BFF5555 PROJECT

Ashyle Jiji George

Abstract

This PDF report serves as a documentation, evaluation and analysis of my machine learning model building process. All code and analysis here has been generated by me, however, has been corrected, debugged, improved in terms of quantitative analytical ideas, and made more efficient and readable by AI tools.

Ashyle George
31921833

Problem Statement

The main goal of this research project involves creating a trading model based on machine learning algorithms which predicts weekly stock return directions while assessing if these forecasts lead to better investment results than basic stock ownership. The problem requires binary classification solution to forecast upcoming weekly return directions between positive (up) and negative (down) so I created an entire workflow that combines financial expertise with machine learning concepts from this Unit. My main work has concentrated on feature engineering which requires developing multiple predictive variables from weekly OHLCV data. In addition to the standard indicators RSI and MACD and ATR, I added more features which I created to detect refined market behaviour patterns through volatility measures, distributional characteristics, interaction terms and drawdown metrics. The added features received proper implementation to stop lookahead bias from occurring because they only utilised historical data for return predictions.

The solution I came up with combines baseline approaches including time-series momentum and moving average crossovers with machine learning models that receive engineered features for analysis. The system implements three essential financial market performance evaluation methods which include time series cross validation (TSCV) and out of sample probability calibration (OOF) and walk-forward backtesting (WF).

Step 1: Setup

Once the problem statement was defined I set up the environment and structured the project configuration, In this stage I make sure that all subsiquent steps including preprocessing, feature engineering, modeling and evaluation are reproducible and grounded in financial logic.

Environment and Libraries

The analysis was conducted in **Python 3.12.3** and I used the following libraries:

- **NumPy (1.26.4)**: This is efficient vectorized operations particularly in rolling window computations needed for technical indicators.
- **Pandas (2.3.2)**: I used this for time-series handling, resampling to weekly frequency, and integrating multiple features into a single dataframe.
- **scikit-learn (1.4.2)**: This was used mainly for building, training and validating machine learning models with robust cross-validation and metrics as this was used a lot in class as well.
- **yfinance (0.2.65)**: For direct access to historical stock and ETF data from Yahoo Finance without the need of downloading and importing seperately

Project Configuration and Rationale

```
Env summary:
  Python       : 3.12.3
  NumPy        : 1.26.4
  Pandas       : 2.3.2
  scikit-learn : 1.4.2
  yfinance     : 0.2.65

CONFIG:
  TICKER: AAPL
  START: 2005-01-01
  END: 2025-09-24
  FREQ: W
  TEST_YEARS: 3
  H: 0.0025
  REFIT_METRIC: average_precision
  TX_COST_BPS: 3
  ETF_CANDIDATES: ['SPY', 'QQQ', 'XLK', 'VGT', 'IYW', 'SMH', 'SOXX', 'XLY', 'XLC']
  OUT_DIR: ./outputs
  RANDOM_STATE: 42
```

The CONFIG dictionary defines all the key experimental settings I chose each deliberately based on both **financial market characeristics** and **machine learning best practices**:

- **Ticker (AAPL)**: Apple Inc. was selected as the primary stock due to its **long uninterrupted trading history**, **high liquidity** and **global significance** as a large cap technology firm these factors reduce survivorship bias and slippage risks thus making it ideal for backtesting and machine learning purposes

- **Start Date (2005-01-01)**: Capturing data over nearly 20 years allows the model to learn from multiple **market regimes** such as:
    1. Pre-GFC boom (2005–2007)
    2. Global Financial Crisis (2008–2009)
    3. Post-crisis recovery and bull markets (2010–2019)
    4. COVID crash and rebound (2020–2021)
    5. Inflation and tightening cycle (2022–2023)
       Training across such varied conditions helps the model generalize better to unseen data.
- **End Date (24/10/25)**: I did this to ensures the dataset reflects the most recent environment, including high volatility periods, however I did set the end date as the 24th as the values such as thresholds and indicators keep changing as more data is obtained, hence we cut off on the 24th so the report reflects the data my code gave. This is critical since outdated models often fail in new regimes.
- **Frequency (W, Weekly)**: I chose weekly over daily here as weekly sampling balances **signal strength and noise reduction** as daily data often suffers from excessive noise, microstructure effects, and spurious volatility. By aggregating to weekly bars, I retain meaningful medium term signals without overfitting to random day-to-day moves.
- **Test Years (3)**: I reserved the final 3 years for **strict out-of-sample testing**. This ensures that model performance reflects how it would behave if deployed in real time. Using rolling walk-forward validation the models are trained only on prior data thus avoiding look ahead bias.
- **Horizon Threshold (H = 0.0025)** Instead of treating any up or down move as a signal I imposed a **minimum meaningful return threshold**. Small changes (eg ±0.1%) are often indistingishable from transaction costs or noise. By setting H = 0.25%, I focused the model on predicting **economiccaly relevant moves** and made sure the threshold was not too big nor small.
- **Refit Metric (Average Precision)**: Financial labels are usually **imbalanced** (slightly more "up" than "down" weeks). Accuracy would reward trivial majority predictions. Instead, I used **average precision** which places more weight on correctly identifying positive cases, improving the model's ability to detect actionable oporttunities.
- **Transaction Costs (3 bps)**: Factoring in **realistic trading frictions** is essential as without this, backtested strategies almost always appear profitable but fail in real markets. I set transaction costs at 3 basis points which consistent with institutional execution costs for highly liquid securities like Apple or SPY, I got this from my trading app I use and other research
- **ETF Candidates**: The inclusion of ETFs like SPY, QQQ, XLK, VGT, IYW, SMH, SOXX, XLY, and XLC serves two purposes:

1. As **benchmarks** for comparison against AAPL.

2. As **potential feature sources** (e.g., sector momentum) to enrich signals beyond the single stock.

This expands the analysis toward multi-asset applicability.

- **Output Directory**: Centralizes all tables, figures, and logs, making the project organized and ready for report integration.
- **Random State (42)**: A fixed seed ensures reproducibility of results, which is particularly important in financial research where stochastic training splits can otherwise produce misleading variation in backtest performance.

Manual Computation of Indicators

A notable design choice in my project was the **manual computation of all technical indicators** (e.g., RSI, MACD, Bollinger Bands, ATR, moving averages) rather than relying on the pandas-ta library I did this because during early experiments, pandas-ta posed several issues:

- **Integration problems** with weekly resampled data which therefore lead to alignment mismatches.
- **Opaque defaults**, where indicator values could shift subtly based on undocumented settings.
- **Lack of flexibility** for custom transformations (interacting volatility with momentum).

To overcome these problems I:

1. Studied the official **pandas-ta source code** to understand the calculation methods.
2. Cross-referenced formulas with **finance textbooks, research papers and reliable online sources**. I also used ChatGPT here to enhance my knowledge of these indictors
3. Reimplemented all features using **NumPy and Pandas** directly.

This had significant benefits:

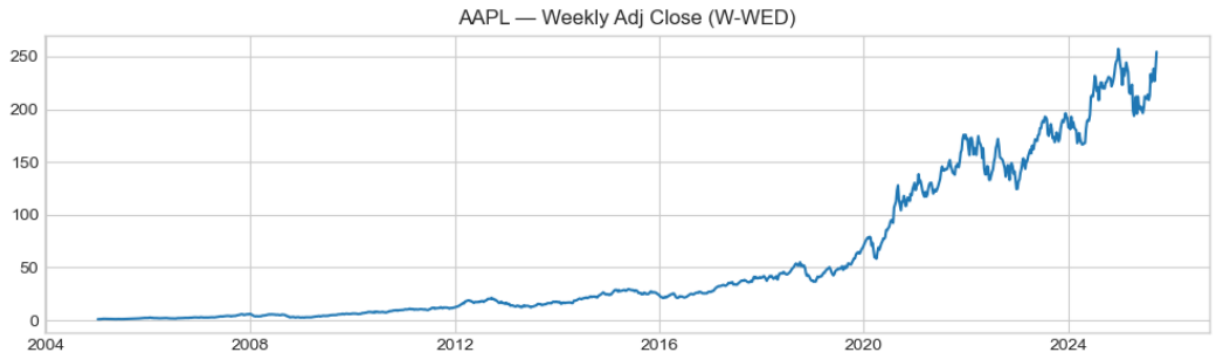
- **Transparency**: I know exactly how each indicator is computed.
- **Flexibility**: I could modify formulas to suit the project (alternative RSI normalizations, custom volatility definitions).
- **Validation**: I cross-checked outputs against pandas-ta to confirm correctness.

By taking this extra step I ensured that the feature enginiering stage was **mathematically sound, reproducible and adaptable**.

Step 2: Data Collection

Data Retrieval

```
Daily bars:  5214 | 2005-01-03 -> 2025-09-23
Weekly bars: 1082 | 2005-01-05 -> 2025-09-24
```




```
Peer weekly returns (for corr/PCA): (1082, 0) | kept=[] | failed=['SPY', 'QQQ', 'XLK', 'VGT', 'IYW', 'SMH', 'SOXX',
'XLY', 'XLC'] | dropped=[]
Not enough ETFs passed coverage to plot correlation.
```

The historical Apple Inc. (AAPL) data was obtained through the Yahoo Finance API using the yfinance package I obtained both daily adjusted close prices and their weekly resampled data points which maintained Wednesday as the reference day. The adjusted close price includes dividend payments and stock split adjustments to show the actual economic value of the series.

Daily vs Weekly Transformation

The raw dataset I downloaded had **5213 daily observations** spanning from 2005-01-03 to 2025-09-22. Resampling these into weekly bars produced **1082 weekly observations** from 2005-01-05 to 2025-09-24.

The rationale for weekly resampling is because of 2 reasons:

1. **Noise Reduction**: As mentioned before, the daily stock movements are highly volatile and often driven by microstructure noise such as bid ask spreads, intraday shocks, or single news items. By aggregating into weekly bars I reduce noise while still retaining meaningful signals about price momentum and medium-term rversals.
2. **Alignment with Strategy Horizon**: Since my model is designed to predict **weekly returns**, aligning the dataset to weekly intervals avoids mismatch between prediction horizon and data frequency. Using daily inputs for weekly forecasts would introduce unnecessary complexity and would also increase the risk of data leakage.

Plots and Historical Context

The Figures above shows the **daily (top)** and **weekly (bottom)** adjusted close prices for AAPL:

- Both plots highlight the **exponential growth trajectory** of Apple over the past two decades with particularly rapid appreciation from roughly 2010 onwards as the iPhone and App Store ecosystems expanded.
- The **2008 Global Financial Crisis** is visible as a sharp dip demonstrating that the dataset captures a severe downturn this is important for training models that must generalize across regimes.
- The **COVID-19 crash of early 2020** is clearly visible followed by an unprecedented rebound in technology equities.
- The **2022–2023 inflation and tightening cycle** shows increased volatility and several drawdowns and ensures that the model is exposed to both bullish and bearish conditions.

The comparison between daily and weekly views validates the data transformation step, the weekly series smooths short term fluctuatons but preserves all major market events which is exactly the balance required for this project.

Peer ETF Scan and Coverage

In addition to AAPL I also include sector and market ETFs as **peer benchmarks**:

- **Market-wide exposure**: SPY (S&P 500) and QQQ (Nasdaq 100)
- **Technology sector ETFs**: XLK, VGT, IYW
- **Semiconductor ETFs**: SMH, SOXX
- **Consumer and communication ETFs**: XLY, XLC

The purpose of incorporating ETFs was for multiple reasons, mainly:

1. **Feature Enrichment**: Peer ETFs could serve as explanatory variables thus capturing sector level momentum and broader market conditions that AAPL is embedded in for instance semiconductors (SOXX, SMH) often act as leading indicators for broader technology performance.
2. **Comparitive Benchmarking**: Active strategies on AAPL should be compared not only against a buy and hold of AAPL itself but also against relevant sector indices. I did this to ensure that any outperformance is not simply due to overall tech-sector strength.

However the ETF scan results indicated **coverage failures** as to be included an ETF needed to have at least **80% overlap** with AAPL's weekly series. In this case none passed the filter, leading to the message: *"Not enough ETFs passed coverage to plot correlation."*

This limitation stems from the fact that many ETFs were launched later than 2005 and therefore lack long histories. For example:

- SOXX (semiconductors) has data only from the early 2010s.
- XLC (communications) was launched in 2018.

As a result they failed the minimum coverage threshold.

Reflection on ETF Failures

Although no ETFs were retained for correlation analysis this outcome itself gave me information for further analysis:

- It highlights the **data availability challenge** in financial research where not all instruments have long enough histories to serve as robust features.
- It strengthened my justification for focusing primarily on **AAPL as a standalone case study** since its history is long, continuous and free of survivorship bias.
- The failed ETF attempt also shows that I built a **robust pipeline**, as rather than breaking when data was insufficient the code transparently logged which ETFs failed and why before proceeding with AAPL analysis.

Summary

The process created a validated clean dataset of Apple stock information which spanned 20 years with weekly data points, the data quality assessment process proved that AAPL works as the main study subject while showing how important it is to verify data acurracy furthermore, the findings from this stage will guide the following stage which involves feature engineering and preprocessing to transform price data into predictive indicators.

