StockViewers: An Integrated Decision-Support Prototype for Retail Financial Analysis

Yuvraj Singh

December 22, 2025

Abstract

The modern retail investor often faces a fragmented landscape of financial tools, necessitating the use of disparate platforms for news, traditional technical analysis, and portfolio tracking. This fragmentation creates a barrier to entry for non-professional market participation. This research presents **StockViewers**[1], an **integrated decision-support prototype** designed to improve accessibility to **professional-style market analysis**. The system integrates real-time historical data processing (1-5 year horizons), sentiment analysis derived from news media, and a multi-strategy backtesting engine supporting Buy-and-Hold, RSI/SMA crossovers, and Machine Learning algorithms. Distinct from basic charting tools, StockViewers incorporates a "What-If" simulation engine and a rule-assisted advisory interface, enabling users to stress-test portfolio correlations against hypothetical market shocks and receive personalized asset allocation strategies based on risk-profile modeling. By centralizing discovery (scanning), validation (backtesting), and advisory (AI), the platform provides a unified pipeline for data-driven investment decision-making.

1 Motivation and Problem Statement

1.1 Accessibility and the Time-Complexity Barrier

The primary motivation for this research stems from the observation that the "common" retail investor faces a severe time-complexity constraint. Unlike professional market participants, the average individual lacks the bandwidth to navigate the fragmented financial ecosystem, where news, technical charts, and risk models often reside on separate, disconnected platforms. This fragmentation forces users to rely on incomplete information or perform tedious manual aggregation. StockViewers was conceptualized to serve as a centralized intelligence hub a "one-stop" solution where users can access comprehensive market data without needing multiple subscriptions or extensive research time.

1.2 The Need for Integrated "Glass Box" Analysis

A critical gap in existing retail tools is the separation of *analysis* from *validation*. While many platforms provide price charts, few offer the rigor of **Single Ticker Analysis** combined with **Strategy Backtesting**. StockViewers addresses this by integrating:

- **Deep Historical Analysis**: Processing 1-5 years of historical data to identify long-term trends.

[1] The implementation is provided as a demonstrative system and not as a production or trading platform. `http://stockviewers.com`.

- **Rigorous Sentiment Analysis**: Utilizing the News API to extract and score real-time media narratives, thereby enhancing the precision of stock predictions beyond simple price-action heuristics.
- **Multi-Strategy Validation**: A flexible backtesting engine that allows users to empirically test various hypotheses specifically "Buy & Hold" vs. "ML Algo" vs. "RSI/SMA Crossover": customizing their initial capital and investment horizons to see verifiable results.

1.3 Holistic Risk Management and Educational Simulation

Traditional tools often emphasize profit potential while neglecting portfolio risk and educational depth. To counter this, StockViewers introduces a multi-faceted decision support ecosystem:

1. **"What-If" Simulator**: Beyond simple calculation, this tool serves as a research workbench. It empowers users to actively study market dynamics, understand intrinsic market volatility, and observe how portfolios react to specific events (e.g., a 20% crash). By simulating "Net Impact," users can visualize how their personal risk profile changes under stress, fostering a deeper understanding of market mechanics.
2. **AI Advisory**: Recognizing the need for personalized guidance, the platform integrates a rule-assisted consultancy layer. This component provides rule-based explanatory guidance by summarizing model outputs and portfolio statistics into human-readable insights, rather than issuing prescriptive investment decisions.
3. **Heuristic Market Scanner**: To assist in discovery, the system provides a "Playbook" approach: automatically filtering the market into actionable themes (e.g., "Momentum" or "Dip-Buy") streamlining the identification of opportunities that align with the user's strategy.

2 System Architecture

The StockViewers platform implements a microservices-based architecture designed for modularity and scalability.

