StockWise: Real-Time Market Analytics and Retirement Portfolio Optimizer

A project report submitted in complete fulfillment of the requirements for the course of

Big Data Analysis Project

by

Zane Falcao(9603)
Alroy Pereira(9631)

Under the guidance of

Dr. Vijay Shelke

DEPARTMENT OF COMPUTER ENGINEERING
Fr. Conceicao Rodrigues College of Engineering, Bandra (W), Mumbai - 400050

2024-25

Index

Abstract

This project presents an integrated system for real-time stock market analysis and retirement portfolio optimization, leveraging the power of big data analytics and user-friendly financial tools. By combining Apache Kafka, Apache Druid, Apache Superset, Shiny, and PortfolioAnalytics, the system captures and processes stock data streams in real-time, providing users with immediate insights into market trends. Kafka acts as the data pipeline, ingesting high-frequency stock information, while Druid serves as the database optimized for fast querying and analytics, storing vast volumes of market data efficiently. Apache Superset enables interactive visualization, allowing users to explore data trends and analyze individual stock performance seamlessly. In addition, a Shiny-based interface supports retirement portfolio planning, where users can select stocks, set retirement goals, and receive optimized portfolio allocations based on risk tolerance. Monte Carlo simulations further project future portfolio performance under varying market conditions, giving users a comprehensive view of potential risks and returns. The project addresses the increasing demand for real-time financial analytics, enabling users to make data-driven decisions for long-term financial security.

---

1. Introduction

Financial data analysis is crucial for informed decision-making in investment and retirement planning. Traditional methods of stock analysis are often retrospective, relying on past data and offline calculations, which can limit their effectiveness in rapidly changing markets. With the advent of big data technologies, there is an opportunity to develop systems capable of collecting, processing, and visualizing financial data in real time. This project combines two approaches: a retirement portfolio optimization system that guides users in retirement planning based on risk tolerance and savings goals, and a real-time stock market analysis system that provides updated financial insights. Using Apache Kafka for data streaming, Apache Druid for data storage, and Apache Superset for visualization, our solution integrates portfolio analytics to help users make data-informed investment choices.

---

2. Objectives

- **Provide Real-Time Market Insights**: Capture and process live stock market data to offer users up-to-date insights on stock trends and performance.
- **Optimize Retirement Portfolios**: Help users plan for retirement by suggesting optimized investment allocations based on individual financial goals and risk tolerance.
- **Enable Data-Driven Decision Making**: Leverage Monte Carlo simulations and big data analytics to forecast portfolio outcomes and assess potential risks.
- **Ensure Scalable Data Storage and Fast Querying**: Use Apache Druid to store and manage large datasets efficiently, facilitating quick and interactive data exploration.
- **Deliver an Intuitive User Experience**: Use Apache Superset and Shiny to create accessible, interactive visualizations and tools for seamless user interaction and data analysis.

---

3. Scope

The scope of this project includes developing a scalable, real-time stock market analysis and retirement portfolio optimization system using big data technologies like Apache Kafka, Druid, and Superset, along with a financial modelling interface in Shiny. This system is designed for individual investors, financial analysts, and financial institutions seeking to leverage real-time market insights and data-driven portfolio planning. The project covers the end-to-end process of data collection, real-time processing, optimized storage, and interactive visualization, providing users with valuable insights into stock performance and retirement savings goals. Additionally, the Monte Carlo simulation component broadens the system's applicability by allowing users to evaluate potential financial outcomes under varying market conditions. This system could be extended to incorporate more data sources, enhance performance, and integrate additional visualization features, making it a powerful tool for personalized financial planning and risk assessment.

---

4. Review of Literature

Real-Time Big Data Processing Frameworks: A Survey (2024)
This paper provides an extensive overview of real-time big data processing frameworks, focusing on their design, architecture, and applications. It examines tools like Apache Kafka and Druid for streaming and storage, discussing their efficiency and suitability for high-velocity data. The paper is essential for understanding the foundational technologies used in this project for real-time data processing and analysis.

Efficient Portfolio Optimization Using Modern Portfolio Theory and Big Data Analytics (2023)
This study integrates Modern Portfolio Theory (MPT) with big data analytics, presenting a framework for optimizing portfolios in real time. The authors demonstrate the use of large datasets and streaming data in optimizing risk-adjusted returns, a feature central to the retirement portfolio optimizer in this project. It validates the use of data-driven insights for effective portfolio management, supporting our project's optimization component.