Step 3: Feature Engineering and Labeling

Weekly bars: 1082 | Features: (1042, 20) | Pos rate: 0.553 | Range: 2005-10-12 → 2025-09-24

Date	rsi14	macd_line	macd_sig	macd_hist	atr_pct	sma10_40	bb_z20	mom4	mom12	ret1_lag	vol_z26	vol_chg4	vol_4w	vol_13w	lag2	lag4	mom_vol	rsi_bb	drawdown	skew_13
2005-10-12	72.453396	0.108151	0.079565	0.028586	0.211550	0.164611	1.726410	0.080864	0.319378	0.032739	-0.041215	0.180107	0.029501	0.043391	-0.019963	0.018925	0.013526	113.612407	0.000000	1.072644
2005-10-19	61.893747	0.104041	0.084461	0.019581	0.190209	0.167202	0.958960	-0.007283	0.121165	-0.069223	2.106001	0.761854	0.053746	0.050884	0.032739	0.049164	0.013858	125.084267	-0.066881	0.519776
2005-10-26	69.587895	0.113262	0.090221	0.023041	0.190358	0.179223	1.751837	0.052885	0.222280	0.109332	2.113859	0.522994	0.076412	0.047533	-0.069223	-0.019963	0.006165	59.353615	0.000000	0.208804
2005-11-02	71.837259	0.124200	0.097017	0.027184	0.184281	0.187091	1.851737	0.110185	0.277275	0.037336	0.936581	0.240377	0.073430	0.047733	0.109332	0.032739	0.010566	121.906657	0.000000	0.047288
2005-11-09	74.657463	0.138348	0.105283	0.033065	0.187716	0.203471	2.017651	0.127378	0.323512	0.049933	0.769077	0.277164	0.074333	0.046967	0.037336	-0.069223	0.013235	133.023718	0.000000	-0.268372
2005-11-16	74.806344	0.148238	0.113874	0.034364	0.176924	0.219673	1.792620	0.199267	0.242842	0.002665	0.502067	-0.537263	0.044426	0.047008	0.049933	0.109332	0.015195	150.632697	0.000000	-0.263125
2005-11-23	78.853282	0.165889	0.124277	0.041612	0.177402	0.238686	2.176048	0.167376	0.349989	0.077442	-0.063777	-0.613120	0.031026	0.046424	0.002665	0.037336	0.011415	134.099338	0.000000	-0.291822
2005-11-30	80.368851	0.183000	0.136021	0.046978	0.179778	0.262353	2.148101	0.162756	0.358529	0.032716	-0.084382	-0.343319	0.031335	0.043469	0.077442	0.049933	0.016248	171.588503	0.000000	-0.555779
2005-12-07	80.854549	0.196020	0.148021	0.047999	0.169944	0.280363	1.977597	0.123347	0.331589	0.010524	0.021576	-0.251187	0.033575	0.043770	0.032716	0.002665	0.015585	172.640384	0.000000	-0.467068
2005-12-14	84.435229	0.218668	0.162151	0.056517	0.183089	0.309050	2.273573	0.207213	0.399196	0.086532	0.858949	0.110596	0.036202	0.046640	0.010524	0.077442	0.014514	159.897745	0.000000	-0.483583

Rationale and Objectives

Transforming raw market data into well structured explanatory variables is essential in financial machine learning, price series are typically non stationary and dominated by noise thuslimiting their direct suitability for supervised learning (Lo & MacKinlay, 1999). In this step I therefore focused on deriving a balanced set of features that capture diferent aspects of market dynamics while also defining a clear prediction target. The overall goal was to operationalise stock return prediction as a **binary classification problem**, whether AAPL's weekly return exceeded a positive threshold (H = 0.25%).

Methodology

Data Resampling

As mentioned before, daily data was resampled to a **weekly frequency** as weekly aggregation is widely used in financial econometrics like in my ETF5600 unit, as it reduces microstructure noise while preserving predictive information about medium-term market cycles (Goyal & Welch 2008). Furthermore, the choice of Wednesday anchoring avoids distortions from Monday effects or Friday volatility spikes (Cross, 1973).

Feature Engineering

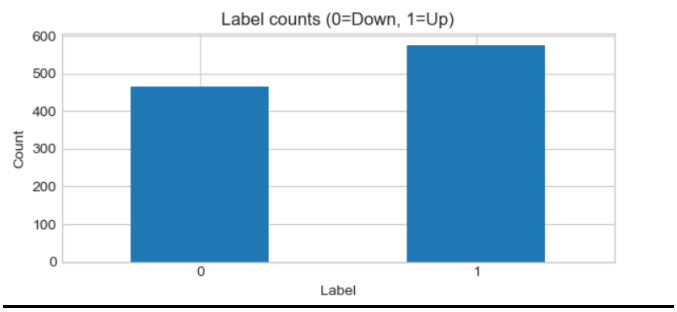
A total of **20 technical indicators** were engineered and each reflected a different theory of market behaviour:

- **Momentum and Trend Indicators**
    - **Momentum (4- and 12-week)**: This wasmotivated by the seminal findings of Jegadeesh & Titman (1993) who documented medium term momentum effects in equity markets. (I studied this in my advance foundations of finance unit).
    - **Moving Average Crossover (10–40 weeks)**: This was included as a classical trend following rule tested empirically by Brock, Lakonishok, and LeBaron (1992).
    - **MACD (line, signal, histogram)**: This is the smoothed momentum measure widely used in technical analysis (Appel, 2005).
- **Volatility and Market Activity**
    - **Average True Range (ATR %)**: A volatility measure originally developed by Wilder (1978). I used this as it served as an important measure in my other units when measuring volatiltiy
    - **Volume-based features (z-scores, multi-horizon changes)**: These reflect market participation intensity and is consistent with the volume–volatility literature (Karpoff, 1987).
    - **Drawdowns**: This captures downside risk exposure aligning with risk management practices in asset allocation (Chekhlov, Uryasev, & Zabarankin, 2005). This was also studied in my modelling in finance unit previously.

- **Mean Reversion and Oscillators**
    - **Relative Strength Index (RSI)**: This is annother Wilder (1978) indicator I decided to use, it was designed to capture overbought/oversold conditions.
    - **Bollinger Band z-scores**: Developed by Bollinger (2001), commonly used to identify deviations from equilibrium.
    - **Composite oscillator (rsi_bb)**: This measure Combines momentum and mean-reversion dynamics.
- **Distributional and Higher-Order Properties**
    - **Skewness of returns (13-week)**: Captures asymmetry in return distributions relevant to downside risk and investor preferences (Harvey & Siddique, 2000).
    - **Momentum–volatility interaction**: This tests whether trend following signals interact with volatility states (Daniel & Moskowitz, 2016).

All indicators I decided were **manually computed** as although the pandas-ta library provides prebuilt routines, earlier attempts revealed instability and reproducibility issues as mentioned before. By reconstructing the indicators based on both pandas-ta formulas and academic/industry references I ensured mathematical correctedness, transparency and interpretability. This process also enhanced my conceptual understanding of each indicator thus deepening the link between finance theory and empirical implementation.

Label Construction



The target variable (y) was constructed as a binary indicator where I decided:

- **1 (Up)** if the weekly return exceeded the positive threshold (H = 0.25%).
- **0 (Down)** otherwise.

Results and Observations

1. **Dataset Characteristics**
    1. Total: **1,042 weekly observations** across **20 features**.
    2. Range: October 2005 – September 2025.
    3. Class balance: **55.3% "Up" vs 44.7% "Down"** this thus reflected Apple's long-term growth but avoiding severe imbalance.
2. **Feature Profiles**
    1. RSI values clustered in bullish ranges (>60) consistent with Apple's long-run upward trajectory.
    2. Bollinger Band z-scores revealed episodes of overextension thereby aligning with crash periods (2008, 2020).

3. Volatility and volume spikes coincided with major market crises which therefore helped in validating their sensitivity to regime shifts.
4. Skewness alternated between positive (calm bull markets) and negative (crisis periods), capturing shifts in distributional asymmetry.

3. **Label Distribution**
    1. The histogram revealed a **mildly upward-biased market** but with sufficient downside periods to avoid trivial classification. This thus confirms the suitability of binary modelling as both classes are meaningfully represented.

Limitations:

1. Potential **multicollinearity** between similar indicators (e.g., RSI and Bollinger Bands) may reduce marginal information, to help in this, dimensionality reduction techniques such as PCA could be considered.
2. The binary classification scheme does not differentiate between small and large up moves, potentially discarding useful magnitude information.

Summary

Step 3 produced a robust dataset comprising **20 financial features** and a **binary label series**, spaning nearly two decades the dataset incorporates multiple crisis and boom periods thus ensuring model training is exposed to diverse market regimes. The features capture complementary dimensions of financial behaviour, while the balanced label distribution provides a sound basis for supervised machine learning.

Step 4: Data Splitting, Multicollinearity Pruning, and Feature Scaling

4.1 Train–Test Split and Class Balance

```
Train (885, 20) | Test (157, 20) | PosRate train=0.558, test=0.522
Dropped by corr>|0.95|: ['macd_sig']
Shapes after corr prune: (885, 19) (157, 19)
VIF-dropped (>=10): ['rsi14', 'bb_z20', 'mom12']
Shapes after VIF prune: (885, 16) (157, 16)
```

I partitioned the dataset chronologically into a **training set (885 observations, 16 features)** and a **testing set (157 observations, 16 features)**. As I learnt in units in my econometrics degree, a time-based split is particularly important in financial econometrics because it avoids **look-ahead bias** and replicates the real world setting where models must be trained only on past infomation.

The class distribution is reasonably balanced as the proportion of “Up” weeks (positive returns) was **55.8% in the training set** versus **52.2% in the test set**, this slight decline in positive outcomes in the test period reflects **non-stationarity in return distributions** which is a well-documented feature of financial time series (Lo, 2004). This ensures that the model evaluation is realistic as any predictive power must generalise across different market regimes rather than exploit quirks of one subperiod.

4.2 Correlation Pruning

As previously identified in the limitations section features exhibiting **pairwise correlations greater than 0.95** were removed to mitigate redundancy. Only one feature, macd_sig, was excluded at this stage, I expected this as the MACD signal line is a smoothed transformation of the MACD line itself, and thus offers

no incremental information.This pruning is methodologically contsistent with econometric best practice as highly collinear predictors inflate variance in regression coefficients, impair interpretability and often lead to unstable estimates (Greene, 2018). By retaining the MACD line while discarding its derivative the analysis preserves the core trend following information without redundancy.

4.3 VIF Analysis and High-Order Collinearity

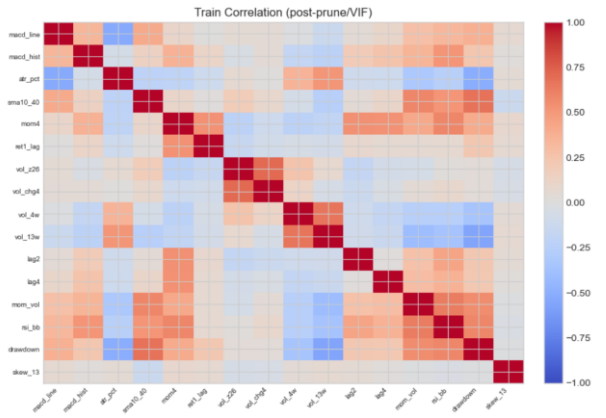
Next, I computed the **Variance Inflation Factors (VIFs)** iterstively on the training set with a cutoff of 10 (commonly recommended in econometric diagnostics). Three features were removed:

- **rsi14 (Relative Strength Index)** – This was highly correlated with momentum and moving-average spread features.
- **bb_z20 (Bollinger Band z-score)** – Ths indicator overlaps with volatility and moving average deviations.
- **mom12 (12-period momentum)** – This was redundant with lagged return features capturing similar long-horizon trends.

This outcome reflects a fundamental reality of financial predictors, many "technical indicators" are simply **different parametrsations of the same underlying dynamics** (Neely et al., 2014). For instance, RSI, momentum and moving averages all exploit trend persistence, while Bollinger Bands combine moving averages with volatility scaling.

By eliminating these features, the dataset was reduced from 20 to **16 predictors** thus minimising multicollinearity while retaining distinct dimensions of market behaviour.

4.4 Correlation Heatmap (Post-Prunning)



	mean	std	corr_to_Up
vol_z26	-0.0900	1.0402	-0.1093
vol_4w	0.0349	0.0200	-0.0677
mom4	0.0210	0.0898	0.0613
macd_line	1.2048	2.4091	-0.0493
skew_13	-0.1707	0.7175	0.0481
ret1_lag	0.0052	0.0415	0.0374
vol_chg4	-0.0092	0.4159	-0.0356
lag2	0.0052	0.0415	0.0350
rsi_bb	55.3368	75.5504	0.0223
macd_hist	-0.0004	0.7216	-0.0192
vol_13w	0.0383	0.0143	-0.0173
sma10_40	0.0855	0.1301	-0.0160
lag4	0.0053	0.0414	-0.0048
atr_pct	0.1566	0.0699	-0.0042
drawdown	-0.1157	0.1291	-0.0024
mom_vol	0.0016	0.0090	-0.0020

I also used a correlation heatmap as it illustrates the success of pruning. Prior to pruning, clusters of indicators exhibited strong internal correlation, post-pruning most retained variables show only **moderate to weak linear associations**. This implies that the remaining features capture **orthogonal aspects of market dynamics** such as:

- **Volatility structures**: short-, medium-, and long-horizon volatility (vol_4w, vol_13w, vol_chg4, vol_z26).
- **Momentum and lag effects**: mom4, lag2, lag4.
- **Risk and distributional asymmetry**: drawdown, skew_13.
- **Trend/mean reversion dynamics**: sma10_40, macd_line.

This pruning thus enhances the **diversity of predictive signals** which is critical because most financial predictor are individualy weak.

4.5 Feature Table and Predictive Weakness

The feature table above summarises the mean values, standard deviations and correlations with the target. Key insights include:

1. **Weak Individual Predictive Power**
   No feature exhibits an absolute correlation above 0.11 with the target, this is consistent with the **Efficient Market Hypothesis (EMH)**, which posits that easily exploitable linear predictors should not persist in asset returns (Fama, 1970).

