

Multi-Model Algorithmic Trading System with Regime Detection, Strategic Stock Selection, and Risk-Aware Allocation

Problem Description

The central question this paper hopes to address is “Can regime detection models identify market conditions in real-time to enable tactical allocation shifts to better strategies?” I will seek to answer this by extending my research from the Checkpoint B assignment looking at creating automated trading for the Anabasis Transformative Technologies Fund (ATTF). I will do this by implementing four complementary algorithmic trading strategies, a Clenow momentum strategy, a Machine Learning (ML) strategy, a mean-reversion strategy, and a hybrid strategy. The research implements a multi-modal regime-switching framework that dynamically allocates capital based on detected market conditions using a Hidden Markov Model (HMM) trained on SPY returns. This approach reflects the assignment’s core objective to build automated trading systems that adapt to changing market environments without requiring manual intervention.

The four implemented strategies represent distinct hypotheses about sources of active management alpha. First, the Clenow momentum strategy implements a systematic trend-following approach combining 90-day intermediate-term momentum with 200-day moving average filters. This dual-filter design concentrates capital in securities exhibiting both strong recent performance and confirmation of long-term uptrend continuation. The ML walk-forward strategy extends Checkpoint B’s Ridge regression framework by implementing monthly

rebalancing with rigorous out-of-sample validation across 70 testing periods. Next, a mean-reversion strategy tests whether transformative technology stocks exhibit oversold bounce-back behavior using z-score thresholds on rolling price deviations. Finally, a hybrid regime-switching portfolio system combines these strategies based on HMM classification of bull, sideways, and bear market regimes.

Data Preparation and Pipeline

I warm-started the Checkpoint B codebase for this assignment, but I rebuilt the data and feature pipeline to remove any sources of leakage and to support robust out-of-sample evaluation. The ability to start this programming assignment from the Checkpoint B codebase enabled me to tweak the models and strategies, creating significantly different results. The investible universe comprises 21 dividend-adjusted ATTF securities grouped into four thematic sleeves with SPY serving as the primary benchmark from 2015 through 2025. Data sources included polygon.io, yahoo finance, and FRED for macro data. Dividend adjustments were calculated as $(1 + \text{dividend}/\text{price})$, and all future prices were multiplied by this factor cumulatively to properly account for dividend cash flows.

For Programming Assignment 3, I incorporated technical feature engineering to generate the signals necessary for algorithmic trading execution. These enabled two complementary layers of predictors; regime features that inform when to take risk, and stock-level features that drive what to own. For regime detection, I paired SPY's daily return and 20-day annualized volatility with macro signals from FRED, including 10- and 2-year yield-curve slope, effective fed funds, unemployment rates, VIX, CPI year-over-year, and industrial production year-over-year, forward-filled and time-normalized before joining them to the market series. For each security,

technical indicators were calculated daily. These were 20-, 60-, and 120-day price momentum, 20 and 60-day return volatility, 20 and 60 day moving average ratio. I also added SPY context features including 20- and 60-day momentum and 20-day volatility.

The regime detection pipeline implements a three-state HMM trained on SPY daily log returns over the full 10-year sample. The HMM estimates transition probabilities between bull, sideways, and bear market regimes based on return and volatility characteristics. The Viterbi algorithm decodes the most likely regime sequence across the entire sample, providing historical regime classification for every trading day. Bull regimes are characterized by positive mean returns and moderate volatility, sideways regimes exhibit near-zero mean returns with elevated volatility, and bear regimes show negative returns with high volatility.

This research utilizes realistic cost capture to enable algorithmic trading implementation. These include a 1.5% management fee, transaction costs of 15 basis points, and an annual 5 basis points per trade for estimated market impact costs. For strategies with monthly rebalancing, costs are calculated from actual position changes rather than assumed turnover rates, providing precise net-of-fees performance measurement.

Research Design

The methodology employs four distinct algorithmic trading strategies, each addressing a specific hypothesis about systematic return generation. The strategies are designed to be mutually exclusive in their signal generation logic, enabling direct performance comparison and combination in a regime-switching framework. Each strategy is implemented with explicit rebalancing rules, net-of-fees performance, and common evaluation metrics to include CAGR, Sharpe ratio, max drawdown, volatility, and alpha/beta vs. SPY.

The Clenow momentum strategy implements a systematic trend-following approach that employs dual-filter design combining intermediate-term momentum with a long-term trend filter. At each month-end, ATTF securities whose price is above the 200-day moving average are eligible. They are then ranked by their 90-day momentum score, and the strategy then allocates equal weight to the top 10 securities, concentrating capital in those exhibiting the strongest recent performance. Monthly rebalancing weighs the persistence of momentum signals against transaction cost drag. The dual-filter design reflects empirical evidence that momentum strategies perform best when combined with trend-following filters, as pure momentum can suffer during trend reversals when high-momentum securities experience sharp corrections.