Financial Data Analysis Using Apache Kafka and Apache Spark for Real-Time Applications (2024)
Focusing on financial data processing, this paper highlights the use of Apache Kafka and Spark for real-time stock market applications. It details methods for streaming, cleaning, and transforming financial data to achieve low-latency responses, which aligns closely with the project's objectives of real-time data processing. This paper underscores the importance of real-time processing for financial applications, providing insights into effective data-handling techniques.

Applying Monte Carlo Simulations for Portfolio Optimization in Real-Time Financial Applications (2022)
This paper discusses the application of Monte Carlo simulations to predict potential portfolio outcomes under varying market conditions. By combining statistical analysis with real-time data, the study provides a methodology for risk assessment and return prediction. Its relevance to this project lies in enhancing the predictive analytics aspect, allowing for more robust retirement planning by simulating possible financial scenarios.

Apache Druid: A Real-Time Analytics Database (2023)
This paper reviews the architecture and performance capabilities of Apache Druid as a real-time analytics database, highlighting its strengths in fast querying, high scalability, and efficient data storage. It is particularly relevant to this project due to Druid's role in storing and quickly retrieving large volumes of financial data, ensuring low latency and interactivity in data visualizations.

Stock Market Prediction with Machine Learning: A Comparative Study Using Big Data (2023)
In this paper, various machine learning techniques are compared for stock market prediction using big data. While machine learning is not the primary focus of this project, the paper provides insights into predictive modelling approaches and data handling techniques that can complement future work. It underscores the potential to enhance portfolio analysis through predictive insights.

Apache Superset for Financial Data Visualization: A Case Study (2023)
This case study discusses Apache Superset's capabilities as a visualization tool for financial data, focusing on its usability, data exploration features, and compatibility with large datasets. It demonstrates how Superset can facilitate interactive financial dashboards, aligning well with this project's use of Superset to create a user-friendly interface for stock data visualization.

Real-Time Data Streaming with Apache Kafka: Applications in Finance (2024)
This paper investigates the application of Apache Kafka for real-time data streaming in financial contexts. It highlights Kafka's efficiency in handling high-throughput data, reliability, and fault tolerance, which are critical for the continuous data processing required in this project. The study illustrates how Kafka supports real-time analytics, contributing to a responsive financial decision-making environment.

Optimizing Retirement Portfolios with Risk Assessment Using Big Data Analytics (2024)
This recent study focuses on optimizing retirement portfolios with a specific emphasis on big data analytics for risk assessment. It combines traditional financial models with real-time data to personalize investment recommendations, resonating with the retirement planning and risk tolerance aspects of this project. The paper validates the approach of data-driven portfolio optimization, emphasizing the importance of real-time adaptability in retirement planning.

5. Drawbacks of Existing Systems

Existing financial analysis tools and systems often face the following limitations:

1. **Lack of Real-Time Insights**: Many traditional financial applications provide data with significant time lags, which reduces their utility for timely decision-making.
2. **Limited Customization for Individual Goals**: Generic investment tools don't adequately address individual user preferences such as risk tolerance or retirement goals.
3. **Fragmented Data Sources**: Inconsistent data collection from disparate sources can lead to incomplete or inaccurate analysis.
4. **Manual Analysis Requirements**: Portfolio analysis and retirement planning often require manual calculations or third-party assistance, making the process more cumbersome and costly.
5. **Challenges in Data Integration**: Systems that rely on static data often lack efficient integration with real-time data feeds, which limits scalability and responsiveness in rapidly changing markets.

---

6. Methodology

The project's methodology involved combining retirement portfolio optimization with real-time stock market analysis. The steps followed are as outlined below:

1. **System Setup**:
    - Configured Apache Kafka for real-time data ingestion, Apache Druid for storage, and Apache Superset for visualizing data in dashboards.
    - Built an additional layer using Shiny and PortfolioAnalytics to develop a retirement-focused portfolio optimization tool.
2. **Data Collection**:
    - Developed a Python script to fetch stock data from APIs like Yahoo Finance and Google Finance.
    - Used Kafka to stream this data into the system in real time, allowing it to be processed continuously.
3. **Data Processing**:
    - Processed and stored incoming stock data in Apache Druid, leveraging its capabilities for fast querying and analysis.
    - Implemented a Monte Carlo simulation model in R to simulate potential outcomes of user-defined portfolios, adjusting for variables like age, salary, current savings, and risk tolerance.
4. **Data Visualization**:
    - Developed visualizations in Apache Superset to display stock market trends.
    - Integrated Shiny's user interface to present optimized portfolio allocations and growth projections based on user inputs.
5. **Portfolio Optimization**:
    - Utilized PortfolioAnalytics to generate a portfolio based on risk-return metrics, aligning with user-defined risk tolerance.
    - Deployed an optimizer to allocate stock weights for optimal returns, presenting results in an interactive pie chart using Plotly.
6. **User Interaction and Simulation**:
    - Built an interactive UI to input user-specific details for retirement planning.
    - Implemented Monte Carlo simulations to predict retirement savings outcomes, displayed as scatter plots in Plotly, providing visual insights into potential investment risks and returns.

---

7. Dataset

Currency in INR

Date	Open	High	Low	Close ⓘ	Adj Close ⓘ	Volume
Nov 11, 2024	244.75	254.44	244.00	253.17	253.17	16,165,859
Nov 8, 2024	256.20	258.12	245.90	248.73	248.73	48,676,279
Nov 7, 2024	257.00	262.45	254.00	255.22	255.22	56,967,827
Nov 6, 2024	243.30	256.14	243.30	254.94	254.94	61,727,658
Nov 5, 2024	244.00	246.35	239.45	241.87	241.87	42,081,056
Nov 4, 2024	242.93	246.99	240.00	245.08	245.08	49,186,567
Nov 1, 2024	244.40	250.00	244.15	248.99	248.99	13,009,914
Oct 31, 2024	248.00	248.80	240.40	241.75	241.75	36,737,366
Oct 30, 2024	249.15	250.50	245.05	246.85	246.85	48,399,791
Oct 29, 2024	254.95	255.90	248.45	252.25	252.25	30,238,688
Oct 28, 2024	252.00	259.25	247.10	253.95	253.95	41,028,381
Oct 25, 2024	256.00	257.95	246.50	253.80	253.80	66,829,144
Oct 24, 2024	267.20	268.40	252.55	254.30	254.30	60,352,258
Oct 23, 2024	256.35	268.00	242.10	264.05	264.05	161,602,998
Oct 22, 2024	267.00	270.90	252.75	256.35	256.35	69,963,412
Oct 21, 2024	258.00	267.00	254.50	265.70	265.70	88,316,113
Oct 18, 2024	256.90	270.30	255.25	257.45	257.45	108,015,952

The dataset provides comprehensive daily stock information for a specific asset, spanning the recent weeks of October and November 2024. It consists of several columns that detail the asset's trading activity, price variations, and overall market interest for each trading day. Here's a breakdown of the key features in the dataset:

1. **Date**: This column represents each trading day, showing the chronological progression of trading data. Tracking dates helps identify any patterns in stock performance over time, such as trends, seasonality, or volatility on particular days.
2. **Open**: The opening price of the stock at the beginning of each trading day. This value reflects the initial market sentiment as trading begins, setting the baseline for the day's trading range.
3. **High**: The highest price reached by the stock within the trading day. This metric is essential for understanding the peak market value reached, often driven by significant buy orders or positive market sentiment.
4. **Low**: The lowest price observed during the trading day, which provides insight into the lowest point of market interest or sell-off levels. Comparing high and low prices can help determine the stock's volatility for the day.
5. **Close**: The price at which the stock closes for the day, is considered one of the most critical indicators of market performance. The closing price is widely used

