Week 2 Capstone Progress Report

PW26_MG_02

Hybrid HFT System Design and Explainability Roadmap

During Week 2 of the capstone project, substantial conceptual and architectural progress was made toward designing a production-grade hybrid High-Frequency Trading (HFT) system. The proposed system uniquely blends the low-latency execution advantages of C++ with the analytical flexibility and rapid experimentation capabilities of Python. The core architecture is structured around a six-layer quantitative model pipeline, encompassing statistical models, machine learning techniques, deep learning frameworks, reinforcement learning strategies, execution models, and market microstructure analysis. In addition, the system explicitly integrates ten critical trading enhancements, including signal normalization, transaction cost–aware filtering, drawdown-based kill switches, and realistic backtesting constraints. These design decisions are aimed at ensuring robustness, risk awareness, and real-world deployability. In the upcoming weeks, this trading engine will be evolved into the foundation of an explainability framework, enabling in-depth analysis and interpretation of strategy behavior, particularly under highly volatile market conditions, which is a central objective of the capstone project.

Polyglot Architecture Exploration and Integration

Following the core system design, a focused effort was made to explore the feasibility of a polyglot architecture supporting multiple programming ecosystems, including Python, C++, Java, R, and the MERN stack. A small-scale prototype was developed to validate interoperability among these heterogeneous components. Each language-specific module was designed to produce structured outputs, which were then standardized and merged using JSON as a common interchange format. This approach facilitated seamless communication across modules while retaining the individual strengths of each language, such as Python's modeling flexibility, C++'s performance efficiency, Java's stability, R's statistical capabilities, and MERN's suitability for web-based interfaces. Although exploratory, this work established a strong foundation for a modular, extensible system architecture capable of supporting complex quantitative workflows.

Live Data Collection and Architecture Development

In parallel with architectural experimentation, a live data collection process was actively designed and implemented during this week. The objective was to establish a reliable, production-ready pipeline capable of continuous data ingestion for real-time analysis and validation. Live market data collection commenced on 14th January and has been consistently operational through 17th January. The pipeline emphasizes data accuracy, precise timestamping, and structured storage to ensure compatibility with downstream modeling, backtesting, and execution components. **In subsequent weeks of the capstone project, this pipeline will remain active with a primary focus on continuous data collection, ensuring the availability of consistent, real-world datasets for analysis and validation, without introducing additional architectural changes during this phase.** This live data architecture is a critical component of the capstone, as it enables realistic evaluation of system performance under real market conditions rather than relying exclusively on historical or simulated datasets.