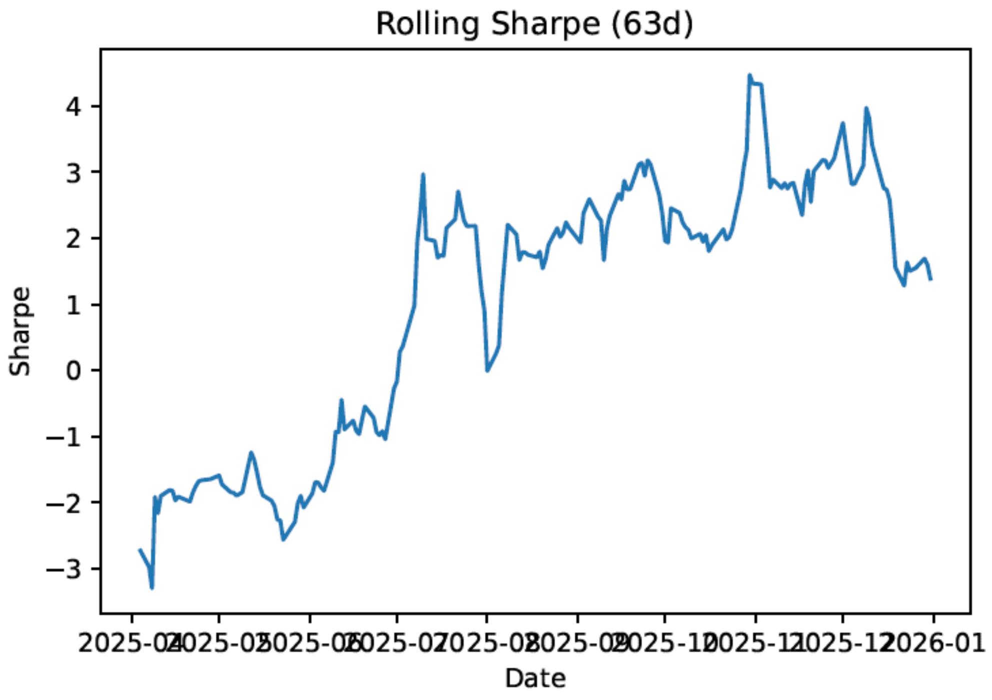


Equity Curve (NAV)
104000
102000
100000
98000
96000
2025-01
2025-03
2025-05
2025-07
2025-09
2025-11
2026-01
Date



Drawdown
0.00
−0.01
−0.02
−0.03
−0.04
−0.05
−0.06
Drawdown
2025-01
2025-03
2025-05
2025-07
2025-09
2025-11
2026-01
Date





Code and Other Contents: https://github.com/ray-islam/End-to-End-AI-Enhanced-Regime-Aware-Hybrid-Alpha-Strategy