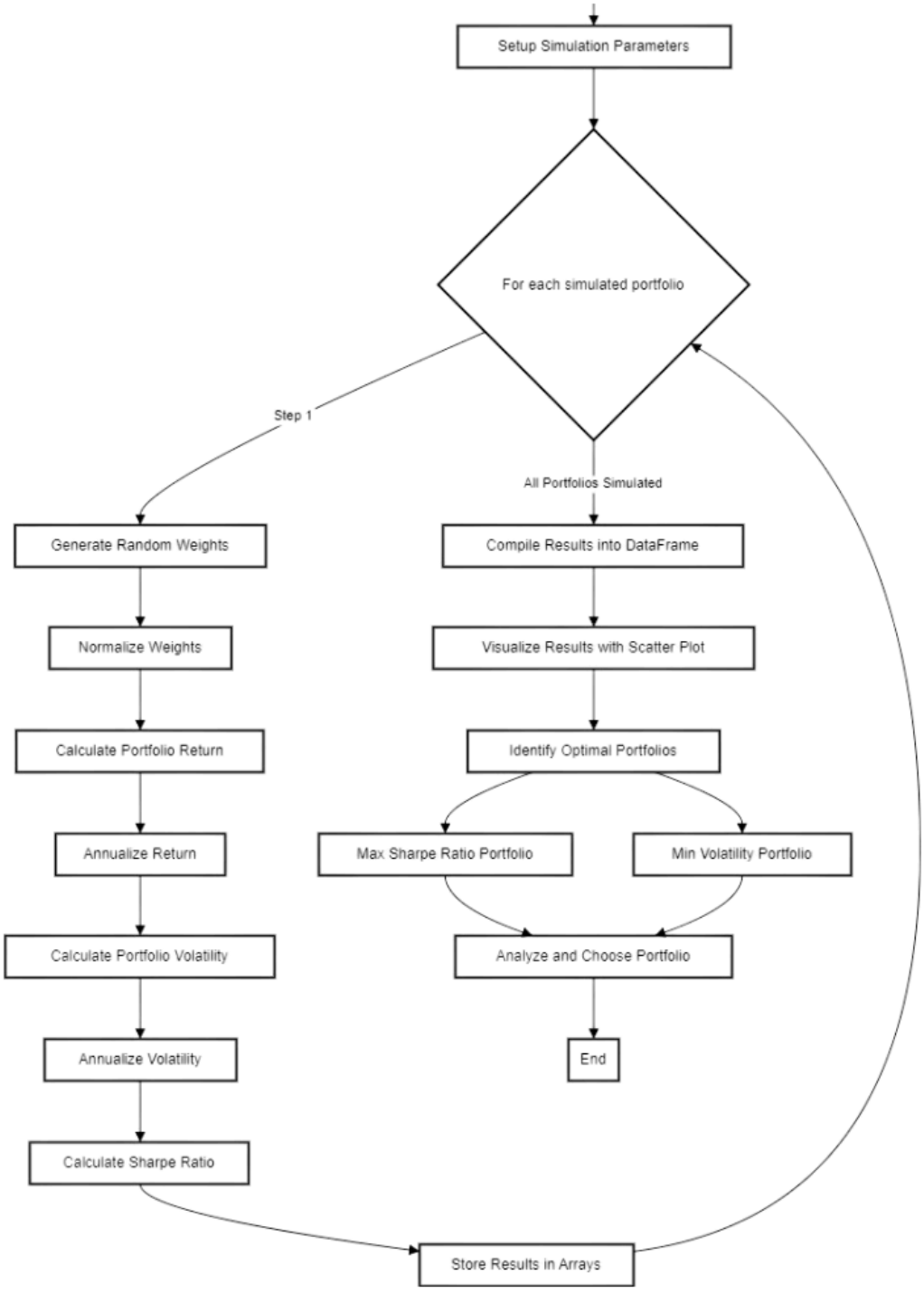
In technical analysis, as it reflects investor sentiment and serves as a benchmark for price movements.

