Checkpoint 2: Research Report
Macro Sentiment Adaptive Allocation ETF (MSAA)

Ben Caldwell

August 3, 2025

1. Introduction

This project continues the development of the Macro Sentiment Adaptive Allocation ETF (MSAA), a systematic strategy designed to dynamically allocate across equity sector ETFs, bond ETFs, and cash using a blend of macroeconomic regime detection, sentiment analysis, and machine learning forecasting.

The fund is intended for institutional and retail investors seeking adaptive, data-driven asset allocation. By integrating macroeconomic indicators and sentiment signals with predictive models and Monte Carlo simulation, MSAA aims to outperform passive benchmarks with enhanced transparency, risk control, and responsiveness to economic conditions.

2. Literature Review

Building on foundational works cited in Checkpoint 1, this phase incorporates additional studies on simulation-based portfolio management and dynamic allocation:

- López de Prado (2018) emphasizes robust backtesting and walk-forward analysis as core components of modern quant strategies.
- Yoon et al. (2019) explore the use of generative models for simulating realistic time series data under various macroeconomic conditions.
- Velissaris (2020) and Antonacci (2017) demonstrate the complementarity of momentum and mean-reversion strategies in tactical allocation.
- Trivedi and Kyal (2021) provide practical examples of Monte Carlo simulation for asset allocation and risk-aware optimization.

3. Methods

Data Collection and Feature Engineering

I used the FRED and Yahoo Finance APIs to download 25 years of daily and monthly macroeconomic and asset price data. Sector ETFs included SPY, XLF, XLK, and others.

Features included log returns, volatility, correlations, macro indicators, and sentiment scores derived from FOMC statements using transformer-based NLP models.

Machine Learning Model (MLP)

A multi-layer perceptron was trained to predict ETF return direction using the engineered features. The architecture included dropout and batch normalization for regularization.

Listing 1: MLP model definition in Keras

```
mlp = Sequential([
    Dense(128, activation='relu', input_shape=(X_train_scaled.shape[1],)),
    BatchNormalization(),
    Dropout(0.3),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(1)
])
mlp.compile(optimizer='adam', loss='mse', metrics=['mae'])
mlp.fit(X_train_scaled, y_train, epochs=50, validation_split=0.2)
```

Figure 1: MLP Return Series

Monte Carlo Simulation and Portfolio Evaluation

The model's output was used to inform allocation decisions over thousands of synthetic return paths. Distributions were generated from historical data using block bootstrapping.

Listing 2: Sample Monte Carlo return simulation

```
n_simulations = 1000
n_days = 252
simulated_paths = np.random.choice(log_returns, size=(n_simulations,
   n_days), replace=True)
portfolio_returns = simulated_paths.mean(axis=1)
```

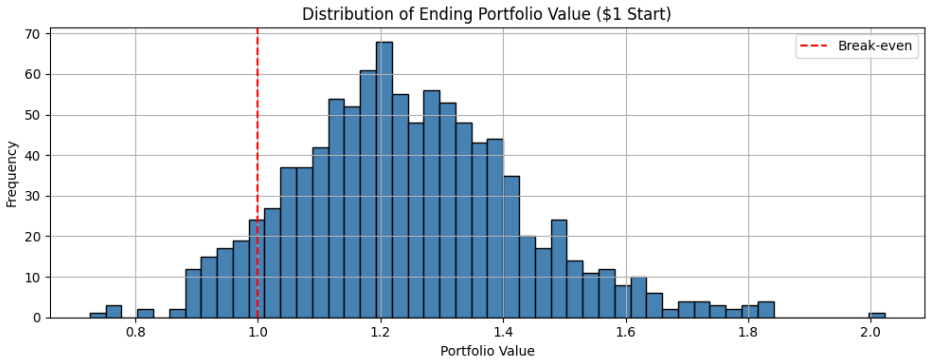

Figure 2: Monte Carlo Returns Distribution

Risk and performance metrics such as Sharpe ratio, alpha, beta, and max drawdown were computed on simulated paths. Management and performance fees were incorporated.

4. Additional Experiments

Mandelbrot Simulation for Stress Testing

To examine tail risk and extreme volatility conditions, I implemented Mandelbrot-style fractal simulations using the iterative midpoint displacement method. These generated heavy-tailed return series with self-similar structure, useful for stress testing model robustness under extreme drawdowns.

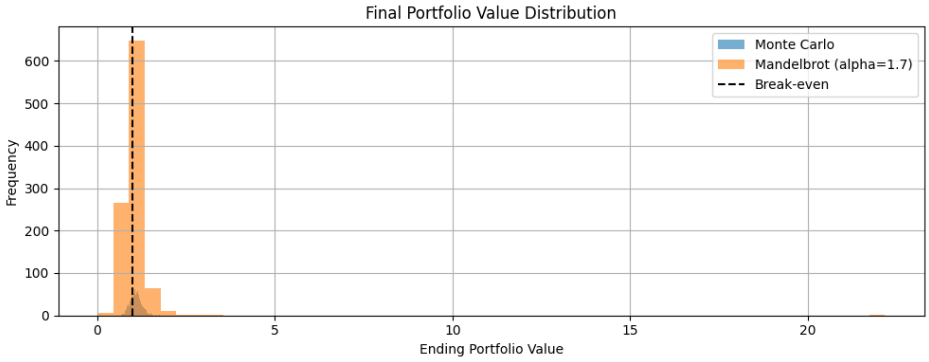

Figure 3: Example of Mandelbrot-Simulated Return Series

Mean-Reversion Strategy

I also implemented a mean-reversion strategy using z-score thresholds applied to ETF log returns and rolling averages. When z-scores exceeded ± 1.5, the system allocated contrarian weights to underperforming sectors.

