Python Code Logic Report

Namit Shrivastava
January 6, 2025

I developed a comprehensive backtesting framework to evaluate a Time-Weighted Average Price (TWAP) strategy for cryptocurrency trading, specifically focusing on BTC and ETH.

Market Data Simulation Architecture

Firstly, I created a MarketDataGenerator class that generates synthetic price data using geometric Brownian motion and implemented realistic market microstructure features such as:

- Time-varying volatility patterns.
- Bid-ask spreads of 0.05%.
- Volume profiles that mirror typical crypto market behavior.
- Multi-venue price discovery across 3 simulated exchanges.

TWAP Strategy Implementation

Next part I did was to design a systematic order slicing mechanism that:

- Divides the total order quantity into 60 equal-sized child orders.
- Distributes executions evenly across the 6-hour trading window.
- Implements smart venue selection for optimal execution.

Keeping in mind that the trading window is from 9:30 AM to 3:30 PM with 1-minute interval.

Execution Results and Conclusions

My backtesting simulation of the TWAP strategy demonstrated promising results across both BTC and ETH. For Bitcoin, the strategy achieved an average execution price of $35,182.78 compared to the VWAP benchmark of $39,393.36, resulting in an execution cost of -1068.9 bps. Similarly, Ethereum showed strong performance with an average execution price of $1,675.01 against a VWAP benchmark of $1,884.66, yielding an execution cost of -1112.4 bps.

Hence, these negative execution costs across both assets indicate that my implementation successfully captured favorable price points below the VWAP benchmark, validating the effectiveness of the strategy in managing market impact through systematic order slicing.

[Comments are written in the code for easier understanding]