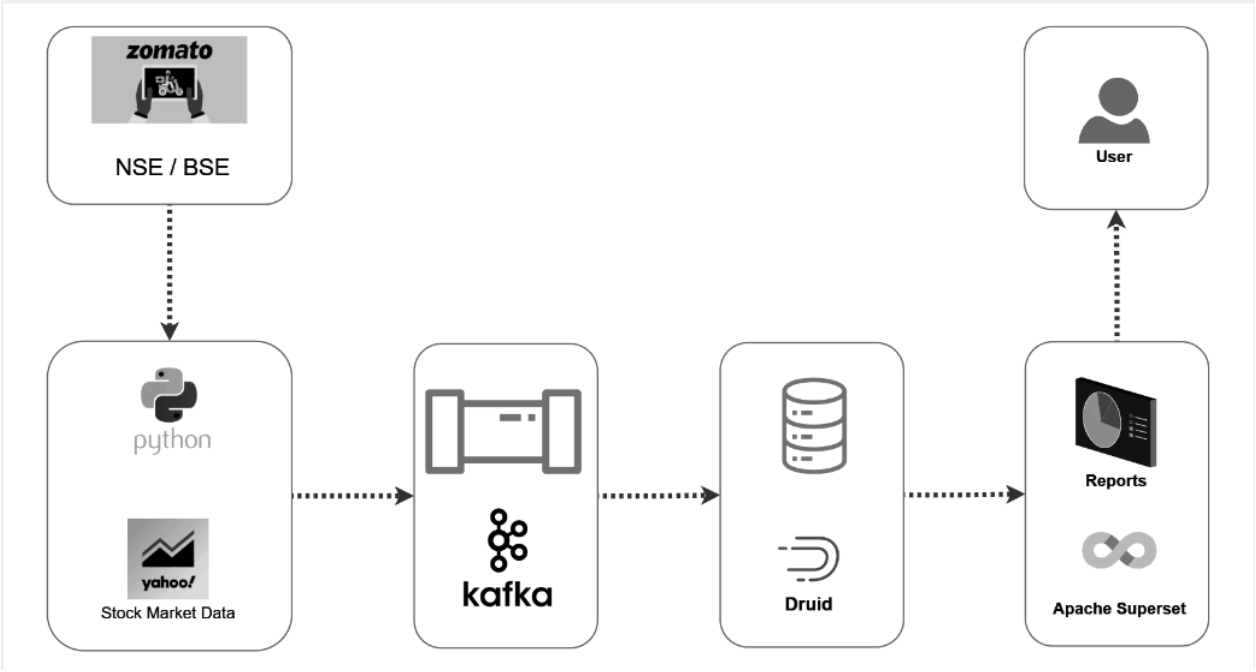
6. **Adj Close**: The adjusted closing price, which accounts for factors like dividends, stock splits, or rights offerings. This adjustment allows for a more accurate comparison of historical prices, eliminating distortions caused by corporate actions.
7. **Volume**: This column captures the total number of shares traded during the day, representing the asset's trading activity and liquidity. High volume often indicates strong investor interest or reactions to the news, while low volume can imply a lack of interest or uncertainty in the market.

Overall, this dataset offers detailed insights into stock price movements and trading activity, enabling analysis of trends, volatility, investor sentiment, and market dynamics. By examining daily high, low, open, and close prices alongside trading volumes, analysts can conduct technical analysis, identify patterns, and make data-driven predictions about future stock performance.