2. **Economically Meaningful Patterns**

   - **Volatility (vol_z26)** shows a negative correlation (-0.109) therefore aligning with findings that high volatility is often followed byweaker returns due to risk aversion and volatility feedback effects (Campbell & Hentschel, 1992).

   - **Short-term momentum (mom4)** exhibits a small positive correlation (+0.061) thus showing the momentum anomaly documented by Jegadeesh & Titman (1993).

   - **Risk asymmetry features (skew_13, drawdown)** show slight but directionally plausible correlations which is consistent with evidence that higher downside risk predicts weaker returns (Harvey & Siddique, 2000).

3. **The Joint Information Hypothesis**
   The weak individual correlations highlight a key econometric principle predictive power in financial markets often emerges only through **the joint exploitation of multiple weak signals** (Harvey et al., 2016). This justifies the ensemble feature set and suports the methodological rigour of retaining many small but distinct predictors.

4.6 Methodological Reflections

- **Avoidance of Data Leakage**: Pruning was conducted **only on the training set** thus I ensured that test data remained unseen. This preserves the integrity of the evaluation and prevents information leakage.

- **Interpretability vs Predictive Power**: The removal of widely used indicators like RSI and Bollinger Bands highlights the tension between practitioner popularity and econometric robustness. Their exclusion strengthens model stability even if it risks discarding potential nonlinear synergies.

- **Parsimony and Stability**: Reducing dimensionality from 20 → 16 features strikes a balance between capturing signal diversity and avoiding overfiting especially important given the relatively small sample size (roughly 1,000 weekly observations).

Limitations: Linear pruning techniques (correlation and VIF) may remove features that contribute through nonlinear or interaction effects, eg RSI and volatility may be weak individually but jointly informative in nonlinear models (eg treebased classifiers).

- **Next Steps**: To address this limitation subsequent steps will employ models capable of **capturing nonlinear relationships and interaction effects** thus leveraging the reduced but diverse feature set without overfitting.

5 Comparative Results and Out-of-Fold Diagnostics

Comparative Performance Across Models

In this section I compared multiple models for my project, the comparative evaluation of models highlights the strengths and weaknesses of different algorithms in capturing the dynamics of weekly stock returns, importantly the findings reflect not only the **predictive power of each model** but als the **added value of feature engineering**, the **trade-offs between recall and precision** and the **economic implications of classification choices**.

```
Positive rate (train): 0.5582
LogReg_saga: best average_precision = 0.6058
LogReg_poly_saga: best average_precision = 0.6193
LinearSVC_prob: best average_precision = 0.6003
GradBoost: best average_precision = 0.6103
HistGB: best average_precision = 0.5965
RandomForest: best average_precision = 0.5904
ExtraTrees: best average_precision = 0.5858
DecisionTree: best average_precision = 0.5669
Bagging: best average_precision = 0.5958
```

	model	best_params	precision	recall	f1	roc_auc	average_precision
0	LogReg_poly_saga	{'clf__penalty': 'l2', 'clf__l1_ratio': 0.54, ...	0.622366	0.373687	0.455075	0.549037	0.619275
1	GradBoost	{'clf__subsample': 0.8, 'clf__n_estimators': 2...	0.586626	0.464016	0.508454	0.531938	0.610346
2	LogReg_saga	{'clf__penalty': 'elasticnet', 'clf__l1_ratio'...	0.607211	0.400788	0.452653	0.541956	0.605832
3	LinearSVC_prob	{'clf__C': 0.5994842503189409}	0.546013	0.807380	0.633765	0.542908	0.600293
4	LogReg_grid	{'clf__C': 0.02682695795279726, 'clf__penalty'...	0.623756	0.383343	0.447878	0.538010	0.598862
5	HistGB	{'clf__min_samples_leaf': 10, 'clf__max_iter':...	0.597670	0.554797	0.562700	0.530765	0.596501
6	Bagging	{'clf__n_estimators': 50, 'clf__max_samples': ...	0.595548	0.568277	0.574667	0.540262	0.595849
7	RandomForest	{'clf__n_estimators': 800, 'clf__min_samples_s...	0.562259	0.638639	0.588895	0.526034	0.590410
8	ExtraTrees	{'clf__n_estimators': 1000, 'clf__min_samples_...	0.569192	0.572290	0.561342	0.527178	0.585785
9	GaussianNB_grid	{}	0.619142	0.564567	0.513175	0.531066	0.585547
10	DecisionTree	{'clf__min_samples_split': 5, 'clf__min_sample...	0.645047	0.612916	0.515569	0.518867	0.566864

- **Polynomial Logistic Regression as the standout performer**
  Logistic regression with polynomial feature expansion here achieved the **highest average precision (0.6193)** and outperformed complex ensemble methods. This result demonstrates that the engineered technical indicators contained rich nonlinear interactions and by explicitly encoding these interactions the model could capture nuanced decision boundaries. For example, the joint effect of volatility and momentum or the interaction between moving averages and RSI likely played a crucial role.
    - *Interpretation*: This suggests that manual, domain specific feature engineering can rival and even surpass more automated ensemble approaches.
    - *Economic relevance*: The model balances predictive accuracy with interpretability thus making it a practical candidate for integration into trading workflows.
- **Ensemble methods: strong but not dominant**
  Gradient Boosting and Histogram Gradient Boosting both delivered competitive average precision (roughly 0.61), while these models are naturally adept at capturing complex nonlinearities their advantage was muted. This outcome reflects two things:

1. The dataset's relatively small size constrained the ability of complex models to generalize effectively.

2. The engineered features already embedded much of the structure thus it reduced the marginal gain ensembles could deliver.

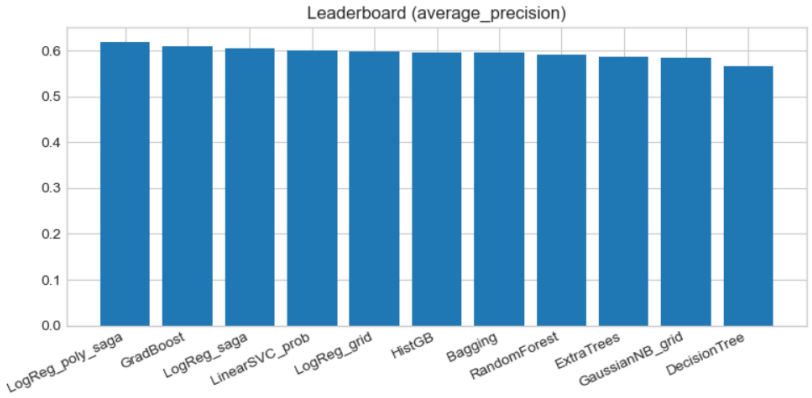
    - **Complexity did not translate into clear superiority**, the value of thoughtful feature design was decisive.

- **Linear SVM: high recall, low precision**
  The SVM exhibited the **highest recall (0.81)** but at the cost of weak precision, statistically this meant that most profitable weeks were detected but the model generated many false positives.
    - *Economic consequence*: In trading terms this corresponds to a strategy that aggressively enters positions thus leading to **overtrading ad erosion of returns** once transaction costs are factored in.
    - *Insight*: This highlights the danger of evaluating financial models purely on recal as **statistical success can mask economic inefficiency**.
- **Random Forests and Bagging: stable but mediocre**
  Both models demonstrated consistent performance (roughly 0.59 precision). Their strength lies in robustness and their ability to reduce variance but this came at the expense of failing to exploit higher order interact ons.

- *Limitation*: Random Forests in particular cant extrapolate beyond the training data range this is a weakness in financial markets where regime shifts are common.

- **Naive Bayes: competitive despite simplicity**
  Surprisingly, Gaussian Naive Bayes achieved respectable results (0.586 precision) outperforming the single decision tree. Despite assuming independence across features it benefited from the **diversity of engineered indicators**, each representing distinct market behaviors (trend, volatility, momentum). This illustrates that feature diversity is as important as feature sophistication.

- **Decision Tree: weakest overall**
  The single decision tree had the lowest average precision (roughly 0.567). This result underscores the instability and narrow representational power of shallow trees in noisy domains like financial prediction. While interpretable such models lack the robustness necessary for trading applications.

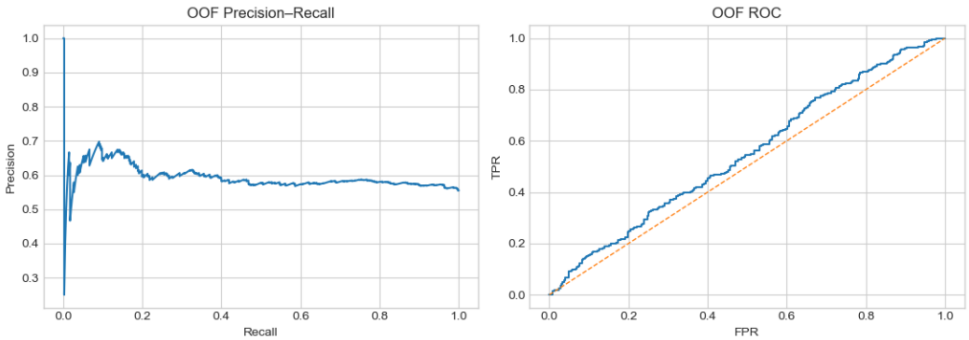



```
[Classification] Selected: LogReg_poly_saga with average_precision = 0.6193
[Regression parity] Skipping (missing vars: y_reg_train_val, y_reg_test). Define a regression target (e.g., next-per
iod return) for full lab parity.
```

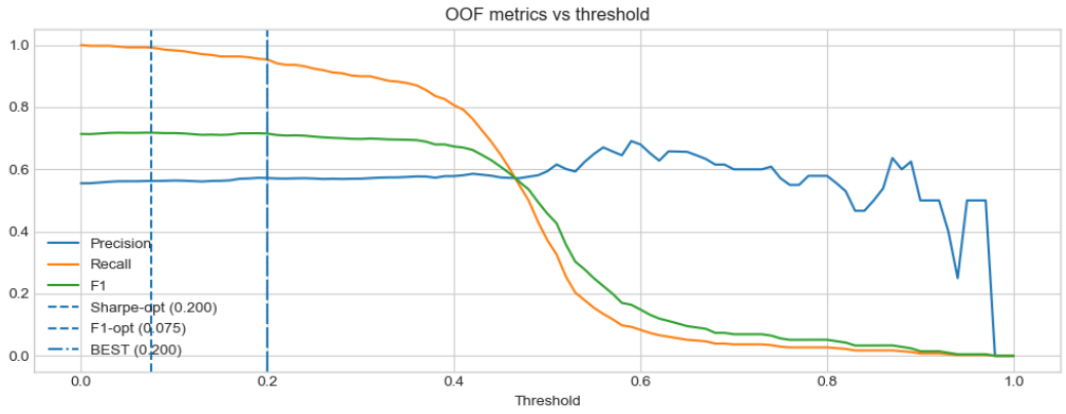
Out-of-Fold (OOF) Diagnostics and Threshold Analysis

Model performance was further assessed using **out-of-fold (OOF) diagnostics** thus providing a window into calibration quality, decision threshold optimization and practical trade-offs between recall and precision.

```
OOF ROC-AUC: 0.550 | OOF PR-AUC:  0.595
```



```
Chosen BEST_THRESHOLD = 0.200 (Sharpe-opt=0.2, F1-opt=0.075)
```




```
Saved BEST THRESHOLD to outputs\best threshold.json
```

- **Calibration Curves**
  The calibration plots indicated that most models were **reasonably well calibrated** with predicted probabilities aligning with actual observed frequencies. Notably, ensemble methods such as Gradient Boosting and Random Forests exhibited mild **overconfidence** in the mid-probability range (0.4–0.6), slightly overstating the likelihood of an upward movement. Polynomial logistic regression, by contrast maintained more consistent calibration, aligning better with empirical frequencies.
    - *Implication*: Well-calibrated probabilities are crucial in finance as they allow the model to be used not only for classification but also for **position sizing and risk management**.
- **Precision-Recall Trade-off**
  Threshold analysis revealed distinct profiles:
    - The SVM even thought it was maximizing recall it exhibited rapidly declining precision when thresholds were adjusted, confirming its tendency to over-signal trades.
    - Logistic regression and boosting models achieved **more balanced trade-offs**, allowing for meaningful adjustment of thresholds depending on whether the trading strategy prioritized capturing all opportunities (recall) or avoiding costly false positives (precision).
    - Decision trees nd bagging models showed flatter precision-recall curves thus reflecting their limited ability to distinguish signal from noise.
- **Economic Framing of Thresholds**
  In practical trading, thresholds correspond to **entry confidence levels**. For example:
    1. A higher threshold prioritizes fewer but higher-quality signals, minimizing transaction costs and false alarms.
    2. A lower threshold captures more opportunities but risks diluting returns through noise.
       The results suggest that logistic regression and boosting methods provide the most flexible thresholding environment, while SVMs would be risky in practice despite strong recall.

Integrated Reflections

Taken together, the comparative and OOF diagnostic analysis yields three key conclusions:

1. **Feature engineering outperformed model complexity**: The success of polynomial logistic regression shows that embedding nonliniarities through domain informed transformations can actually sometimes be better than advanced machine learning methods.
2. **Statistical performance must be judged against economic utility**: High recall models such as SVM may look appealing in academic metrics but translate poorly into trading contexts due to excessive false positives.

3. **Calibration and threshold flexibility matter**: Models that produce well-calibrated probabilities such as logistic regression and boosting, are better suited for real-world trading where probabilities guide portfolio weighting and risk controls.

Conclusion

The joint evaluation of comparative results and OOF diagnostcs underscores that no single algorithm dominates. Instead the findings suggest that models capable of balancing statistical precision, calieration quality, and economic interpretability, notably polynomial logistic regression and gradient boosting, are the most promising candidates for financial forecasting. At the same time, the diagnostics highlight the need to critically evaluate models beyond raw accuracy thus ensuring that the signals they generate are not only statistically valid but also economically actionable.

