StockPredictorAI Quant System v3.0: Event-Driven Machine Learning Architecture

Author: StockPredictorAI Quantitative Research Team
Date: December 2025
Version: 3.0 (Release Candidate)

Abstract

This paper details the architecture of **Quant System v3.0**, an advanced event-driven stock prediction engine designed to capitalize on volatility windows surrounding earnings announcements. Building upon the foundational linear regression models of v1.0, this iteration introduces a **Gradient Boosted Decision Tree (XGBoost)** ensemble, "Days-Until-Event" temporal feature engineering, and a strict Nasdaq-100 macro-correlation constraint. The system moves from a continuous daily prediction model to a high-precision, scheduled inference engine, reducing noise and enhancing signal-to-noise ratio during critical market catalysts.

---

1. Introduction

Financial markets exhibit non-stationary behavior, often rendering static statistical models obsolete. However, corporate earnings events provide recurring, high-volatility windows where institutional behavior becomes more predictable due to portfolio rebalancing and consensus benchmarking.

Quant System v3.0 shifts our predictive paradigm from "General Market Direction" to "Event-Driven Opportunity Detection." By isolating the T-14 to T-1 pre-earnings window, the model learns specific price action signatures—such as "Run-Up" momentum or "Quiet Period" mean reversion—that precede major volatility events.

---

2. Methodology

2.1 The Core Algorithm: XGBoost

We utilize **Extreme Gradient Boosting (XGBoost)** as the primary regressor. Unlike linear models, XGBoost captures non-linear relationships between technical indicators and forward returns.

$$ \hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F} $$

Where:

- $\hat{y}_i$: Predicted 5-day Forward Return.
- f_k: Regression tree functions.
- $\mathcal{F}$: Space of regression trees.

The objective function minimizes the regularized squared error:

$$\mathcal{L}(\phi) = \sum_i (y_i - \hat{y}_i)^2 + \sum_k \Omega(f_k)$$

2.2 Feature Engineering (The Alpha Vectors)

The input vector X_t is constructed from 9 unique dimensions:

1. **Days_Until (d_t)**: Temporal proximity to next earnings. This allows the model to differentiate between "Mid-Quarter" drift and "Pre-Earnings" hype.
2. **Hype Factor (H_t)**: Cumulative Abnormal Return (CAR) over the trailing 30 days relative to QQQ.

$$H_t = \sum_{j=t-30}^{t} (R_{stock,j} - R_{QQQ,j})$$

3. **V_Rev (V_{rev})**: Volatility Mean Reversion interaction term.

$$V_{rev} = R_{lag1} \times \sigma_{5d}$$

4. **Sympathy Score (S_t)**: Real-time correlation strength with peer group (e.g., NVDA $\leftrightarrow$ AMD).
5. **Macro Trend (M_t)**: Constraint variable derived from Nasdaq-100 SMA(20) slope.

3. System Architecture

3.1 Training Pipeline (Quarterly Cycle)

To optimize computational resources and prevent overfitting to microstructure noise, the model undergoes a full retraining cycle **Quarterly**.

- **Training Window:** 2 Years of historical data.
- **Event Filter:** Only rows where $Days_Until \in [1, 14]$ are used for training. This enforces the "Event-Driven" specialization.
- **Persistence:** Trained boosters are serialized to JSON for rapid inference.

3.2 Inference Pipeline (Daily Cycle)

The system runs a lightweight inference pass daily:

1. **Schedule Check:** Queries Nasdaq API. If $T_{event} > 7$, the system sleeps for that ticker.
2. **Macro Gatekeeper:** If $M_t < 0$ (Bear Market) and Model Signal > 0 (Buy), the signal is dampened by a penalty factor $\lambda = 0.7$.
3. **Execution:** If $|Predicted_Return| > 1.2$, a signal is published to the platform.

---

4. Performance & Validation

4.1 "Zero-Shot" Calibration

Upon initialization, v3.0 operates in "Zero-Shot" mode, utilizing a pre-trained base on the "Magnificent Seven" tech stocks before fine-tuning on individual tickers.

4.2 Metrics

- **RMSE (Root Mean Square Error):** Primary loss metric during training.
- **Directional Accuracy:** Percentage of predictions where $sign(\hat{y}) == sign(y_{actual})$.

---

5. Conclusion

Quant System v3.0 represents a significant leap in our algorithmic capabilities. By constraining the model to high-probability event windows and introducing non-linear decision trees, we aim to deliver "Quality over Quantity"—providing actionable, high-conviction insights for the modern trader.