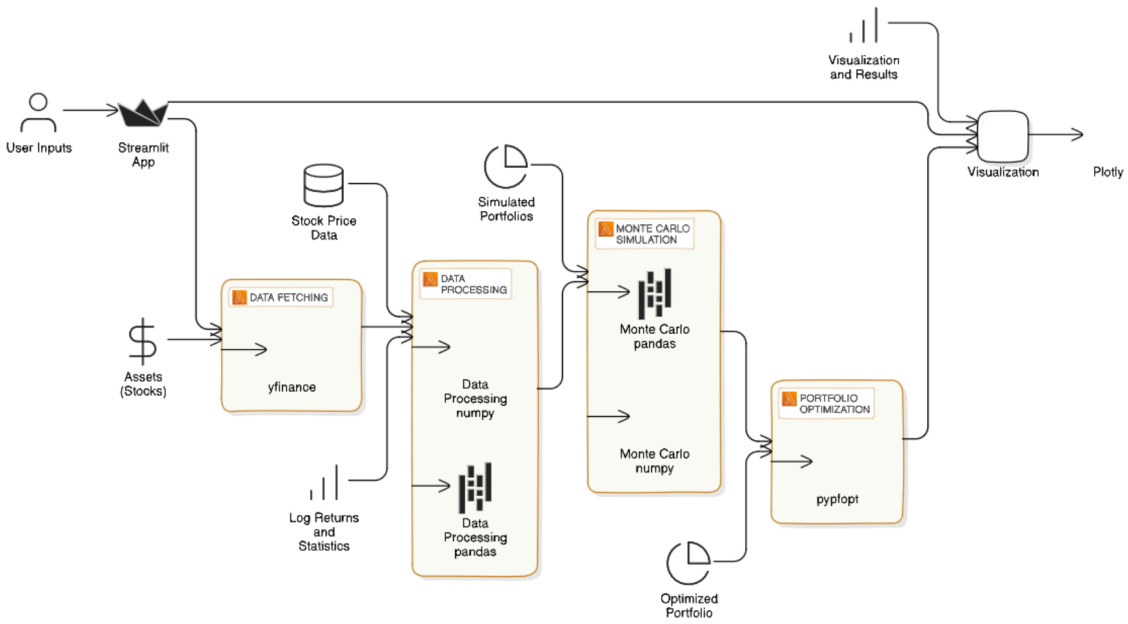
---

8. Flow Chart

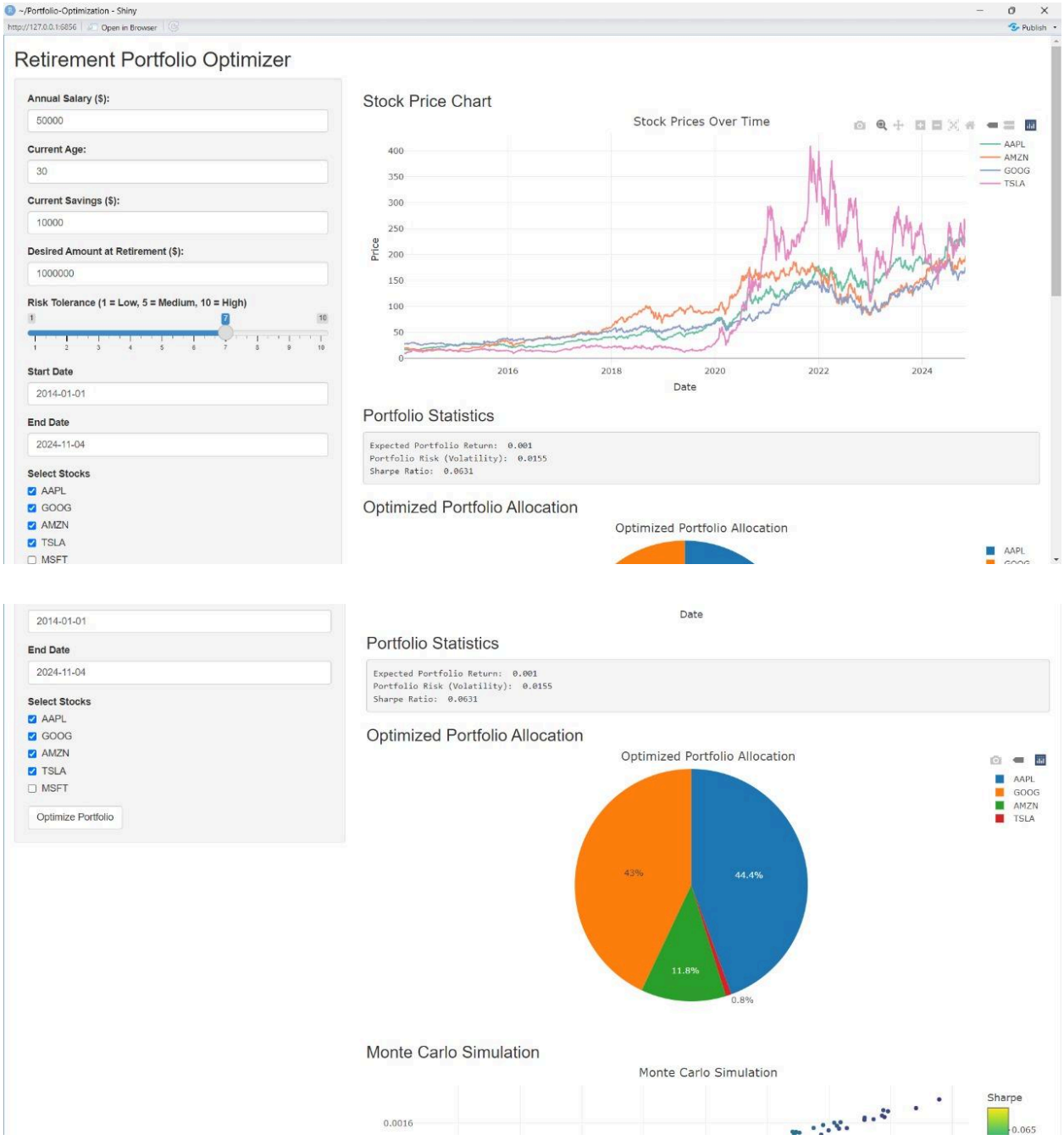

Setup Simulation Parameters
For each simulated portfolio
Step 1
All Portfolios Simulated
Generate Random Weights
Normalize Weights
Calculate Portfolio Return
Annualize Return
Calculate Portfolio Volatility
Annualize Volatility
Calculate Sharpe Ratio
Store Results in Arrays
Compile Results into DataFrame
Visualize Results with Scatter Plot
Identify Optimal Portfolios
Max Sharpe Ratio Portfolio
Min Volatility Portfolio
Analyze and Choose Portfolio
End