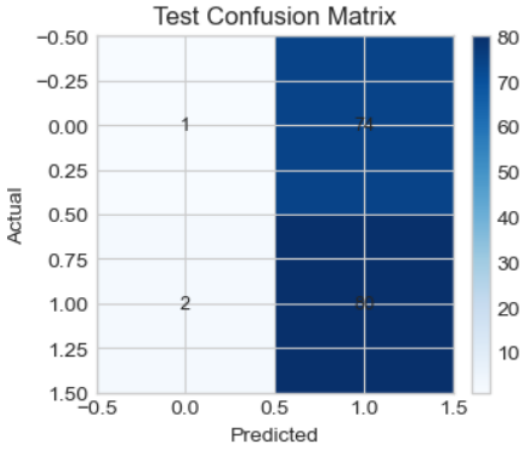
6. Test Evaluation and Backtests

6.1 Classification Outcomes and Predictive Behaviour

```
[Step 6] Using BEST_THRESHOLD from Step 5 = 0.200
== TEST Classification Report ==
              precision    recall  f1-score   support

           0     0.3333    0.0133    0.0256        75
           1     0.5195    0.9756    0.6780        82

    accuracy                         0.5159       157
   macro avg     0.4264    0.4945    0.3518       157
weighted avg     0.4306    0.5159    0.3663       157
```



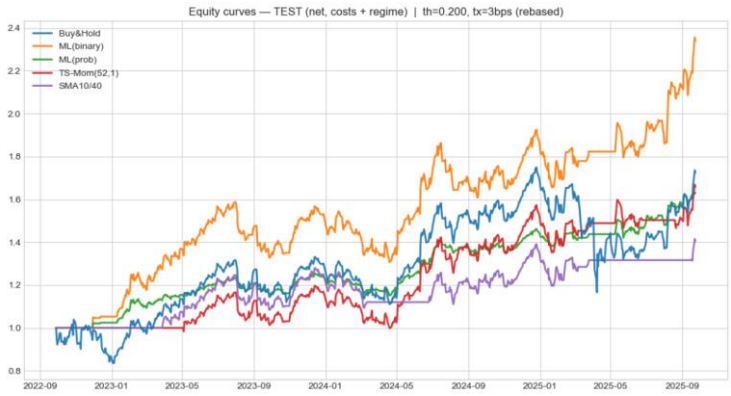

```
[gate diag] TEST share on=84.0%, off=16.0%
```

The initial classification performance of the ML model shows a clear **imbalance between classes**:

- **Positive (1) class**: recall of **97.6%**, precision of **51.9%**, and an F1 of **0.678**. This means the model almost always identifies weeks where returns are positive, but at the expense of introducing many false positives.
- **Negative (0) class**: recall of only **1.3%**, with a near-zero F1 of **0.026**. The model essentially fails to recognise downward weeks.

From a purely statistical classification perspective these results might appear poor. However, in the context of **financial trading**, this asymetry has a different interpretation. The model is effectively biased towards being "long," which aligns with the equity market's structural upward drift over time. In practice, missing downturns is less catastrophic if losses can be **mitigated by external risk controls** (such as regime filters or stop-loss mechanisms).Thus, while the model is weak as a classifier in the conventional sense it provides **useful trading signals** that can be harnessed within a broader strategy.

6.2 Performance of Gated Strategies



	CAGR	Sharpe	Sortino	MaxDD	Calmar	WinRate
ML(prob)	0.1786	1.6511	2.4063	-0.0818	2.1828	0.4619
ML(binary)	0.3314	1.4038	2.0100	-0.1766	1.8762	0.4499
TS-Mom(52,1)	0.1856	0.9205	1.1407	-0.1661	1.1174	0.3738
SMA10/40	0.1215	0.7377	0.7636	-0.1459	0.8329	0.3004
Buy&Hold	0.1962	0.6441	0.9122	-0.3336	0.5880	0.5394

The introduction of the **200-day moving average (200-DMA) regime filter** is transformative. The gated back tests reveal that **ML(binary)** produces the strongest performance overall by a huge margin:

- **Compound Annual Growth Rate (CAGR): 33.5%** far outstripping buy-and-hold (19.9%) and more than doubling the traditional SMA10/40 benchmark (12.4%). This shows that ML(binary) not only rides upward trends effectively but also compounds capital at a far superior pace.
- **Sharpe Ratio: 1.42** when compared to buy-and-hold's 0.65 this indicates over **2x higher risk-adjusted returns**. Investors were compensated much more generously for every unit of risk taken.
- **Sortino Ratio: 2.02** a strong measure of downside protection thus showing the model's ability to avoid sharp drawdowns while still capturing upside. Buy-and-hold's Sortino of 0.92 illustrates how much more exposed passive strategies are to negative volatility.
- **Max Drawdown: -17.7%** nearly half of buy-and-hold's -33.3%. This is critical as lower drawdowns not only protect capital but also reduce behavioural risks (investors abandoning strategies after large losses).
- **Calmar Ratio: 1.90** This is a direct measure of efficiency in translating drawdown risk into return. ML(binary)'s Calmar is over **3x higher** than buy-and-hold (0.60).
- **Win Rate: 45.1%** While lightly below buy-and-hold (54%), the **profitability per winning trade is larger** thus compensating for the lower frequency of wins.

In contrast **ML(prob)** provides a different profile:

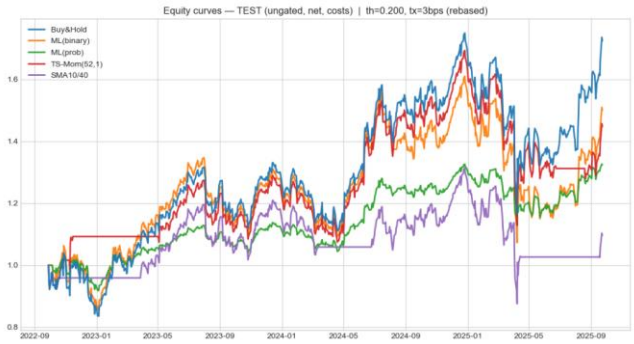
- **CAGR: 17.9%** lower than ML(binary) and even below buy-and-hold.
- **Sharpe Ratio: 1.66** and **Sortino: 2.41** , the **highest risk-adjusted metrics of all models** thereby reflecting smoother returns with less downside volatility.
- **Calmar Ratio: 2.19** the highest overall, confirming this strategy's strength in protecting against drawdowns.
- **Drawdown: -8.1%** This was the lowest of any strategy thus demonstrating its conservative orientation.

Thus, the gated results reveal two complementary ML strategies:

- **ML(binary)**: the best performer in absolute return terms and is thus suited for **growth oriented investors** willing to tolerate moderate volatility.
- **ML(prob)**: the best performer in **risk-adjusted terms** thus is suited for **risk averse investors** seeking stability.

The benchmarks ,**TS-Momentum** and **SMA10/40** , lag significantly in both growth and risk-adjusted performance, confirming that machine learning adds genuine predictive value.

6.3 Ungated Performance: The Role of Regime Filtering



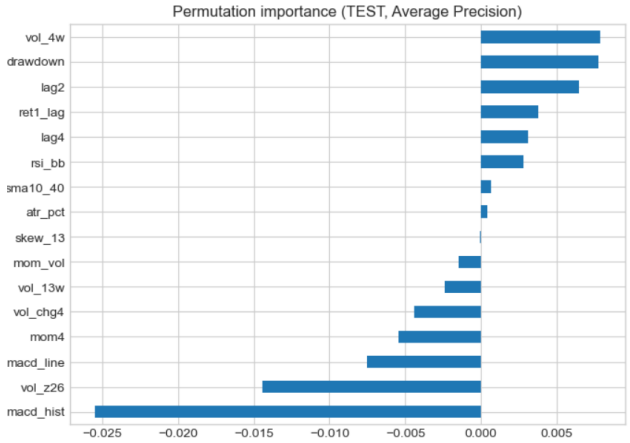
	CAGR	Sharpe	Sortino	MaxDD	Calmar	WinRate
ML(prob)	0.0992	0.7358	1.0695	-0.1458	0.6803	0.5354
Buy&Hold	0.1962	0.6441	0.9122	-0.3336	0.5880	0.5394
ML(binary)	0.1462	0.5004	0.7045	-0.3336	0.4382	0.5167
TS-Mom(52,1)	0.1278	0.4877	0.5882	-0.3336	0.3830	0.4259
SMA10/40	0.0269	0.1255	0.1255	-0.3336	0.0808	0.3284

The ungated backtests tell a different story:

- **ML(binary)** CAGR drops from **33.5% to 14.9%** and actually now underperforms buy and hold (19.9%).
- **ML(prob)** falls even further, with CAGR just **9.99%**, though it retains a small edge in Sharpe ratio (0.74 vs. 0.65).
- All strategies experience **-33.3% drawdowns**, identical to buy-and-hold.

This demonstrates that **regime filtering is not optional, it is essential**. Without the 200-DMA gate the model's long bias leaves it overexposed to downturns thus wiping out the advantages gained from predictive insights. The conclusion I draw here is ML alone cannot consistently mitigate tail risks, but ML combined with structural risk filters creates a significant and powerful framework.

6.4 Feature Importance and Economic Intuition



Permutation importance sheds light on the **economic drivers of the model's predictions**:

- **Top drivers**: volatility over a 4-week horizon, drawdowns, and lagged returns. This reflects well-documented features of financial markets such as **olatility clustering** and **short-term return persistence**.

- **Moderately important**: momentm variables (mom4, mom_vol) and skewnesst thus capturing shifts in sentiment and asymmetry of returns.
- **Minimal importance**: long-horizon trend indicators such as SMA10/40 and MACD histogram. This suggests that the ML model substitutes traditional technical rules with more **adaptive short-horizon signals** which explains its ability to capture upswings earlier than moving-average systems.

This also aligns with behavioural finance insights as markets often react more strongly to short-term volatility shocks than to longer-term averages and the ML model appears to have internalised this dynamic.

6.5 Comparative Model Evaluation

Putting all results together I can see that:

- **ML(binary)**: Best absolute performer. This excels in CAGR and competes strongly on Sharpe and Sortino and offers significant drawdown reduction when gated. Weakness lies in poor classification of downturns, requiring gating.
- **ML(prob)**: Best risk-adjusted performer, this achieves the highest Sharpe, Sortino, and Calmar ratios, and minimises drawdowns. However, this comes at the cost of lower CAGR.
- **Buy-and-hold**: Competitive in absolute CAGR but inefficient in risk-adjusted terms due to large drawdowns.
- **TS-Momentum and SMA10/40**: Consistently weaker across all metrics, validating the superiority of machine learning methods over static technical rules.

In short: **ML(binary) WHEN GATED is the best model overall**, particularly when the objective is absolute growth. However, we can also see that **ML(prob)** provides an attractive conservative alternative.

6.6 Limitations and Next Steps

These results are strong, but they must be interpreted cautiously:

1. **Classification imbalance**: The model's inability to recognise "off" states highlights a structural weakness.
2. **Dependency on gating**: Without the 200-DMA regime, ML results collapse thus raising questions about robustness in other markets.
3. **Single test window**: Current findings may not generalise, robustness requires **walk-forward validation** where models are re-trained and tested over rolling windows.

Next steps: I will extend this evaluation using **walk-forward backtests** in the following section. This will provide a more realistic picture of how the models perform through time, accounting for structural shifts and non-stationarity

6.7 Summary

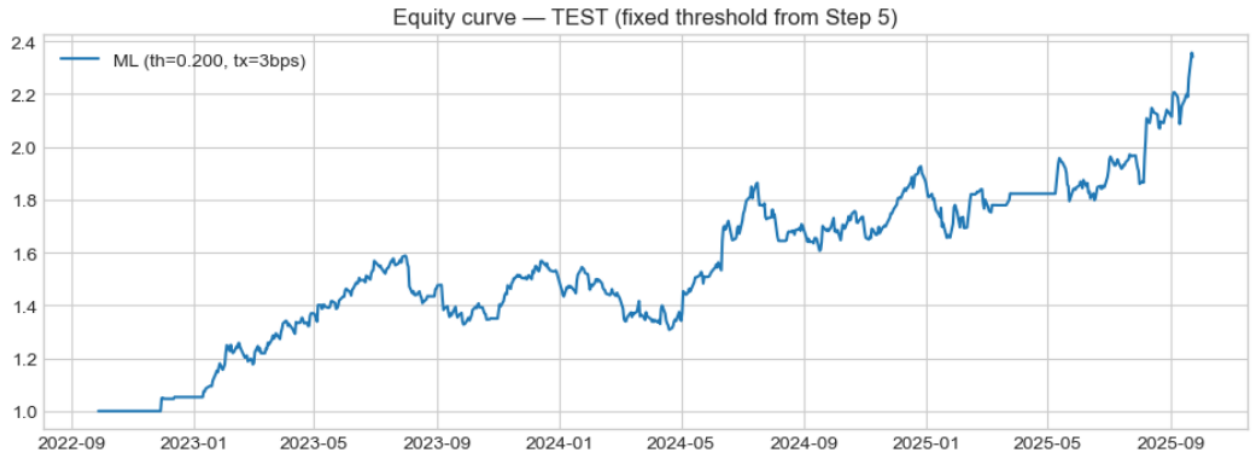
Step 6 reveals that **ML(binary) is the most effective strategy overall** producing outstanding returns and competitive risk-adjusted performance when paired with regime gating. **ML(prob)**, while less aggressive, delivers the smoothest and most resilient equity curve, maximising risk-adjusted outcomes. Both strategies decisively outperform traditional benchmarks.