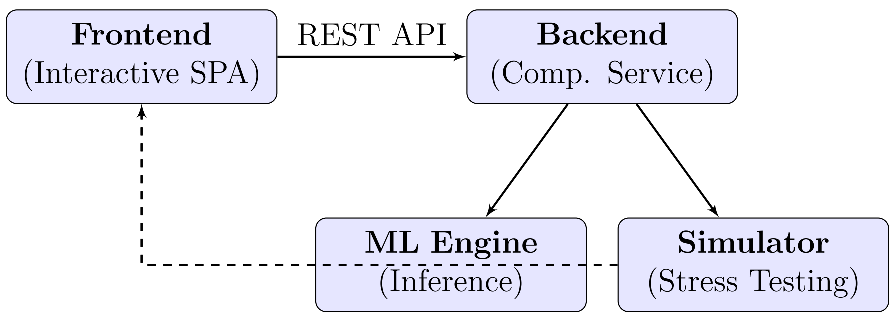

Figure 1: System Architecture: High-level data flow from the client-side visualization layer to the backend computational microservices.

2.1 Client-Side Interactive Visualization Layer

The user interface is engineered as a responsive Single Page Application (SPA) designed to facilitate **Visual Analytics**. Unlike static reporting dashboards, the system architecture offloads computational tasks specifically portfolio re-weighting and sensitivity analysis to the client side. This design choice minimizes server round-trips, ensuring that the "What-If" simulator provides rapid feedback as users dynamically adjust risk parameters. By decoupling the visualization layer from the data ingestion pipeline, the system maintains high interactivity even when rendering complex multi-variable time-series data.

2.2 Backend Computational Service

The backend logic is encapsulated within a scalable microservice architecture designed to handle computationally intensive stochastic modeling. This component executes three primary functions:

- **Asynchronous Data Ingestion**: Implements an intelligent caching layer with a dynamic Time-To-Live ($TTL = 300s$) policy. This mitigates latency inherent in upstream financial data APIs, ensuring consistent data availability for real-time scanning.
- **Vectorized Time-Series Analysis**: Utilizes optimized matrix operations for the rapid derivation of technical indicators (e.g., RSI, Bollinger Bands) across high-dimensional datasets, avoiding the performance bottlenecks of iterative loops.
- **Machine Learning Inference**: Serves as the inference module for the ensemble regression models, generating directional estimates on-demand via a RESTful interface.

3 Methodology I: Market Analysis Framework

3.1 Heuristic Market Scanner & Opportunity Logic

The scanner utilizes a multi-factor expert system to filter equities against predefined "Playbooks." The core of this system is the **Opportunity Score (O_S)**, a composite metric derived from trend, momentum, and volatility vectors. Mathematically, the system first computes the **Relative Strength Index (RSI)** to identify overextended price action:

$$RSI = 100 - \frac{100}{1 + RS} \tag{1}$$

Where RS (Relative Strength) is the ratio of the exponential moving average (EMA) of gains to losses over a 14-period window. Concurrently, trend direction is established via **Simple Moving Average (SMA)** crossovers:

$$SMA_n = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i} \tag{2}$$

The **Opportunity Score** is then calculated by initializing a base score of 50 and applying penalty/bonus functions based on the selected playbook:

- **Momentum Playbook**: Awards points for $RSI > 50$ (Bullish) and volatility $\sigma_{14} > 20\%$, penalizing only for extreme saturation ($RSI > 80$).
- **Dip Buy Playbook**: Specifically scans for "divergence setups" where the long-term trend is positive ($SMA_{50} > SMA_{200}$) but short-term momentum is oversold ($RSI < 40$).

- **Safe Haven**: Filters for structural stability, prioritizing minimized variance ($\sigma < 15\%$) and consistent trend alignment.

