Prediction Market ML-Ensemble Strategy with Volatility Gating and Kelly-Criterion Sizing

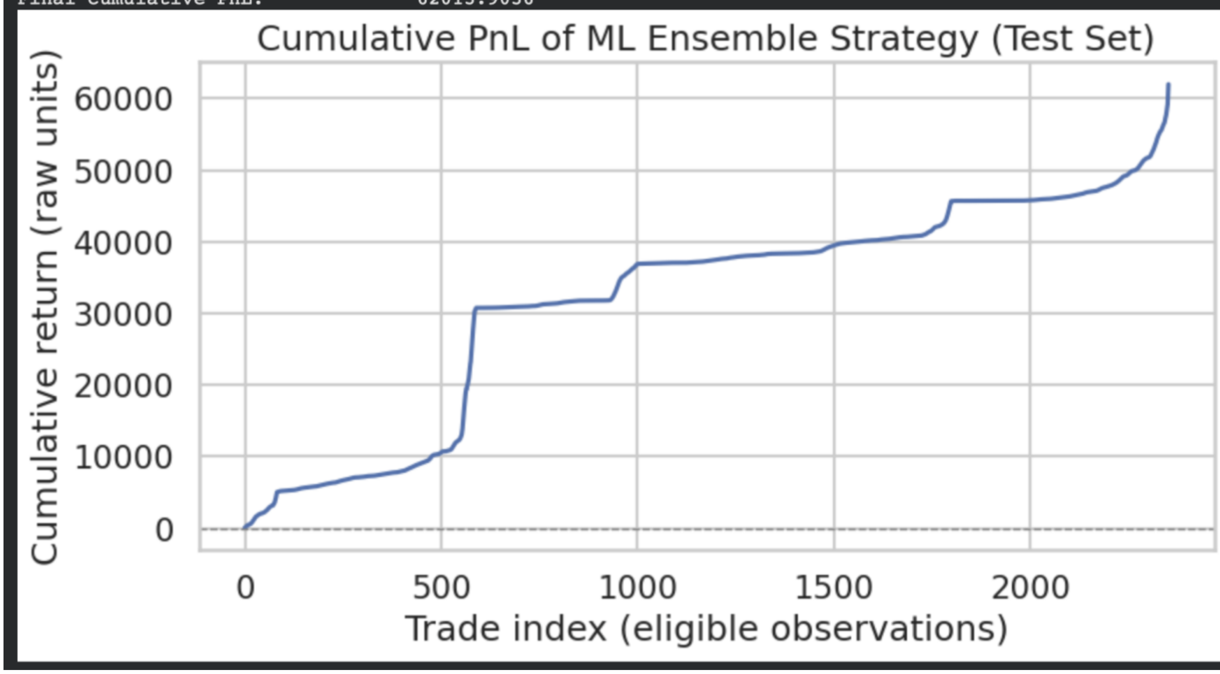
My strategy exploits short-term statistical arbitrage for prediction markets using ensemble learning with topological anomaly detection, machine learning, and volatility prediction with Kelly Criterion sizing.

Initially, this report started off as a way to study insider trading bets in prediction markets and positive feedback loops due to mispricing of events in different markets. Both proved to be infeasible. It is very difficult to retrieve analogous data and same (or even similar) contracts for the same exact events across platforms; therefore, it would not be suitable for fast algorithmic trading bots. Second, my initial hypothesis of insider bets in prediction markets turned out to be way off in larger (more liquid) markets where events were based on a collection of individuals vs a single person (for example, harder to predict insider trades for elections with large number of people voting versus simply how many times Elon Musk will mention robotaxi). Given those limitations, I pivoted to the following:

- Don't bet with smart people (insiders) but bet heavily against very dumb people. Large and anomalous bets (those detected by the mathematical techniques used in topological data analysis, or TDA for short) almost always reverted back to the mean of consensus of the market. This inefficiency leads to great opportunity when markets are more efficient. Therefore, my first signal comes from identifying these anomalies and betting against them, which works roughly 61% of the time (probability of being on the right side of the bet) in my dataset of over 10,000 trades. First figure in Appendix shows the returns of this bet. I improved this further by including stop loss and take profit levels (detailed in my Juypter notebook).
- A simple machine learning classifier with technical analysis of features of the trade in the form of a logistic regression. In essence, a technical support for my decision boundaries in the previous point.
- Play it safe with volatility by measuring and predicting how volatility clusters and can impact risk to reward ratio. I used the generalized arch model that incorporates the covariances of previous trades to band my decisions to buy and sell only when the volatility is within the 60% central range (i.e., don't mess with tail volatility bets).
- And finally, I introduced sizing my bets with the Kelly criterion so that I can reduce the risk of me losing all my money and the volatility of returns.

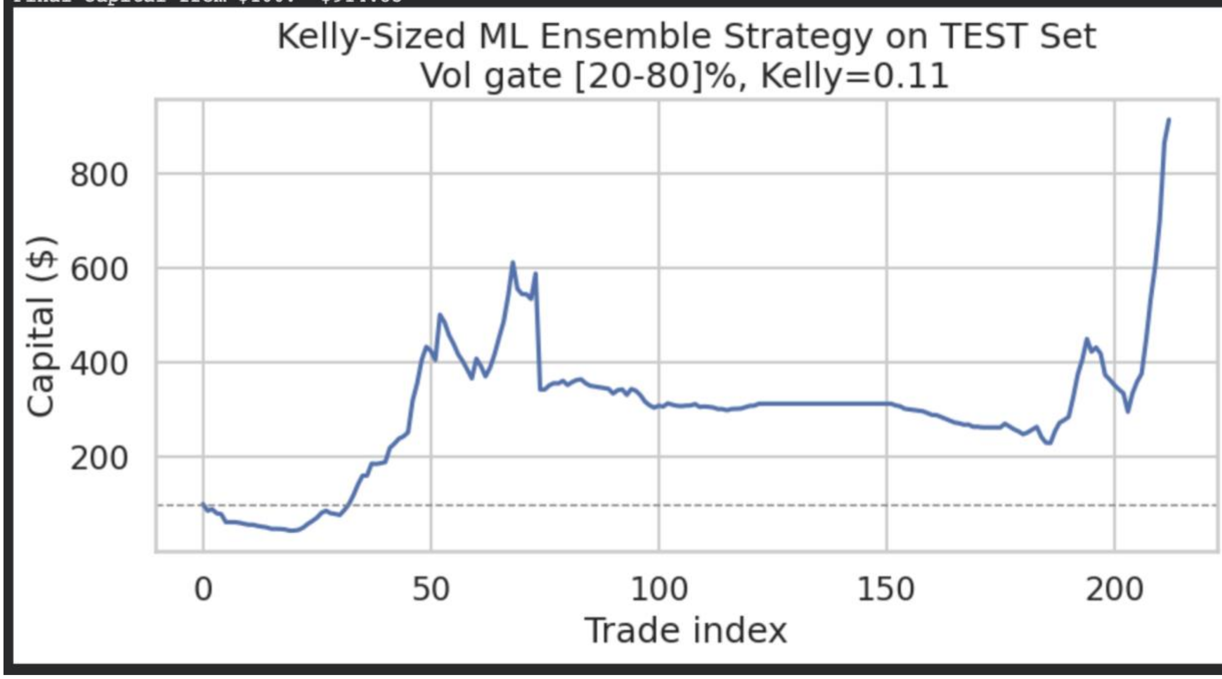
In the end, I bring this all together by simply asking each component to vote whether to buy (vote yes) or sell (vote no) a contract for the event. I use SoftMax voting, which essentially means that each component votes with a percentage of confidence in their decision, and I pick the highest cumulative percentage. Since, all of them are good classifiers by themselves, in the limit of convergence, they should almost always make a better choice together than alone. This is reflected in back test (I use a 30-70 test-train split where I reserve 30% of the trade data for only testing so that I don't overfit my classifiers), where my classifier does exceedingly well with a win percentage of 91.2%. The following graph shows the returns to $1 with my strategy:

```
=== ML Ensemble Trading Strategy (Test Set) ===
Number of signals (|position|>0): 2352
Win rate on those trades:         91.20%
Average return per trade:         26.3665
Std dev per trade:                83.5826
Per-trade Sharpe (raw):           0.32
Final cumulative PnL:             62013.9036
```