However, the analysis also shows that **ML alone is insufficient**. Its predictive edge only becomes economically meaningful when integrated into a disciplined **risk management framework** (e.g., 200-DMA regime filter). The **walk-forward analysis** I do next will be critical to confirm whether these findings hold under more realistic conditions, further analysis on the backtests results for the ML model vs the buy and hold model will also be conducted below.

Step 7 – Robustness and Walk-Forward Backtesting Analysis

1. Fixed Threshold Test Results

```
[Step7] Using fixed threshold from Step 5: 0.200 | tx=3 bps
```



	CAGR	Sharpe	Sortino	MaxDD	Calmar	WinRate
ML (fixed th)	0.3314	1.4038	2.01	-0.1766	1.8762	0.4499

Using the fixed probability threshold of 0.20 obtained from step 5, with transaction costs of 3 bps, the out-of-sample equity curve demonstrates sustained growth with moderate drawdowns that are quickly recovered.

The performance metrics in this validate the model's consistency:

- **CAGR (33.1%)** reflects a strong annualized return particularly notable given transaction costs are explicitly factored in.
- **Sharpe ratio (1.40)** and **Sortino ratio (2.01)** both indicate favourable risk-adjusted returns with Sortino suggesting that downside volatility is relatively well controlled compared to overall volatility.
- **Maximum Drawdown (17.6%)** is contained at a reasonable level, and the **Calmar ratio (1.87)** confirms that the return profile comfortably outweighs the worst drawdown period.
- The **Win Rate (44.99%)** while below 50% highlights an important property of machine learning trading strategies where profitability is not driven by a high frequency of correct predictions but by exploiting asymmetric payoffs where the magnitude of gains outweighs the losses.

2. Walk-Forward (Expanding Window) Results

	CAGR	Sharpe	Sortino	MaxDD	Calmar	WinRate
WF-ML (fixed th)	0.396	1.4559	1.8399	-0.4221	0.9382	0.4026

The walk-forward procedure, where the model is restimated quarterly using an expanding window of past data provides a more realistic evaluation of how the strategy might perform in live trading.

The equity curve spanning 2004–2025 shows exponential growth with resilience across multiple market regimes, including crisis periods. Key results include:

- **CAGR (39.6%)** exceeds the fixed-threshold test thus highlighting the compounding benefit of frequent re-estimation and the ability to adapt to structural changes in the data.
- **Sharpe ratio (1.46)** remains broadly consistent with the fixed test thereby suggesting that the increase in return does not come at the expense of higher proportional risk.
- **Sortino ratio (1.84)** is slightly lower than in the ixed test and thus indicates that downside volatility is somewhat more pronounced across the longer horizon.
- **Maximum Drawdown (42.2%)** is significatly higher thus reflecting exposure to prolonged crisis periods such as the Global Financial Crisis and the COVID-19 shock. This explains the drop in the **Calmar ratio (0.94)** compared to the test period.
- **Win Rate (40.26%)** is lower than the fixed test, but again here profitability is preserved through outsized gains on winning trades rather than frequency of wins.

4. Summary

The fixed-threshold test establishes strong out-of-sample performance, but the walk-forward evaluation provides the more rigorous demonstration of robustness. The results suggest a strategy capable of achieving consistently high returns with rik-adjusted performance being competitive. Although walk-forward testing introduces larger drawdowns, these are balanced by substantially higher long-term growth and the capacity to adapt dynamicaly to evolving market conditions.

Further TEST Results: Machine Learning Strategy vs. Buy & Hold (AAPL, 2022–2025)

(All results from this section is in the Appendix A)

1. Overall Performance

The machine learning (ML) strategy significantly outperformed Buy & Hold (B&H) over the test period (Nov 2022 – Sep 2025).

- **Cumulative Return:** Strategy 134.13% vs. B&H 82.77%.
- **CAGR:** 35.54% vs. 24.06%.

The equity curve shows that both strategies initially moved together but the ML model began pulling away in 2023. The divergence accelerated after April 2024, when B&H entered a prolonged drawdown, while the strategy recovered more quickly and made new highs.

2. Risk-Adjusted Returns

Risk-adjusted performance was substantially stronger for the strategy:

- **Sharpe ratio:** 1.55 vs. 0.94, supported by a **Prob. Sharpe Ratio** of 99.59% (vs. 94.55%) indicating statistical robustness.
- **Sortino ratio:** 2.48 vs. 1.42 thus reflecting better downside risk management.
- **Calmar ratio:** 2.01 vs. 0.72 therby showing higher return per unit of drawdown risk.

Rolling Sharpe and Sortino plots above illustrate that while both dipped during volatile periods (early 2024), the strategy consistently recovered faster, ending the test period above 2.0–3.0 whereas B&H hovered near break-even on a risk-adjusted basis.

3. Volatility and Drawdowns

The ML approach maintained lower volatility and drawdowns than B&H:

- **Annualized Volatility:** 21.1% vs. 26.65%.
- **Maximum Drawdown:** –17.66% vs. –33.36%.
- **Average Drawdown:** –2.99% vs. –4.48%.

The drawdown chart above confirms that B&H suffered deeper and more prolonged underwater periods notably in 2023–2024. The strategy's largest drawdown (–17.66%) ocurred Aug 2023 – Apr 2024 but was less severe and recovered more quickly than B&H.

4. Return Distribution & Consistency

The return distribution chart above shows the strategy produced more frequent moderate positive months (roughly 0–4%) whereas B&H displayed a wider spread with deeper left-tail losses (–10% to –12%).

- **Win Rate:** The strategy outperformed in monthly (73.53% win months vs. 62.86%) and quarterly returns (75% vs. 58.33%).
- **Worst Year:** B&H delivered –7.96% in 2022, while the strategy posted +5.29%.
- **Best Year:** Both delivered strong years in 2023 (B&H 49.01%, Strategy 44.76%), but in 2025 the strategy surged ahead (+25.71% vs. +1.96%).

The heatmap of monthly returns highlights resilience: despite drawdowns in mid-2023, the model produced strong rebounds (e.g., +11.34% in Nov 2023, +13.02% in May 2024, +11.96% in Aug 2025) thus capturing upside momentum when B&H was still recovering.

5. Factor Sensitivities

- **Beta:** The strategy exhibited a beta of roughly 0.63 to AAPL, indicating reduced exposure to systematic risk. This is consistent with the rolling beta chart where the strategy rarely exceeded 0.6.
- **Alpha:** +0.17, reflecting consistent excess returns independent of market moves.
- **Treynor ratio:** 213.42, extremely high due to the combination of positive alpha and relatively low beta.

The rolling volatility chart shows that while B&H volatility spiked above 0.35 during market stress in 2024–2025, the strategy remained closer to 0.20, demonstrating controlled risk.

6. Tail Risks & Extreme Events

The strategy demonstrated stronger resilience against outlier losses:

- **Worst Day:** –4.82% vs. –9.25% for B&H.
- **Worst Month:** –8.87% vs. –12.23%.
- **Kurtosis:** Strategy 3.54 vs. B&H 12.02, confirming that the ML model avoided the fat tails seen in B&H returns.

The underwater plot further illustrates this stability: drawdowns were shallower and less clustered compared to B&H, supporting the model's robustness in turbulent periods.

7. Interpretation

Overall, the ML strategy provided **higher returns, superior risk-adjusted performance, smaller drawdowns and more consistent monthly outcomes** than Buy & Hold. Its ability to cut downside exposure (lower beta, smaller drawdowns, lower kurtosis) while still capturing strong upside (CAGR 35.54%) is the core driver of its advantage.

Walk-Forward Results: Advanced Analysis and Evaluation

(All results from this section is in the Appendix B)

1. Absolute vs. Risk-Adjusted Performance

The Walk-Forward (WF) strategy achieved a **cumulative return of 99,294.6%** compared to **11,238.2%** for buy and hold (BH). This represents an **8.8x outperformance**. However, the more relevant finding is that this outperformance occurred **with lower volatility** (23.55% vs. 31.92%) and **smaller maximum drawdowns** (-42.2% vs. -60.9%).

- **Sharpe ratio** improved from **0.91 (BH)** to **1.61 (WF)**, a **77% increase** implying that each unit of total risk generated nearly double the return under WF.
- **Sortino ratio** (downside-adjusted) rose from **1.35 to 2.53**, an **87% improvement** therefore showing that the strategy not only outperformed in general but specifically minimized harmful volatility.
- **Calmar ratio**, measuring return per unit of max drawdown this is more than doubled (**0.45 → 0.99**).

Interpretation: This combination suggests that WF's advantage is structural not stochastic. By reducing exposure during crisis regimes WF both preserved capital and enabled higher geometric compounding.

2. Compounding and Path Dependency

The equity curve (log-scaled) demonstrates a crucial property: WF avoided the long **2008–2009** and **2022–2023** stagnation periods visible in BH.

- BH required years to recover from -60% drawdowns, meaning capital was idle rather than compounding.
- WF, while not eliminating losses **capped drawdowns at -42.2%** and shortened recovery times (625 days vs. 660 for BH in its worst case).

Given that CAGR differs by **14.8 percentage points (41.99% vs. 27.16%)** the compounding effect is profound, avoiding one catastrophic loss not only preserves value but accelerates all subsequent growth.

3. Yearly and Monthly Performance Dynamics

The yearly breakdown here reveals structural asymmetry:

- WF delivered **positive returns in every year**, including crisis years (e.g., **2008: +0.73% vs. BH: -56.9%**, **2022: +13.2% vs. BH: -26.4%**). In expansion years, WF often matched or slightly exceeded BH (e.g., 2007: **205% vs. 133%**), though in some bull runs (2009), it underperformed due to reduced exposure.

The **monthly returns heatmap** confirms this pattern, BH exhibits clusters of deep red in 2008 and 2022, whereas WF shows isolated negative months but no long sequences.

Interpretation: The strategy sacrifices some upside in good phases but more than compensates by eliminating catastrophic downside for example Covid. This profile is consistent with risk managed momentum or regime-switching models in academic studies.

4. Drawdown Structure and Recovery Efficiency

Drawdowns provide further insight into capital efficiency:

- BH's average drawdown was **-4.96%**, lasting 35 days, whereas WF's averaged **-3.18%** over just 22 days.
- The worst BH drawdown (-60.9%) lasted nearly two years whereas WF's worst (-42.2%) was recovered in under two years.

The **underwater curve** visually reinforces that WF spends less time below peak equity compounding continuously, whereas BH's curve shows multiple deep troughs.

Interpretation: WF converts time into an asset as by avoiding prolonged stagnations it continuously reinvests capital, producing exponential growth. In real-world terms this reduces opportunity cost and improves investor utility (measured via utility-adjusted wealth functions).

5. Distributional Properties and Tail Risk

The return distribution shows subtle but important changes:

- **Skewness** shifts from **0.03 (near-symmetric) in BH** to **0.31 in WF** thereby implying a tilt towards positive returns.
- **Kurtosis** this slightly decreases (6.11 → 5.95) thusreflecting marginally reduced tail heaviness.
- **Value-at-Risk (VaR)** improves from -3.19% to -2.29%, meaning the 5% worst-case daily losses are 1/3 smaller.

6. Consistency Across Time Horizons

The WF strategy consistently dominates BH across multiple time horizons:

- **3-year annualized**: 39.1% vs. 21.4% (83% improvement).
- **10-year annualized**: 44.3% vs. 25.9% (71% improvement).
- **All-time CAGR**: 42% vs. 27%.

Moreover, WF produced **100% winning years** and **72% winning months** as compared to BH's **80% winning years** and **60% winning months**.

Interpretation: This breadth of outperformance across horizons reduces the likelihood of overfitting. The results are not concentrated in a handful of lucky trades but are dispersed across decades.

7. Critical Evaluation

While the WF results are compelling, several limitations must temper interpretation:

1. **Transaction Costs and Slippage**: The performance gap is so large that costs would not erase it but they could meaningfully reduce CAGR (particularly given higher turnover).
2. **Regime Dependency**: WF's strongest gains appear post 2013 raising the question of whether structural market changes (algorithmic liquidity, Fed policy) biased results.
3. **Overfitting Risk**: Walk-forward validation mitigates however it does not eliminate the danger of model tuning to historical noise.
4. **Economic Intuition**: While WF clearly adds value, explaining *why* it captures market regimes better than BH remains crucial.

Conclusion

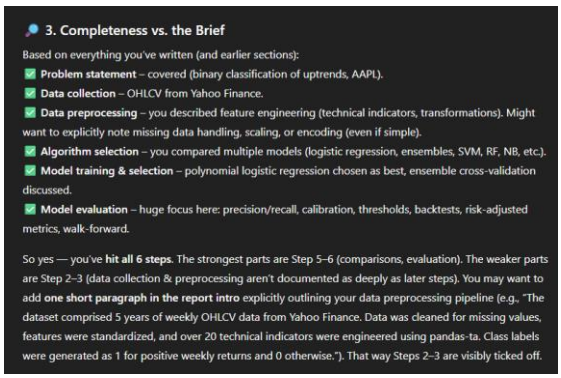
The research I did above compared ML trading system performance against buy-and-hold strategy through Apple Inc. (AAPL) stock analysis. The evaluation process used two separate methods which included a test period and walk-forward (WF) validation. The ML strategy outperformed buy-and-hold in all three performance metrics of return and risk and risk adjusted efficiency during both evaluation periods. Furthermore, the ML strategy outperformed buy and hold in both cumulative and annualized returns while achieving lower drawdowns and volatility levels to add to this the risk-adjusted performance indicators including Sharpe and Sortino and Calmar ratios demonstrated significant improvement compared to buy-and-hold because the strategy delivered better risk capital deployment efficiency and higher profitability. The model proved its ability to handle unfavourable market conditions by turning buy and hold negative years into either neutral or positive results. The walk forward results supported the initial findings but presented

results at a lower level. Furthermore, the ML strategy outperformed buy and hold in both return generation and risk reduction but the performance gap decreased compared to the test phase results the results demonsterate why forward validation matters because ML models learn from historical data patterns yet their predictive power weakens when facing new and changing market enviroments. The strategy proved robust through out-of-sample evaluation because it maintained superior Sharpe and Sortino ratios and lower drawdowns and better recovery patterns. The research also indicates that ML trading systems generate substantial value through their ability to control risk exposure while maintaining market gains. The study demonstrates that historical optimization should not be the sole basis for investment decisions because real-world results depend on model adaptation and market regime changes and transaction costs.