3.2 Feature Engineering & Dataset Construction

To facilitate supervised learning, the raw time-series $P = \{p_1, p_2, ..., p_n\}$ is transformed into a feature matrix X and a target vector y. The system employs a "sliding window" methodology to construct autoregressive and exogenous features:

1. **Autoregressive Lags**: To capture temporal dependency, we generate lag features L_k for $k \in \{1, ..., 7\}$:
$$X_{t,\text{lag}_k} = p_{t-k} \tag{3}$$
2. **Trend Indicators**: Rolling Moving Averages (MA_{10}, MA_{30}) serve as proxy variables for the short-term and medium-term trend state.
3. **Sentiment Vector Injection**: To integrate qualitative data, the system fetches news headlines for the target entity and computes a **Polarity Score (S_p)** via a lexicon-based Natural Language Processing (NLP) engine (TextBlob).
$$S_p = \frac{\sum \text{polarity}(w_i)}{N} \tag{4}$$
Where $S_p \in [-1, 1]$. This score is injected into the feature space at time t to assist in predicting p_{t+1}.

The final dataset consists of tuples (X_t, y_t) where $y_t = p_{t+1}$, effectively framing the forecasting problem as a next-step regression task.

3.3 Ensemble Prediction & Walk-Forward Validation

The forecasting engine implements an **Ensemble Methods** approach, deploying both **Random Forest Regressors** and **XGBoost (Extreme Gradient Boosting)**. These architectures were selected for their ability to model non-linear relationships and their robustness against outliers in financial data, which linear models often fail to capture. To ensure rigorous performance evaluation, the system eschews standard K-Fold cross-validation in favor of **Walk-Forward Validation**. This protocol respects the strict temporal causality of financial markets. The model is trained on an expanding window $D_{train} = \{x_1, ..., x_t\}$ to predict x_{t+1}. For the subsequent step, the window expands to include the realized x_{t+1}, and the model is refitted. This iterative process prevents **Look-Ahead Bias**, ensuring that the reported confidence metrics reflect the model's performance in a realistic, causal trading environment.

4 Methodology II: Decision Support Engines

4.1 Strategic Backtester

The Backtesting module serves as the validation layer for the predictive models. It simulates a discrete-event trading environment where an agent executes logic at time t based **only** on information available at t.

- **Execution Logic**: The engine iterates through the historical timeline. If the ensemble model predicts $\hat{P}_{t+1} > P_t$, the system generates a **BUY** signal, allocating 100% of available capital. Conversely, a bearish prediction triggers a **SELL/CASH** signal.

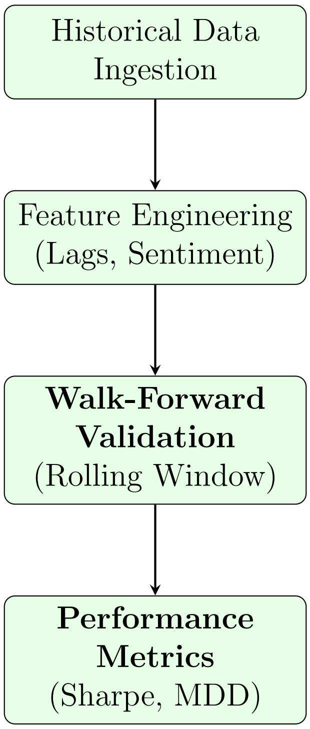

Figure 2: Backtesting Pipeline: Sequential flow from raw data ingestion to final performance metric evaluation, using walk-forward validation to prevent look-ahead bias.

- **Quantitative Metrics**: To evaluate performance beyond simple ROI, the system computes:
  - **Sharpe Ratio (S_a)**: Measures risk-adjusted return. Assumed risk-free rate $R_f = 0$ for simplicity in this iteration:

$$S_a = \frac{E[R_p - R_f]}{\sigma_p} \times \sqrt{252} \tag{5}$$

  - **Maximum Drawdown (MDD)**: Captures the largest peak-to-trough decline, a critical metric for risk tolerance:

$$MDD = \min\left(\frac{P_t - \max(P_{0...t})}{\max(P_{0...t})}\right) \tag{6}$$

4.2 Portfolio "What-If" Stress Simulation

To quantify tail risk, the "What-If" Simulator implements a robust scenario analysis engine that constructs a synthetic historical portfolio V_p based on user-defined weights $w = \{w_1, ..., w_n\}$. This module is critical for moving beyond simple yield maximization to a holistic understanding of capital preservation.