While this is good, I wanted to put guard rails on my strategy, so I don't go bust. Therefore, I included hard limits of only betting when volatility is slightly less than 1 standard deviation away from the mean:

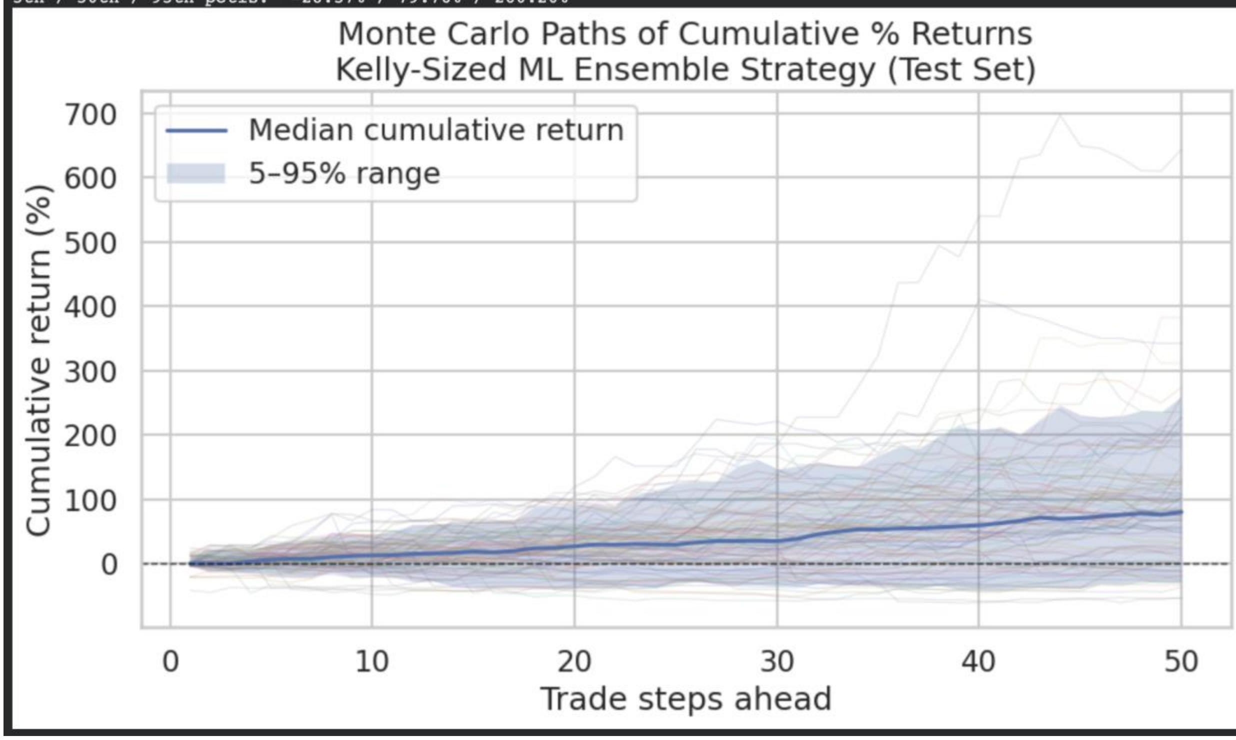
```
=== Kelly-sized ML Ensemble Strategy (TEST) ===
Number of trades:          212
Win rate (test):           38.68%
Mean scaled return:        0.1213
Std scaled return:         0.6682
Per-trade Sharpe (scaled):0.18
Final capital from $100:  $914.83
```