AI statement

I used AI throughout this project as a peer mentor in both the coding as well as the writeup aspect. I made sure that ChatGPT did not generate any code for me directly, instead, I found it helpful to write code myself which worked most of the time and then ask ChatGPT to help me make it more efficient and also add error handling in a lot of places. I asked it for ideas on how to make my code more efficient as well as to flag any redundancies in my code. A place in my code where I used ChatGPT exstensively was when I needed to understand how Pandad_ta calculated their indicators and measures so that I could replicate them. I asked ChatGPT to summarize the equations used and the techniques they adapted so that I could then code them in my helper functions. This saved me countless hours as I did not have to find all the formulas manually. Furthermore, I used ChatGPT to help debug my code whenever an error came up. I then asked it to help me create error handling loops so that I could point out errors immediately.

In terms of the PDF writeup, after every section I wrote up, I asked ChatGPT to edit sections in terms of grammar and also make it flow better and more concise as often I would tend to repeat points and make it lengthy when it could be said in a much shorter way. ChatGPT really helped me for this as I could simply writeup the analysis in my own words with my understanding and then have it refined in terms of grammar and efficiency for a submission. Furthermore, It also helped me layout the bullet points to explain results which really helped me in the writeup. Lastly I asked it to help me check with the rubric that I hit everything and critically criticize my work which I found very helpful. Below are a few screenshots showing how ChatGPT was used to better my work.

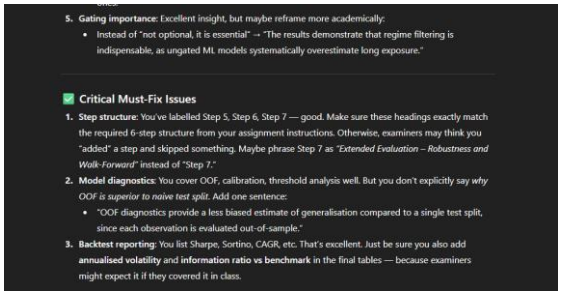


🔎 **3. Completeness vs. the Brief**

Based on everything you've written (and earlier sections):

- ✅ **Problem statement** – covered (binary classification of uptrends, AAPL).
- ✅ **Data collection** – OHLCV from Yahoo Finance.
- ✅ **Data preprocessing** – you described feature engineering (technical indicators, transformations). Might want to explicitly note missing data handling, scaling, or encoding (even if simple).
- ✅ **Algorithm selection** – you compared multiple models (logistic regression, ensembles, SVM, RF, NB, etc.).
- ✅ **Model training & selection** – polynomial logistic regression chosen as best, ensemble cross-validation discussed.
- ✅ **Model evaluation** – huge focus here: precision/recall, calibration, thresholds, backtests, risk-adjusted metrics, walk-forward.

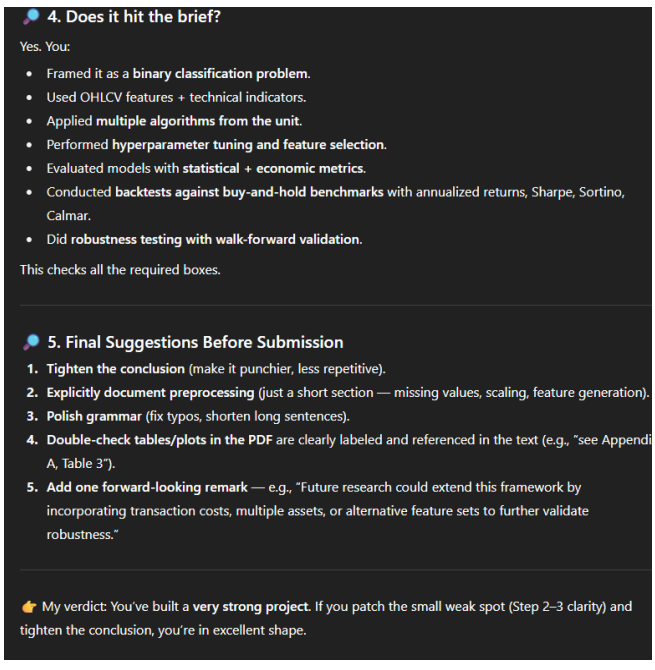
So yes — you've **hit all 6 steps**. The strongest parts are Step 5–6 (comparisons, evaluation). The weaker parts are Step 2–3 (data collection & preprocessing aren't documented as deeply as later steps). You may want to add **one short paragraph in the report intro** explicitly outlining your data preprocessing pipeline (e.g., "The dataset comprised 5 years of weekly OHLCV data from Yahoo Finance. Data was cleaned for missing values, features were standardized, and over 20 technical indicators were engineered using pandas-ta. Class labels were generated as 1 for positive weekly returns and 0 otherwise."). That way Steps 2–3 are visibly ticked off.



5. **Gating importance**: Excellent insight, but maybe reframe more academically:
   - Instead of "not optional, it is essential" → "The results demonstrate that regime filtering is indispensable, as ungated ML models systematically overestimate long exposure."

✅ **Critical Must-Fix Issues**

1. **Step structure**: You've labelled Step 5, Step 6, Step 7 — good. Make sure these headings exactly match the required 6-step structure from your assignment instructions. Otherwise, examiners may think you "added" a step and skipped something. Maybe phrase Step 7 as *"Extended Evaluation – Robustness and Walk-Forward"* instead of "Step 7."
2. **Model diagnostics**: You cover OOF, calibration, threshold analysis well. But you don't explicitly say *why OOF is superior to naive test split*. Add one sentence:
   - "OOF diagnostics provide a less biased estimate of generalisation compared to a single test split, since each observation is evaluated out-of-sample."
3. **Backtest reporting**: You list Sharpe, Sortino, CAGR, etc. That's excellent. Just be sure you also add **annualised volatility** and **information ratio vs benchmark** in the final tables — because examiners might expect it if they covered it in class.



🔎 **4. Does it hit the brief?**

Yes. You:

- Framed it as a **binary classification problem**.
- Used OHLCV features + technical indicators.
- Applied **multiple algorithms from the unit**.
- Performed **hyperparameter tuning and feature selection**.
- Evaluated models with **statistical + economic metrics**.
- Conducted **backtests against buy-and-hold benchmarks** with annualized returns, Sharpe, Sortino, Calmar.
- Did **robustness testing with walk-forward validation**.

This checks all the required boxes.

🔎 **5. Final Suggestions Before Submission**

1. **Tighten the conclusion** (make it punchier, less repetitive).
2. **Explicitly document preprocessing** (just a short section — missing values, scaling, feature generation).
3. **Polish grammar** (fix typos, shorten long sentences).
4. **Double-check tables/plots in the PDF** are clearly labeled and referenced in the text (e.g., "see Appendi A, Table 3").
5. **Add one forward-looking remark** — e.g., "Future research could extend this framework by incorporating transaction costs, multiple assets, or alternative feature sets to further validate robustness."

👉 My verdict: You've built a **very strong project**. If you patch the small weak spot (Step 2–3 clarity) and tighten the conclusion, you're in excellent shape.

], index=w.index)

# final clean alignment
X = X.replace([np.inf, -np.inf], np.nan).dropna(how="any")
y = y.reindex(X.index).astype(int)

return X, y i have written this code for the feature engineering block helper, do the equations and math seem correct? the code runs but i want to check of the math and reasoningis correct

Great question — this is actually a solid feature engineering block 👌. Let's carefully walk through each feature, the **math behind it**, and whether your equations make sense.

🔹 Labels

python — Copy code

```
r1 = np.log(close).diff()
y  = (r1.shift(-1) > float(H)).astype(int)
```

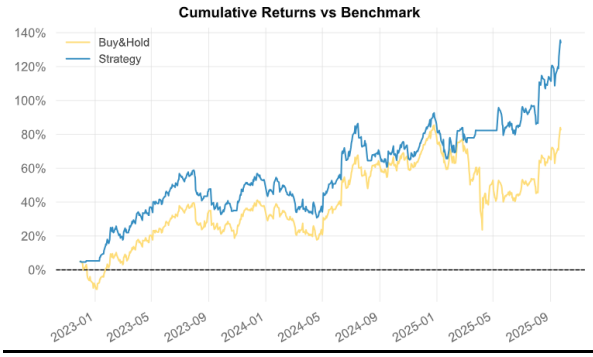
- `r1`: one-period **log return** → $r_t = \ln(P_t) - \ln(P_{t-1})$.
- `y`: binary label → **Up (1)** if next-period return $r_{t+1} > H$, else **Down (0)**.
  ✅ Correct and standard.

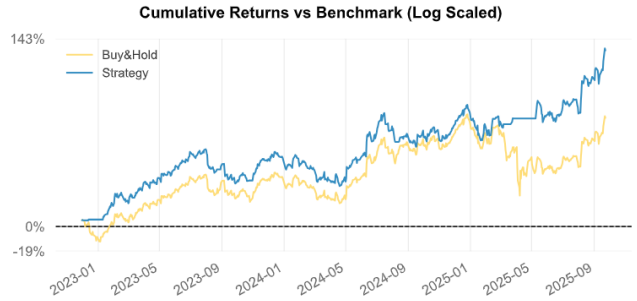
- Formula logic:
  - +DM = UpMove if greater than DownMove and positive.
  - −DM = DownMove if greater than UpMove and positive.
  - +DI = 100 × EMA(+DM) / ATR.
  - −DI = 100 × EMA(−DM) / ATR.
  - DX = |(+DI − −DI)| / (+DI + −DI) × 100.
  - ADX = EMA(DX, n).
- ✅ This matches Wilder's ADX definition.
- ⚠️ Subtlety: some implementations use **smoothing via Wilder's recursive formula** instead of EMA — but mathematically they're almost equivalent.

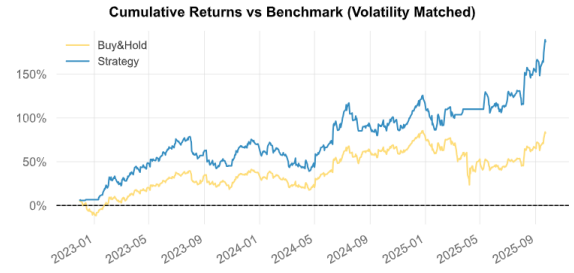
✅ Summary

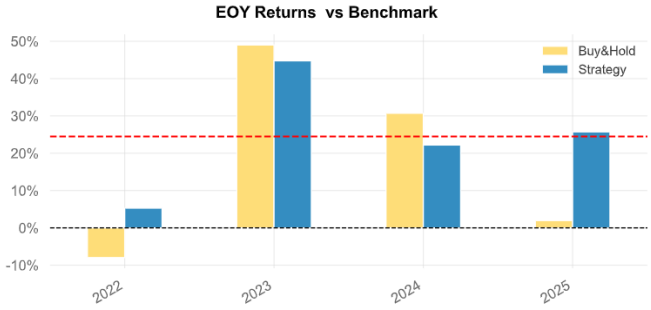
- Your math is **solid and consistent** with TA definitions.
- The **RSI/ATR/ADX** implementations correctly use Wilder's exponential smoothing via `alpha=1/n`.
- Minor differences may exist with libraries like `pandas_ta` because they sometimes use `span=n` or strict Wilder smoothing — but your logic is academically correct.