Programming Assignment 3's ML strategy utilizes monthly testing periods where Ridge regression models are trained on rolling 3-year windows of historical data to predict forward 21-day returns. Ridge regression provides L2 regularization that prevents overfitting while maintaining computational efficiency suitable for monthly retraining. At month-end, the strategy selects the top 10 securities by predicted return and allocates equal weight to the selected portfolio, which could be less than 10. Positions are held for 21 trading days while calculating realized returns and transaction costs, then the model rolls forward one month and retrains on the updated data window. This training approach ensures that models incorporate all historical information without introducing look-ahead bias, providing realistic estimates of strategy performance in live trading. The monthly rebalancing frequency balances the need for model adaptation to changing market conditions against transaction cost minimization.

My third strategy incorporated into this assignment was a mean-reversion strategy. This strategy calculates z-scores on rolling 60-day price deviations from the mean, identifying

securities trading more than 0.3 standard deviations below their recent average. At each daily rebalancing date, the strategy allocates equal weight to the 10 most oversold securities, betting on reversion to fair value.

As an upgrade to Checkpoint B, I incorporated a hybrid regime-switching strategy that combined the Clenow momentum and ML strategies on HMM classification of market regimes. The system implements distinct allocation rules for each regime. In bull markets characterized by strong uptrends and low volatility, the system allocates 40% to Clenow momentum and 60% to ML for signal generation, then choosing the top names but no more than 15 from this blend and equal weights them, capturing both trend-following and ML's pattern recognition. In sideways markets with elevated uncertainty, the system allocated 100% to ML, relying on its adaptive retraining to navigate choppy conditions. In bear markets, the system moves entirely to cash equivalents to preserve capital, meeting the assignment's objective of establishing a system that liquidates and suspends trading. At each month-end rebalancing date, the system queries the HMM regime classification for that day and implements the corresponding allocation. Transaction costs are calculated from the notional value of all position changes, including both strategy rotation within regimes and regime transactions. The system's performance depends critically on regime transaction frequency. Over the 10-year backtest, the HMM identified 64 bull, 35 sideways, and 20 bear months, with 118 regime switches. This relatively high transition frequency suggests that transaction costs from regime switching may erode the system's tactical allocation benefits. Two benchmarks provide performance comparison. The SPY buy-and-hold benchmark represents passive large-cap US equity exposure, and QQQ was introduced into this assignment as a technology focused, non-financial benchmark. All active strategies must clear these hurdles net of fees to demonstrate value creation from systematic trading rules.

Programming Implementation

I used Jupyter notebook for this assignment, leveraging Python, with modular sections for data loading, feature engineering, strategy execution, and visualization. Key libraries include NumPy for numerical computation, Pandas for time series manipulation, Scikit-learn for machine learning, and Matplotlib/Seaborn for visualization. The data acquisition module interfaces with the Polygon.io API to fetch daily OHLCV dividend-adjusted data for each ATTF security and the SPY benchmark. Price data is converted into logarithmic returns and missing data is handled through forward-filling for securities with temporal halts and exclusion for securities with insufficient history. The resulting DataFrame contains aligned daily log returns for all securities and SPY, indexed by trading date with proper time-zone handling.

The feature engineering pipeline calculates technical indicators using Pandas rolling window operations for momentum and volatility features. These features are calculated with proper `shift(1)` operations to prevent look-ahead bias. For machine learning training, features are standardized using `sklearn.preprocessing` on the training window only. This ensures that test set features are scaled using only information available at prediction time. The feature matrix `X` has shape (trading days, 129) and the largest vector `y` contains forward 21-day returns for each security. For ML, the `sklearn Ridge` class provides efficient L2-regularized regression with `.fit()` and `.predict()` methods. Model predictions are stored in a predictions DataFrame for subsequent analysis of which securities the ML strategy selected over time. The Clenow momentum strategy is implemented as a function that accepts price data and parameters. At each month-end, the function conducts the calculations to move around securities within the strategy. The function generates a DataFrame with shape(trading days, 21) where each cell contains the allocation

weight for that security on that day. For mean-reversion, z-scores are calculated using $(\text{prices_rolling_mean})/\text{rolling_std}$ on 20-day windows. Securities with z-scores below -0.3 are flagged as oversold. The regime-switching hybrid system queries an HMM regime classification DataFrame at each rebalancing date and implements regime-conditional allocation rules. My hybrid system calculates portfolio weights as a weighted combination of strategy signals, then applies these weights to daily returns to generate portfolio returns.

My notebook includes modular function design, comprehensive error handling, and extensive validation checks. The code includes sanity checks at critical junctures verifying that feature matrices have expected dimensions, confirming that walk-forward windows don't overlap, checking for data leakage by examining feature calculation timing, and validation that transaction costs are applied correctly. Extensive debugs and print statements provide examples of troubleshooting conducted, examples of real-time feedback during execution, tracking progress through data downloads, strategy signal generation, and performance calculation. The notebook format enables iterative execution, facilitating debugging and experimentation. Output includes progress bars for long-running operations, intermediate results tables for inspection, and diagnostic plots showing cumulative returns, drawdowns, and rolling Sharpe ratios.

Exposition

The results from this assignment validate that an HMM regime aware machine learning strategy is the best strategy for the ATTF portfolio. Table 1 presents the comprehensive performance comparison across all strategies and benchmarks net of fees. ML dominates with 116.64% CAGR and 3.685 Sharpe, the hybrid regime system delivered 41% CAGR and 1.54

Sharpe, the Clenow momentum produced 23.5% CAGR with 0.594 Sharpe, and the mean-reversion produced 15.8% CAGR with 0.33 Sharpe.