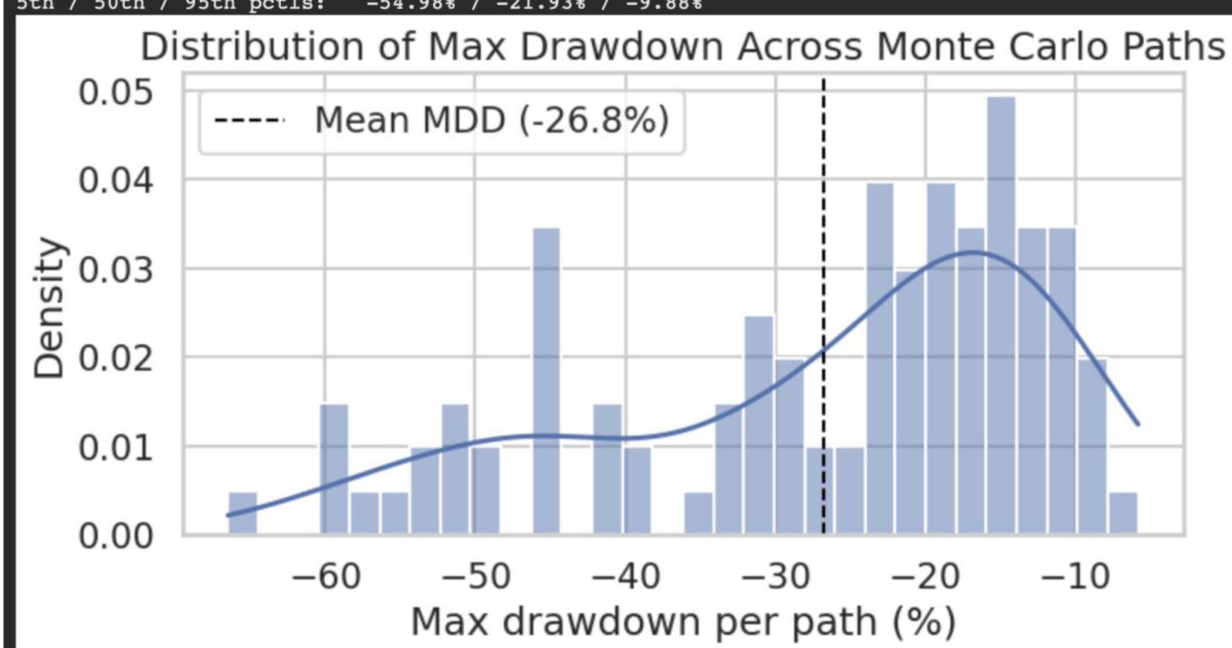
And, while this graph looks less pretty, it is significantly more robust: even with a lower win percentage I see significantly outsized positive returns. That initially seemed weird; however, on further inspection, I can see that it protects us from extreme volatile movements when prediction markets make stepwise changes in prediction probabilities (let's say win percentage skyrockets from 10% to 60%). My sample dataset, in fact, shows that simply betting on this strategy improves returns without limits, but that could just be an artifact of my sample number of markets (5 events) and limited trade information (I can only get a maximum of 10,000 trades per market from polymarket's API).

Doing a Monte Carlo Simulation on this more conservative approach yields further better results, as it emphasizes that my strategy works on data that is simply not just what I trained on. The simulation below is only of 50 trades, my returns actually improve significantly when I increase the simulated trades. And, to strength my case, I also looked at some risk metrics we discussed in class using a Monte Carlo simulation of 100 trades. The results of both are below.

```
=== Monte Carlo on Kelly-Sized TEST Paths (Bootstrap on test_r) ===
Simulated paths:          100
Horizon (trades):         50
Mean final cumulative %:  93.57%
Std final cumulative %:   104.63%
5th / 50th / 95th pctls:  -28.57% / 79.76% / 260.20%
```




```
=== Monte Carlo Max Drawdown Across Paths ===
Mean max drawdown:          -26.79%
5th / 50th / 95th pctls:    -54.98% / -21.93% / -9.88%
```



Clearly the returns are great; however, the volatility is also extremely high even with us restricting our bet sizes. But this does make sense given we are working on a prediction market. While it is hard to calculate hedge fund performance metrics because I could not convert this into a time interval, the returns are clearly staggering and all take place in between 2023 and 2024. I would expect this strategy to make approx. 200 trades per year given the number of markets on PolyMarket, and with those very high returns.