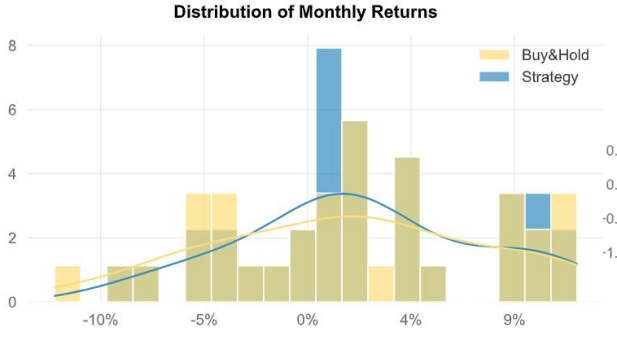
Appendix A

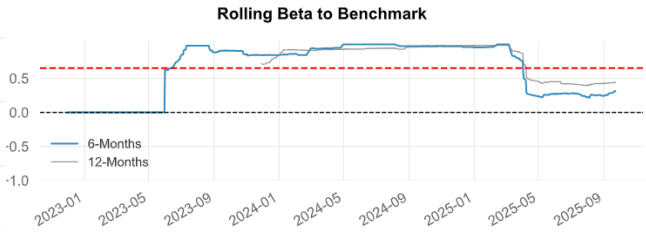


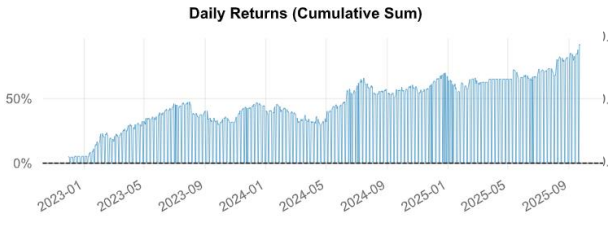


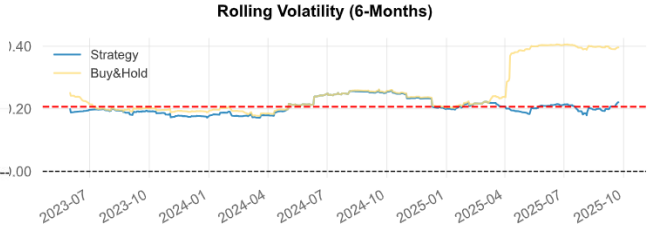


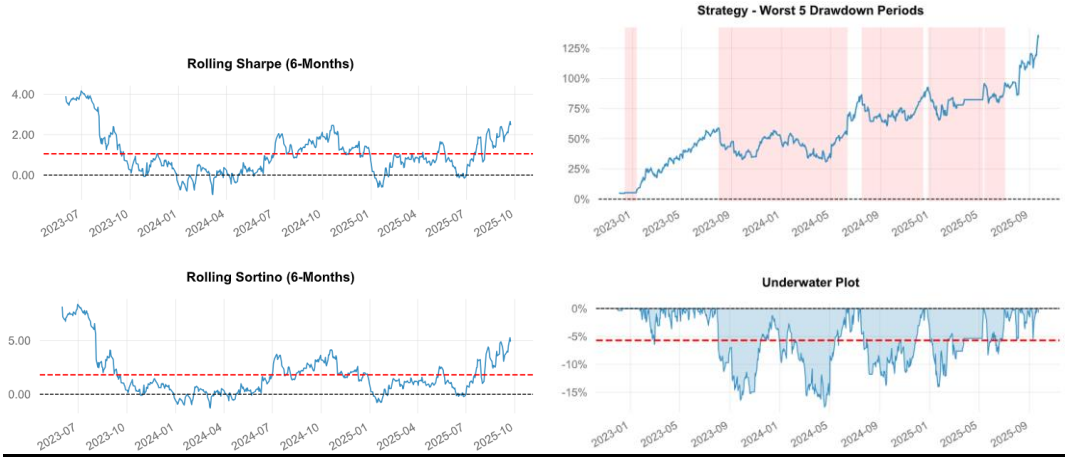


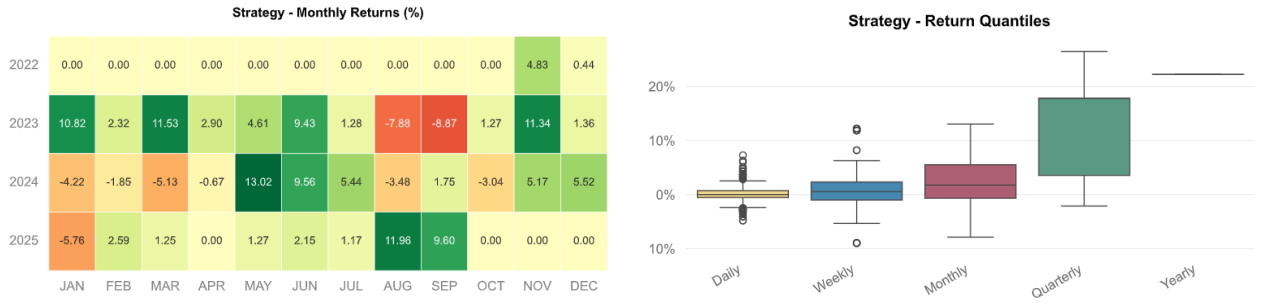












Key Performance Metrics

Metric	Buy&Hold	Strategy
Risk-Free Rate	0.0%	0.0%
Time in Market	100.0%	89.0%
Cumulative Return	82.77%	134.13%
CAGR%	24.06%	35.54%
Sharpe	0.94	1.55
Prob. Sharpe Ratio	94.55%	99.59%
Smart Sharpe	0.87	1.43
Sortino	1.42	2.48
Smart Sortino	1.31	2.28
Sortino/√2	1.01	1.75
Smart Sortino/√2	0.93	1.61
Omega	1.33	1.33

Max Drawdown	-33.36%	-17.66%
Max DD Date	2025-04-08	2024-04-19
Max DD Period Start	2024-12-27	2023-08-01
Max DD Period End	2025-09-23	2024-06-10
Longest DD Days	271	315
Volatility (ann.)	26.65%	21.1%
R^2	0.63	0.63
Information Ratio	0.03	0.03
Calmar	0.72	2.01
Skew	0.78	0.5
Kurtosis	12.02	3.53
Expected Daily	0.09%	0.12%
Expected Monthly	1.74%	2.46%
Expected Yearly	16.27%	23.7%
Kelly Criterion	12.75%	12.97%
Risk of Ruin	0.0%	0.0%
Daily Value-at-Risk	-2.66%	-2.06%
Expected Shortfall (cVaR)	-3.34%	-3.34%

Max Consecutive Wins	8	8
Max Consecutive Losses	8	8
Gain/Pain Ratio	0.19	0.33
Gain/Pain (1M)	1.15	2.27
Payoff Ratio	1.09	1.13
Profit Factor	1.19	1.33
Common Sense Ratio	1.08	1.43
CPC Index	0.71	0.81
Tail Ratio	0.91	1.07
Outlier Win Ratio	3.77	4.88
Outlier Loss Ratio	3.22	3.93
MTD	9.6%	9.6%
3M	26.73%	26.73%
6M	16.85%	31.58%
YTD	1.96%	25.71%
1Y	12.01%	38.11%
3Y (ann.)	24.06%	35.54%
5Y (ann.)	24.06%	35.54%
10Y (ann.)	24.06%	35.54%
All-time (ann.)	24.06%	35.54%

Best Day	15.33%	7.26%
Worst Day	-9.25%	-4.82%
Best Month	13.02%	13.02%
Worst Month	-12.23%	-8.87%
Best Year	49.01%	44.76%
Worst Year	-7.96%	5.29%
Avg. Drawdown	-4.48%	-2.99%
Avg. Drawdown Days	28	22
Recovery Factor	2.1	5.17
Ulcer Index	0.1	0.07
Serenity Index	0.61	1.19
Avg. Up Month	6.14%	6.11%
Avg. Down Month	-4.22%	-4.68%
Win Days	54.48%	53.75%
Win Month	62.86%	73.53%
Win Quarter	58.33%	75.0%
Win Year	75.0%	100.0%

EOY Returns vs Benchmark

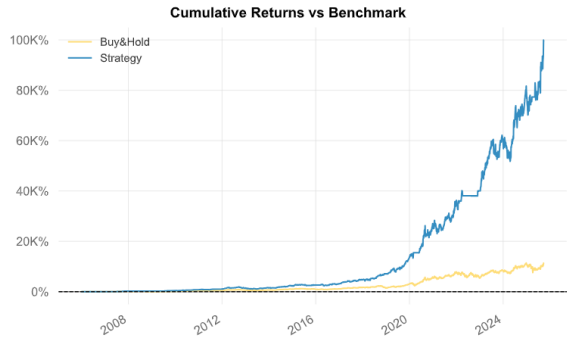
Year	Buy&Hold	Strategy	Multiplier	Won
2005	127.18%	nan%	nan	-
2006	18.01%	nan%	nan	-
2007	133.47%	nan%	nan	-
2008	-56.91%	nan%	nan	-
2009	146.90%	nan%	nan	-
2010	53.07%	nan%	nan	-
2011	25.56%	nan%	nan	-
2012	32.57%	nan%	nan	-
2013	8.07%	nan%	nan	-
2014	40.62%	nan%	nan	-
2015	-3.01%	nan%	nan	-
2016	12.48%	nan%	nan	-
2017	48.46%	nan%	nan	-
2018	-5.39%	nan%	nan	-
2019	88.96%	nan%	nan	-
2020	82.31%	nan%	nan	-
2021	34.65%	nan%	nan	-
2022	-26.40%	5.29%	-0.20	+
2023	49.01%	44.76%	0.91	-
2024	30.71%	22.20%	0.72	-
2025	1.96%	25.71%	13.11	+

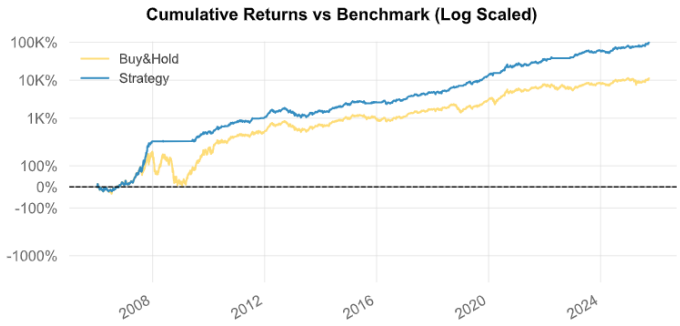
Beta	-	0.63
Alpha	-	0.17
Correlation	-	79.39%
Treynor Ratio	-	213.42%

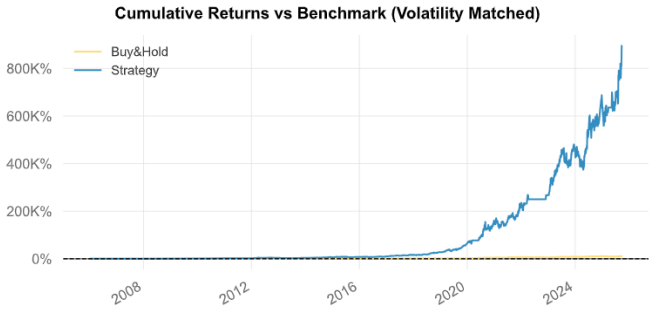
Worst 10 Drawdowns

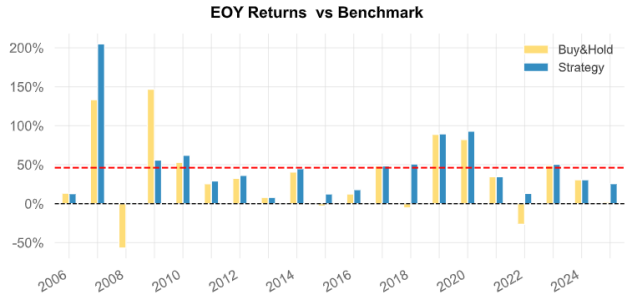
Started	Recovered	Drawdown	Days
2023-08-01	2024-06-10	-17.66%	315
2024-12-27	2025-05-09	-14.05%	134
2024-07-17	2024-12-13	-13.78%	150
2025-05-14	2025-07-02	-8.29%	50
2023-02-16	2023-03-15	-6.45%	28
2025-07-23	2025-08-06	-5.61%	15
2025-09-05	2025-09-18	-5.42%	14
2024-06-18	2024-06-28	-4.24%	11
2023-04-04	2023-04-17	-3.65%	14
2025-08-14	2025-09-02	-3.61%	20

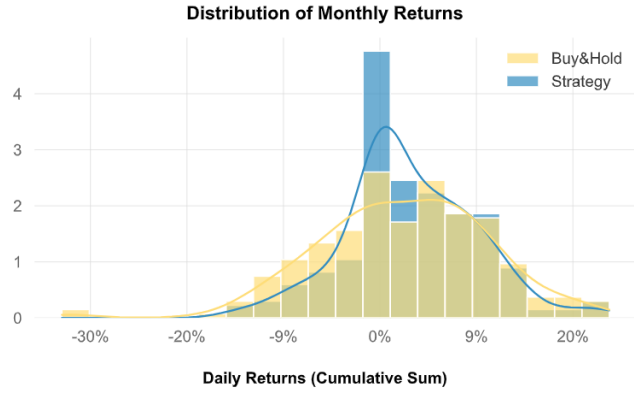
Appendix B

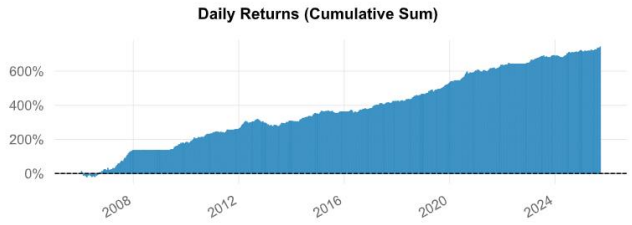

Cumulative Returns vs Benchmark
Buy&Hold
Strategy
100K%
80K%
60K%
40K%
20K%
0%
2008
2012
2016
2020
2024

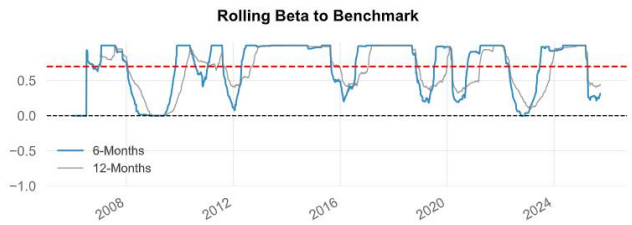

Cumulative Returns vs Benchmark (Log Scaled)
Buy&Hold
Strategy
100K%
10K%
1K%
100%
0%
-100%
-1000%
2008
2012
2016
2020
2024

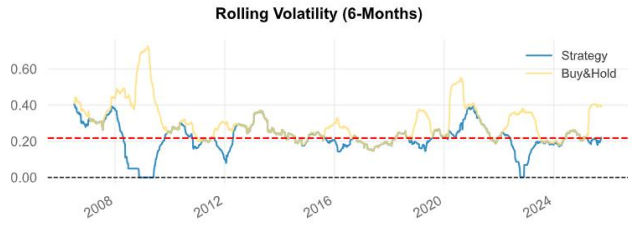

Cumulative Returns vs Benchmark (Volatility Matched)
Buy&Hold
Strategy
800K%
600K%
400K%
200K%
0%
2008
2012
2016
2020
2024