Table 1: Final Comparison of All Strategies + Benchmarks with Fixed Betas and Full Costs

=====						
FINAL COMPARISON - CORRECTED BETAS & FULL COSTS						
=====						
	Clenow	ML	Mean-Rev	Hybrid	SPY	QQQ
CAGR	0.234640	1.166368	0.157594	0.409009	0.121388	0.133677
Volatility	0.327736	0.305689	0.357811	0.239871	0.182066	0.196713
Sharpe	0.593892	3.684689	0.328648	1.538364	0.447027	0.476215
Max Drawdown	-0.695848	-0.209681	-0.840275	-0.315506	-0.341047	-0.301906
Alpha	0.100910	1.064231	0.003794	0.323121	-0.000033	0.021711
Beta	1.151628	0.763471	1.398237	0.563826	1.000401	0.884233

This ML strategy's exceptional performance reflects three structural advantages. First, the portfolio concentration averages only 6 stocks at any rebalancing date, enabling significant outperformance when the selected securities experience explosive growth. Second, the strategy's annual turnover of just 6.8% minimizes transaction cost drag while allowing sufficient adaptation to changing market conditions. Third, the walk-forward validation ensures that the strategy captures genuine patterns rather than overfitting to historical noise. Clenow momentum underperformed expectations, and the -69.58% drawdown during the 2022-2023 correction highlights momentum's vulnerability to sharp trend reversals. The strategy's 1.152 beta and minimal alpha indicate that returns were primarily driven by market exposure rather than genuine strategy selection skill. My hypothesis going into the assignment was that mean-reversion strategies do not work for transformative technology portfolios. However, mean-reversion was able to produce 15.8% CAGR for this assignment. It suffered catastrophic drawdowns when forced to take positions and the CAGR and Shape ratio represents pure market exposure without value-add from the reversion signals. This empirical result confirms that

securities in the ATTF portfolio tend to sustain directional moves for extended periods, making contrarian strategies unsuitable. The hybrid regime-switching system delivered compelling risk-adjusted performance with 41% CAGR, 23.99% volatility, and 1.54 Sharpe ratio. The system's lower beta of 0.564 compared to ML's 0.763 reflects cash allocation during the bear months. Maximum drawdown was also more favorable than Clenow or mean-reversion, suggesting that regime detection provided genuine downside protection. However, the 118 switches indicate frequent transitions that may degrade performance when regime classification accuracy falls below historical levels. Of note, all four strategies outperformed the SPY and QQQ benchmark.

Programming Assignment 3 demonstrates that automated algorithmic trading strategies can generate exceptional returns when applied to transformative technology sectors. Four key findings emerge from this analysis. Machine learning with rigorous out-of-sample validation outperforms traditional technical strategies for innovative technological securities. Portfolio concentration in high-conviction names generates superior returns compared to diversification. ML's average position size of six securities enabled participation in explosive AI Infrastructure growth during the past several years. Next, transformative technology stocks exhibit persistent momentum, making contrarian strategies unsuitable for this asset class. Finally, regime detection frameworks provide tactical allocation benefits but at the cost of foregone upside during bear market rallies.

Based on these findings, the ATTF should adopt machine learning as the primary systematic strategy with 70% portfolio allocation, with the remaining 30% going to opportunistic tactical positions or Monte Carlo optimization rebalancing quarterly. Clenow should continue to be investigated as an appropriate strategy but mean-reversion should not be considered moving

forward for the ATTF. Regime-switching frameworks remain promising but require additional development to investigate look-ahead bias.

The exceptional ML performance raises important questions about strategy sustainability and implementation risk. The 116% CAGR reflects a historical bull market in AI, but future returns will likely moderate as markets mature and alpha decay intensifies. The strategy's concentration also introduces risk from individual security failures. Sector diversification and individual security caps should be implemented to mitigate this risk. Monthly rebalancing creates implementation challenges around timing, liquidity, and market impact costs. ATTF should implement pre-trade analytics to assess market conditions and potentially defer rebalancing when spreads widen or volatility spikes.

Several critical enhancements will strengthen future research and implementation. Feature engineering should incorporate alternative data and sentiment analysis and fundamental signals beyond technical indicators should be included. Transaction cost sensitivity analysis should vary assumptions across market cap segments to bracket realistic implementation costs for small-cap high-spread names versus large-cap liquid securities. Ensemble methods combining multiple ML models such as Ridge, Lasso, RandomForest, and Gradient Boosting, may improve robustness and reduce overfitting risk compared to single-model strategies. I also need to validate these strategies using proxies against more significant market downside, such as testing QQQ and SPY during the dot-com bubble or Great Financial Crisis. Overall I learned that it is challenging creating an algorithm that is able to capture alpha in a meaningful way, and even though I made progress beyond Checkpoint B with this assignment, and more things are working than before, there is much more integration, testing, and validation to go in order to create a system that I can actually utilize for live trading.