Unfortunately, I cannot simulate this in real time yet because PolyMarket does not publicly make that data available; but my test data and Monte Carlo simulations are meant to replicate the viability of my strategy through unseen data.

This was fun and pretty cool!

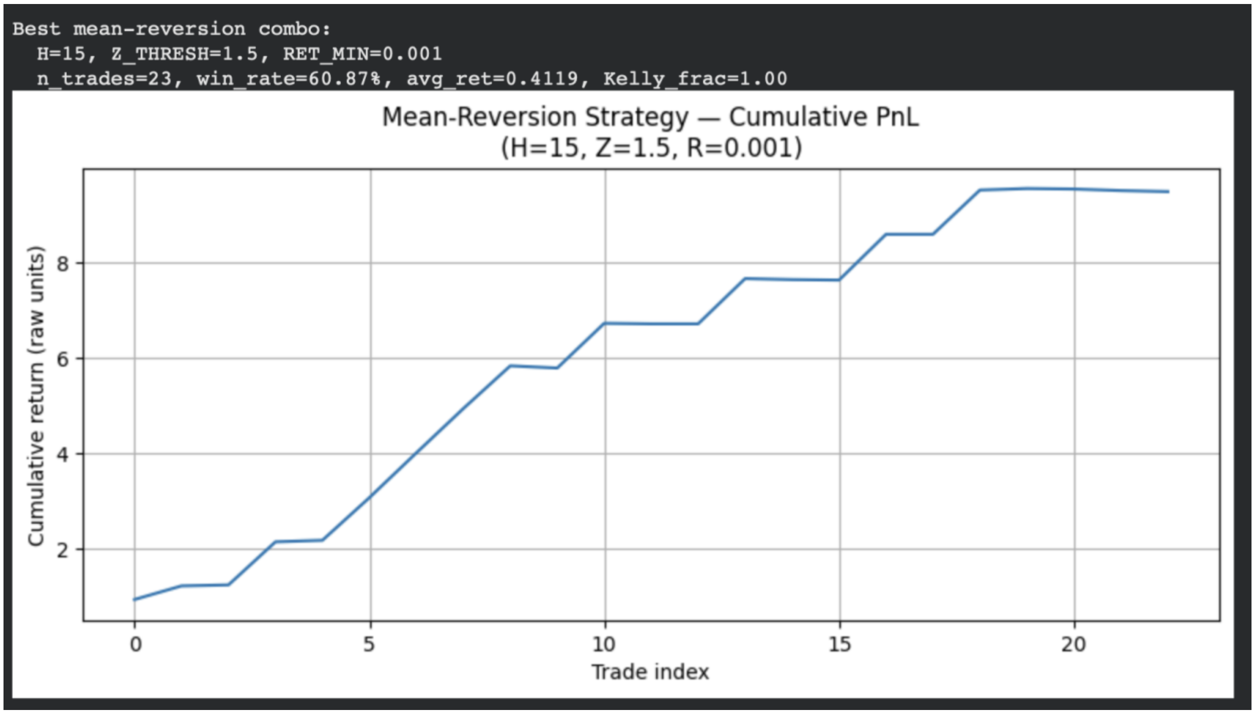


Figure 1, Mean Reversion Strategy Using TDA (Best)