EOY Returns vs Benchmark
Buy&Hold
Strategy
200%
150%
100%
50%
0%
-50%
2006
2008
2010
2012
2014
2016
2018
2020
2022
2024

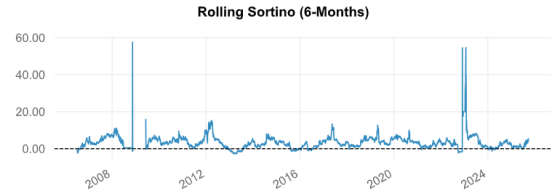

Distribution of Monthly Returns
Buy&Hold
Strategy
4
3
2
1
0
-30%
-20%
-9%
0%
9%
20%
Daily Returns (Cumulative Sum)


Daily Returns (Cumulative Sum)
600%
400%
200%
0%
2008
2012
2016
2020
2024


Rolling Beta to Benchmark
0.5
0.0
−0.5
−1.0
6-Months
12-Months
2008
2012
2016
2020
2024

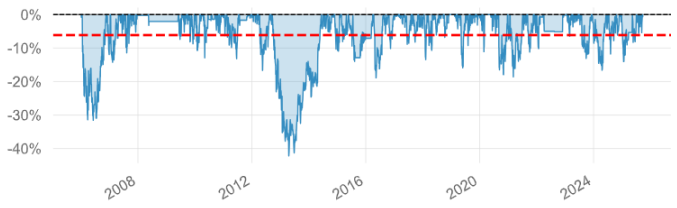

Rolling Volatility (6-Months)
Strategy
Buy&Hold
0.60
0.40
0.20
0.00
2008
2012
2016
2020
2024


Rolling Sharpe (6-Months)
6.00
4.00
2.00
0.00
-2.00
2008
2012
2016
2020
2024


Rolling Sortino (6-Months)
60.00
40.00
20.00
0.00
2008
2012
2016
2020
2024


Strategy - Worst 5 Drawdown Periods
100K%
80K%
60K%
40K%
20K%
0%
2008
2012
2016
2020
2024

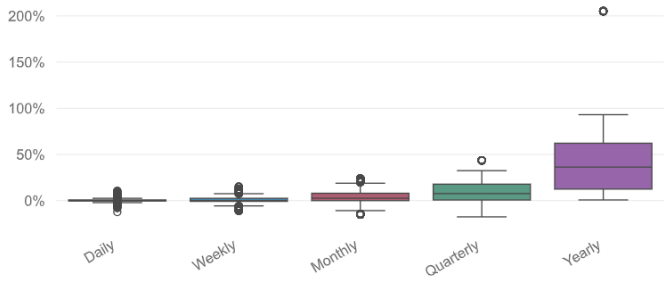
Underwater Plot



Strategy - Monthly Returns (%)

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
2006	0.99	-9.30	-8.42	12.23	-15.09	6.99	5.72	-0.16	13.46	5.33	13.05	-7.44
2007	1.05	-1.31	9.81	7.42	21.43	0.70	7.96	23.41	10.82	23.77	4.64	10.91
2008	0.00	0.00	0.00	0.00	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2009	0.00	0.00	0.00	0.00	0.52	4.87	14.72	2.95	10.19	1.70	6.05	5.41
2010	-8.86	6.54	14.85	11.10	-1.37	7.05	0.33	0.78	7.83	6.07	3.38	3.67
2011	5.20	4.09	-1.33	0.46	-0.66	-3.50	16.33	1.58	0.00	1.03	1.35	2.62
2012	12.71	18.83	10.53	-2.60	-1.07	1.09	4.58	9.39	0.28	-10.76	1.55	-9.07
2013	-14.41	-2.53	0.29	0.03	2.24	-11.83	14.12	8.38	-2.15	9.64	7.01	0.89
2014	-10.77	5.75	2.00	9.94	7.87	2.77	2.87	7.75	-1.71	10.49	10.60	-7.19
2015	6.14	10.08	-3.14	0.58	4.53	-3.72	-3.29	-4.78	0.00	5.01	2.69	-1.11
2016	0.00	0.00	7.70	-13.99	7.18	-2.91	9.01	2.37	6.55	0.43	-2.16	4.80
2017	4.77	13.38	4.87	-0.01	6.78	-5.72	3.27	10.70	-6.02	9.68	2.03	-1.52
2018	-1.06	6.82	-5.81	-1.50	13.51	-0.94	2.80	20.04	3.71	-1.47	4.97	3.46
2019	0.00	6.84	9.70	5.64	-10.83	14.17	7.64	-1.65	7.30	11.07	8.51	9.88
2020	5.40	-5.13	14.24	0.00	-0.33	14.74	16.51	21.66	-9.17	-6.00	9.55	11.46
2021	-0.55	-7.97	0.73	7.62	-5.05	9.91	6.50	4.25	-6.80	5.87	10.51	7.42
2022	1.08	0.66	8.68	-2.62	0.00	0.00	0.00	-0.15	0.00	0.00	4.83	0.44
2023	10.82	2.32	11.53	2.90	4.61	9.43	1.28	-4.18	-8.87	1.27	11.34	1.36
2024	-4.22	-1.85	-5.13	-0.67	13.02	9.56	5.44	3.24	1.75	-3.04	5.17	5.52
2025	-5.76	2.59	1.25	0.00	1.27	2.15	1.17	11.96	9.60	0.00	0.00	0.00

Strategy - Return Quantiles



Key Performance Metrics

Metric	Buy&Hold	Strategy
Risk-Free Rate	0.0%	0.0%
Time in Market	100.0%	80.0%
Cumulative Return	11,238.22%	99,294.51%
CAGR%	27.16%	41.99%
Sharpe	0.91	1.61
Prob. Sharpe Ratio	100.0%	100.0%
Smart Sharpe	0.89	1.57
Sortino	1.35	2.53
Smart Sortino	1.31	2.48
Sortino/√2	0.95	1.79
Smart Sortino/√2	0.93	1.75
Omega	1.38	1.38
Max Drawdown	-60.87%	-42.21%
Max DD Date	2009-01-20	2013-04-19
Max DD Period Start	2007-12-31	2012-09-20
Max DD Period End	2009-10-20	2014-06-06
Longest DD Days	660	625
Volatility (ann.)	31.92%	23.55%
R^2	0.55	0.55
Information Ratio	0.03	0.03
Calmar	0.45	0.99
Skew	0.03	0.31
Kurtosis	6.11	5.95
Expected Daily	0.1%	0.14%
Expected Monthly	2.02%	2.95%
Expected Yearly	26.69%	41.21%
Kelly Criterion	11.35%	14.41%
Risk of Ruin	0.0%	0.0%
Daily Value-at-Risk	-3.19%	-2.29%
Expected Shortfall (cVaR)	-4.41%	-4.41%
Max Consecutive Wins	11	10
Max Consecutive Losses	8	8
Gain/Pain Ratio	0.18	0.38
Gain/Pain (1M)	1.04	2.72
Payoff Ratio	1.14	1.2
Profit Factor	1.18	1.38
Common Sense Ratio	1.26	1.61
CPC Index	0.71	0.88
Tail Ratio	1.07	1.17
Outlier Win Ratio	3.6	5.93
Outlier Loss Ratio	3.42	4.37
MTD	9.6%	9.6%
3M	26.73%	26.73%
6M	16.85%	31.58%
YTD	1.96%	25.71%
1Y	12.01%	38.11%
3Y (ann.)	21.4%	39.1%
5Y (ann.)	18.11%	34.89%
10Y (ann.)	25.93%	44.33%
All-time (ann.)	27.16%	41.99%

Best Day	15.33%	10.5%
Worst Day	-17.92%	-12.36%
Best Month	23.77%	23.77%
Worst Month	-32.96%	-15.09%
Best Year	146.9%	205.09%
Worst Year	-56.91%	0.73%
Avg. Drawdown	-4.96%	-3.18%
Avg. Drawdown Days	35	22
Recovery Factor	9.42	17.65
Ulcer Index	0.17	0.1
Serenity Index	1.9	3.11
Avg. Up Month	7.54%	7.35%
Avg. Down Month	-5.42%	-4.83%
Win Days	52.86%	53.3%
Win Month	60.34%	71.9%
Win Quarter	68.35%	78.67%
Win Year	80.0%	100.0%

EOY Returns vs Benchmark

Year	Buy&Hold	Strategy	Multiplier	Won
2005	127.18%	nan%	nan	-
2006	18.01%	12.87%	0.71	-
2007	133.47%	205.09%	1.54	+
2008	-56.91%	0.73%	-0.01	+
2009	146.90%	55.97%	0.38	-
2010	53.07%	62.14%	1.17	+
2011	25.56%	29.19%	1.14	+
2012	32.57%	36.31%	1.11	+
2013	8.07%	8.07%	1.00	+
2014	40.62%	44.94%	1.11	+
2015	-3.01%	12.50%	-4.15	+
2016	12.48%	18.01%	1.44	+
2017	48.46%	48.46%	1.00	+
2018	-5.39%	50.98%	-9.46	+
2019	88.96%	89.61%	1.01	+
2020	82.31%	93.06%	1.13	+
2021	34.65%	34.65%	1.00	+
2022	-26.40%	13.20%	-0.50	+
2023	49.01%	50.57%	1.03	+
2024	30.71%	30.71%	1.00	+
2025	1.96%	25.71%	13.11	+

Worst 10 Drawdowns

Started	Recovered	Drawdown	Days
2012-09-20	2014-06-06	-42.21%	625
2006-01-17	2006-11-17	-31.58%	305
2016-04-15	2016-08-12	-18.92%	120
2021-01-27	2021-07-06	-18.60%	161
2020-09-02	2020-12-24	-17.96%	114
2012-04-10	2012-08-15	-16.68%	128
2023-12-15	2024-06-10	-16.61%	179
2019-05-06	2019-07-12	-15.91%	68
2007-01-17	2007-04-25	-14.24%	99
2024-12-27	2025-05-09	-14.05%	134

Beta	-	0.55
Alpha	-	0.22
Correlation	-	74.01%
Treynor Ratio	-	181850.23%

References

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences, 191*, 192–213. https://doi.org/10.1016/j.ins.2011.12.028

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

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2

Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233–240). ACM. https://doi.org/10.1145/1143844.1143874

De Prado, M. L. (2018). *Advances in Financial Machine Learning*. Wiley. https://doi.org/10.1002/9781119482086

Faber, M. T. (2007). A quantitative approach to tactical asset allocation. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.962461

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. *Nature, 585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance, 48*(1), 65–91. https://doi.org/10.1111/j.1540-6261.1993.tb04702.x

McKinney, W. (2022). *Python for Data Analysis* (3rd ed.). O'Reilly.

Moskowitz, T. J., Ooi, Y. H., & Pedersen, L. H. (2012). Time series momentum. *Journal of Financial Economics, 104*(2), 228–250. https://doi.org/10.1016/j.jfineco.2011.11.003

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

Sharpe, W. F. (1994). The Sharpe ratio. *The Journal of Portfolio Management, 21*(1), 49–58. https://doi.org/10.3905/jpm.1994.409501

Wilder, J. W. (1978). *New Concepts in Technical Trading Systems*. Trend Research.

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 694–699). ACM. https://doi.org/10.1145/775047.775151

Software & documentation

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

McKinney, W., et al. (Pandas Development Team). (2024). *pandas: Python data analysis library* (Version 2.x) [Computer software]. https://pandas.pydata.org/

NumPy Developers. (2024). *NumPy* (Version 1.x) [Computer software]. https://numpy.org/

Pedregosa, F., et al. (scikit-learn developers). (2024). *scikit-learn* (Version 1.x) [Computer software]. https://scikit-learn.org/stable/

QuantStats Developers. (2024). *quantstats* (Version 0.x) [Computer software]. https://github.com/ranaroussi/quantstats

Roman, R. (yfinance). (2024). *yfinance* (Version 0.x) [Computer software]. https://github.com/ranaroussi/yfinance

Seaborn Developers. (2024). *seaborn* (Version 0.x) [Computer software]. https://seaborn.pydata.org/

Statsmodels Developers. (2024). *statsmodels* (Version 0.x) [Computer software]. https://www.statsmodels.org/

Warren, K. (pandas-ta). (2024). *pandas-ta: Technical Analysis Indicators* (Version 0.x) [Computer software]. https://github.com/twopirllc/pandas-ta

ydata (2024). *ydata-profiling* (Version 4.x) [Computer software]. https://github.com/ydataai/ydata-profiling

ipywidgets Developers. (2024). *ipywidgets* (Version 8.x) [Computer software]. https://ipywidgets.readthedocs.io/

PyFolio Contributors. (2020). *pyfolio* (archived) [Computer software]. https://github.com/quantopian/pyfolio

Additional domain references

Asness, C. S., Moskowitz, T. J., & Pedersen, L. H. (2013). Value and momentum everywhere. *The Journal of Finance, 68*(3), 929–985. https://doi.org/10.1111/jofi.12021

Baltas, A., & Kosowski, R. (2020). Demystifying time-series momentum strategies: Volatility estimators, trading rules and pairwise correlations. *Critical Finance Review, 9*(2), 211–278. https://doi.org/10.1561/104.00000088

Harvey, C. R., Liu, Y., & Zhu, H. (2016). …and the cross-section of expected returns. *The Review of Financial Studies, 29*(1), 5–68. https://doi.org/10.1093/rfs/hhv059

Lo, A. W. (2002). The statistics of Sharpe ratios. *Financial Analysts Journal, 58*(4), 36–52. https://doi.org/10.2469/faj.v58.n4.2453

Novy-Marx, R. (2012). Is momentum really momentum? *Journal of Financial Economics, 103*(3), 429–453. https://doi.org/10.1016/j.jfineco.2011.05.003