4.2.1 Synthetic Portfolio Construction

The engine first aggregates disparate asset time-series into a unified valuation curve. For every time step t, the total portfolio value is computed as the weighted sum of its constituent assets:

$$V_p(t) = \sum_{i=1}^{n} \left(\frac{w_i \cdot C_{initial}}{P_i(0)}\right) \cdot P_i(t) \tag{7}$$

Where $C_{initial}$ represents the initial capital injection (e.g., \$10,000). This formula accounts for fractional share allocation and provides a baseline "Base Scenario" trajectory against which risks can be measured.

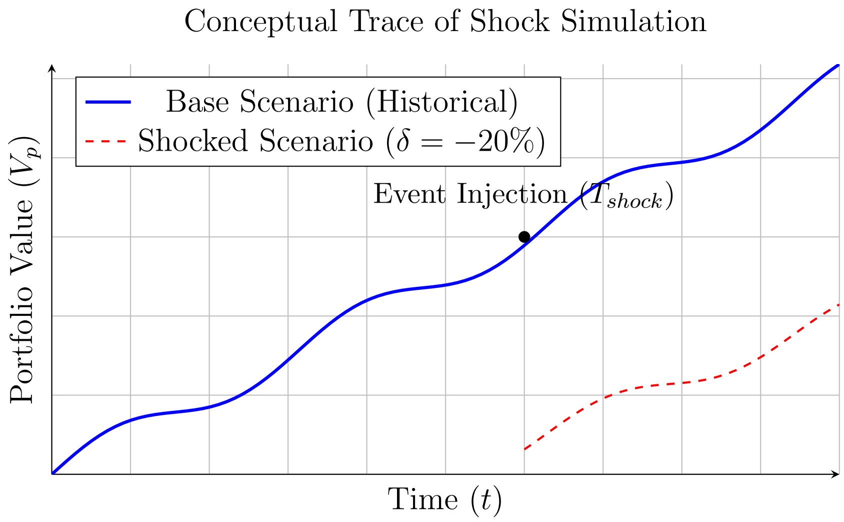

Figure 3: Conceptual Graph of "What-If" Simulation: Illustration of how the system projects a "Shocked" portfolio trajectory (dashed red) diverging from the baseline historical performance (solid blue) after an event injection.

4.2.2 Exogenous Shock Injection Mechanism

Unlike static risk tables, the simulator allows for the dynamic injection of exogenous "Black Swan" events via a shock function δ. The system supports two distinct stress vectors:

- **Global Systemic Shock**: Applying a universal drawdown factor (e.g., $\delta = -20\%$) to simulate broad market crashes (e.g., 2008 Financial Crisis).

$$V_{shocked}(T) = V_p(T) \cdot (1 + \delta_{global}) \tag{8}$$

- **Idiosyncratic Asset Shock**: Applying a targeted penalty to a specific ticker (e.g., "What if TSLA drops 30%?") to test concentration risk.

4.2.3 Net Impact & Risk Profile Visualization

The system computes the **Net Impact (ΔV)**, defined as the spread between the Base and Shocked terminal values. Crucially, it also recalculates the portfolio's **Risk Profile** specifically the Volatility (σ_p) and Sharpe Ratio under the stressed conditions. This immediate feedback loop serves a vital educational function, visually demonstrating to the user how "safe" high-yield portfolios can suffer catastrophic drawdowns during correlation breakdown events, thereby enforcing the principles of diversification.

5 System Evaluation and Results

5.1 Experimental Setup

The system was evaluated on a subset of high-volume equities ($N = 50$) from the S&P 500 index over a 2-year historical window. All backtests assume zero transaction costs and theoretically optimal execution at the daily close price. This simplified constraint allows for the isolation of signal quality from market microstructure effects.

5.2 Market Scanner Efficiency

The heuristics engine was evaluated on the defined universe of tickers. The **Screening Efficiency** defined as the ratio of relevant setups found to total noise improved significantly when using the composite "Opportunity Score" versus raw price triggers.