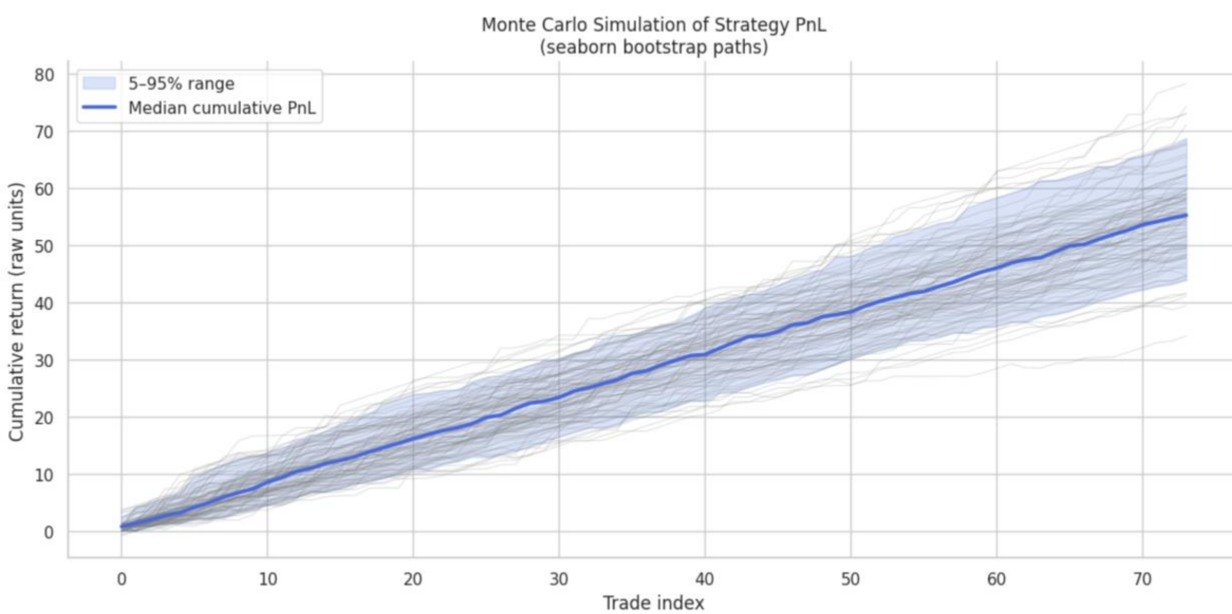


Figure 2, Monte Carlo Simulation of TDA Mean Reversion with Stop Loss and Take Profit Levels