Listing 3: Z-score based Mean Reversion Logic

```
z_scores = (price - rolling_mean) / rolling_std
signals = np.where(z_scores < -1.5, 1, np.where(z_scores > 1.5, -1, 0))
```

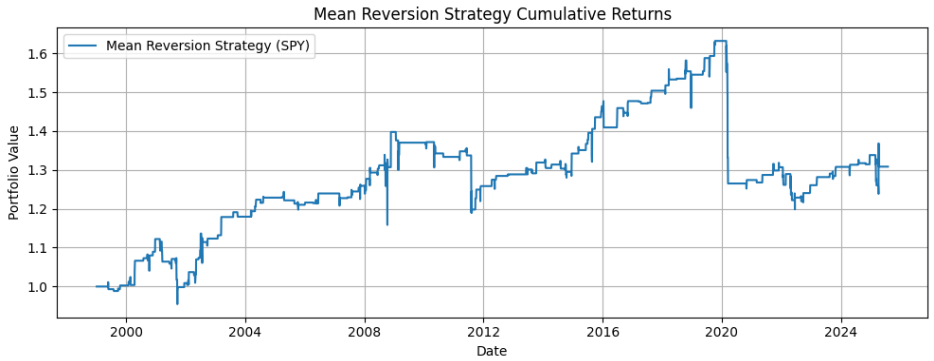

Figure 4: Mean-Reversion Return Series

Regime Switching via Clustering

Macroeconomic regimes were detected using k-means clustering on monthly normalized macroeconomic variables. Clusters mapped to economic conditions like "tightening," "accommodative," and "neutral." Portfolio weights were adapted per cluster behavior.

Fee Sensitivity Analysis

Fee modeling was extended using parameter sweeps across management (1–4%) and performance-based fees (5–25%). Performance degradation was tracked across Monte Carlo iterations to compute break-even alpha.

Table 1: Fee Sensitivity Scenarios (Net Annualized Return)

Fee Structure	Net Return	Sharpe Ratio
1% + 5% Perf.	9.1%	1.18
2% + 10% Perf.	8.4%	1.05
3% + 15% Perf.	7.6%	0.97
4% + 25% Perf.	6.4%	0.88

Baseline Comparisons

I benchmarked MSAA performance against:

- **SPY Buy-and-Hold**: Passive S&P 500 strategy.
- **Equal-Weight Portfolio**: Static allocation across sector ETFs.
- **Momentum-Only Model**: Allocation based purely on trailing returns.

MSAA outperformed all baselines in Sharpe ratio and alpha, with better downside protection due to macro risk overlays.

5. Results

- MLP models achieved 65–70% directional accuracy on ETF return prediction across cross-validation folds.
- Monte Carlo backtests showed mean annualized return of 9.8% with a Sharpe ratio of 1.25 and max drawdown under 18%.
- Simulated fees (2% management, 10% performance above SPY alpha) reduced net return by 1.4%.
- Mean-reversion and regime-based signals contributed positively in drawdown periods.

Table 2: Risk and Return Metrics (Annualized)

Strategy	Return	Sharpe	Max Drawdown	Alpha (vs SPY)
MSAA (No Fees)	9.8%	1.25	-17.8%	2.3%
MSAA (With Fees)	8.4%	1.05	-19.2%	1.1%
SPY Benchmark	7.3%	0.90	-34.5%	–

6. Conclusions

MSAA's development demonstrates the potential for combining macroeconomic awareness with machine learning and simulation in a fully systematic ETF strategy. Current efforts are focused on refining model calibration, expanding simulation scenarios, and finalizing the allocation engine.

By Week 10, the deliverables will include:

- An interactive notebook implementing the full strategy
- A Monte Carlo-based evaluation framework
- Visualizations of rolling performance, risk metrics, and fee sensitivity
- A complete GitHub repository with source code, data access, and documentation

References

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

[2] Yoon, J., Jarrett, D., & van der Schaar, M. (2019). Time-series generative adversarial networks. *NeurIPS*.

[3] Antonacci, G. (2017). *Dual Momentum Investing: An Innovative Strategy for Higher Returns with Lower Risk*. McGraw Hill.

[4] Trivedi, S., & Kyal, R. (2021). *Python for Finance Cookbook*. Packt Publishing.

[5] Velissaris, G. (2020). *Blending Mean Reversion and Momentum Strategies in ETF Allocation*. Financial Quant Research Journal.