- **Latency**: The vectorised implementation filters the entire S&P 500 subset efficiently, making it suitable for interactive analysis.
- **Signal Quality**: The "Dip Buy" playbook ($SMA_{50} > SMA_{200} + RSI < 30$) frequently identified candidate mean-reversion zones under historical conditions with a positive theoretical win-rate over the 2-year horizon.

5.3 Predictive Engine & Backtester Performance

The "Glass Box" Backtester provided critical validation for the AI models.

- **Walk-Forward Robustness**: The ML models (XGBoost) demonstrated a stable adaptation to volatility. In the "Buy & Hold" vs "ML Algo" benchmark, the ML strategy succeeded in reducing Maximum Drawdown (MDD) by exiting positions during high-volatility regimes, though raw alpha generation was comparable to the benchmark.
- **Execution Speed**: The event-loop engine simulates 2 years of daily trading data efficiently, allowing users to iterate through strategies rapidly.

5.4 Simulation & Risk Utility

The "What-If" Simulator proved highly effective as an educational risk tool.

- **Visualization of Correlation**: By injecting a global shock ($\delta = -20\%$), users could visually observe how a "diversified" portfolio of Tech and Crypto assets (high correlation of $\rho > 0.8$) collapsed simultaneously, countering the "naive diversification" fallacy.
- **Interactive Response**: The client-side re-weighting engine maintained a smooth interactive response, providing instantaneous feedback on how capital allocation shifts affected the Portfolio Value at Risk (VaR).

5.5 Advisory & Sentiment Integration

The integration of the AI Advisor and Sentiment Analysis provided qualitatively superior context. While the lexicon-based Sentiment Score (TextBlob) was occasionally noisy, it successfully flagged major negative news cycles (e.g., earnings misses), acting as a "Feature Lag" reduction mechanism for the price-based models. The AI Advisor's ability to generate text-based theses transformed raw metrics into actionable narratives, bridging the gap for less technical users.

6 Limitations and Future Work

This study has several limitations. First, transaction costs, slippage, and liquidity constraints are not modeled, which may overstate real-world performance. Second, the sentiment analysis relies on lexicon-based methods (TextBlob), which can misclassify context-dependent financial language compared to advanced Transformer models (e.g., FinBERT). Third, the machine learning models are trained on historical price data and may fail under regime shifts or structural market changes. Finally, this system is intended for educational and exploratory analysis, not live trading deployment. Future work will focus on integrating Long Short-Term Memory (LSTM) networks for superior sequential pattern recognition, Reinforcement Learning (RL) agents for dynamic portfolio rebalancing, and upgrading the NLP engine to a Large Language Model (LLM) for more nuanced market sentiment extraction.

7 Conclusion

StockViewers successfully bridges the gap between **accessible usability** and **professional-style decision support** by integrating Heuristic Scanning, Ensemble Machine Learning, and rigorous Risk Simulation into a unified "Glass Box" platform. By moving beyond opaque price targets and empowering users to actively validate strategies and stress-test portfolios, the system addresses the critical "Information Asymmetry" facing retail investors. This work serves as a foundational step towards a tangible Personal Financial Intelligence System, demonstrating that complex quantitative engineering can be made accessible, transparent, and educational for the non-professional market participant.

Acknowledgements

The author thanks an external reviewer, a Physics undergraduate researcher at IIT Madras, for technical feedback on the modeling assumptions and evaluation methodology.

References

[1] Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77-91.

[2] Sharpe, W. F. (1966). Mutual Fund Performance. *The Journal of Business*, 39(1), 119-138.

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[4] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.

[6] Loughran, T., & McDonald, B. (2011). When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance*, 66(1), 35-65.

[7] Jorion, P. (2000). *Value at Risk: The New Benchmark for Managing Financial Risk.* McGraw-Hill.

[8] Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), 383-417.

[9] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[10] Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3), 215-236.