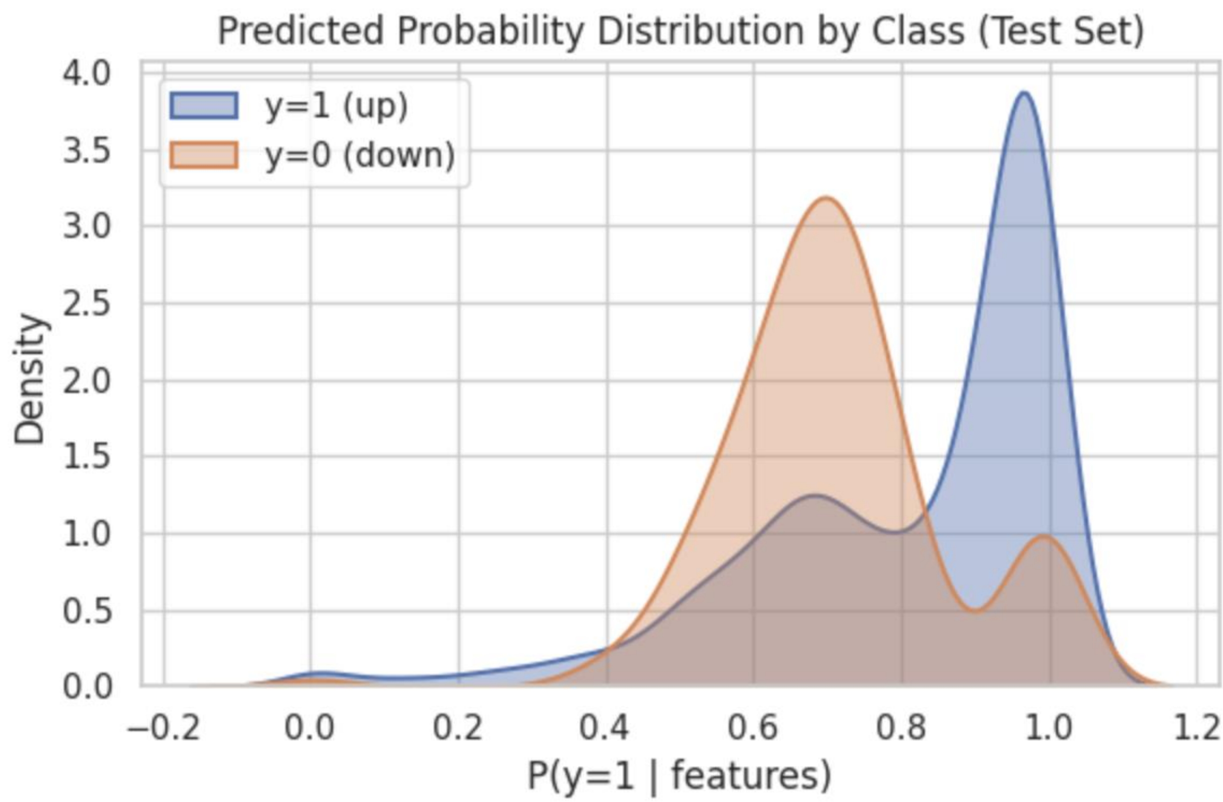


Figure 3, Logistic Regression Classification Distribution

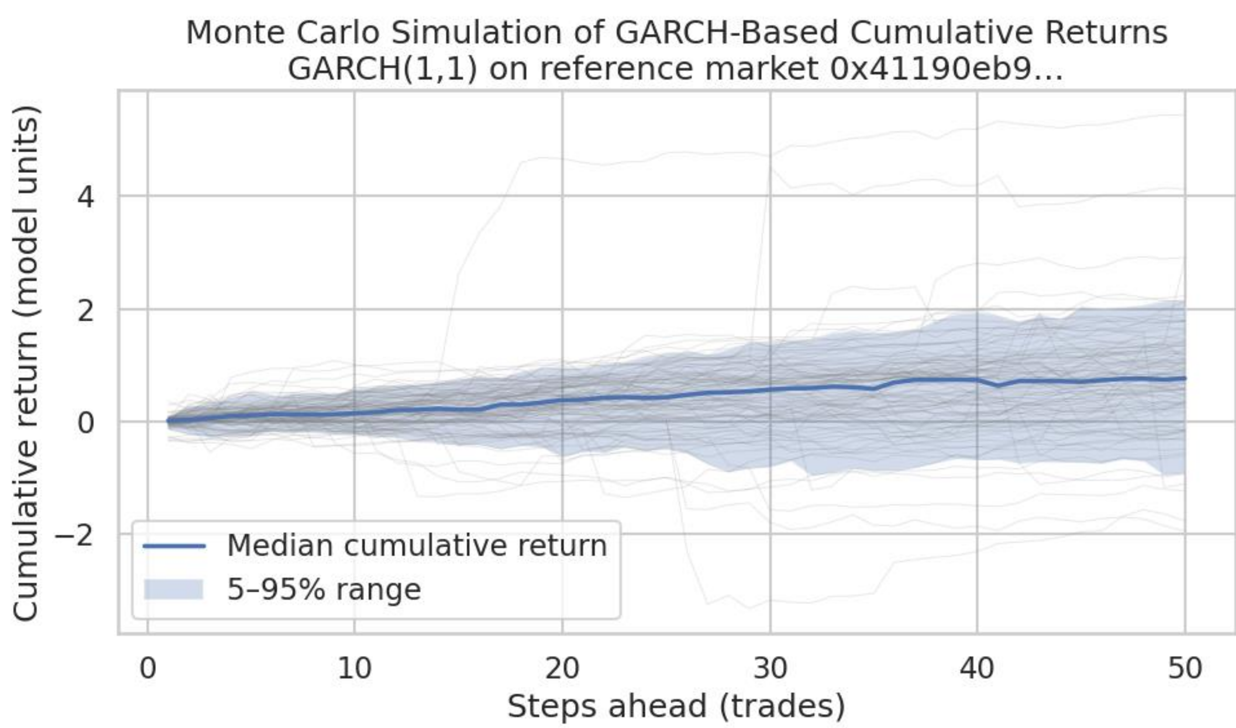


Figure 4, Modeling Volatility 50 Trades Ahead