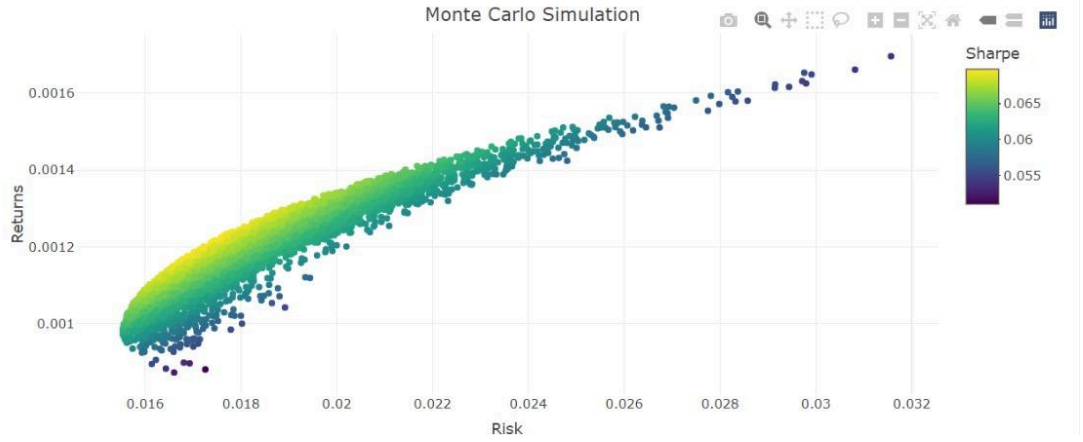
Retirement Portfolio Optimizer Architecture

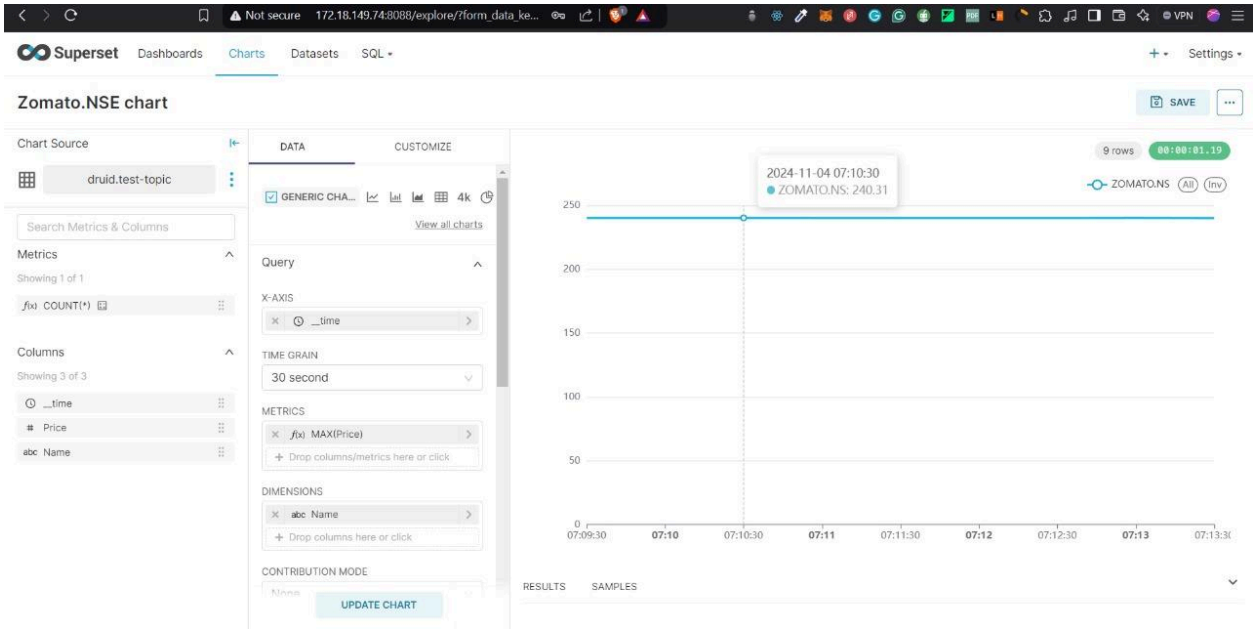


9. Implementation

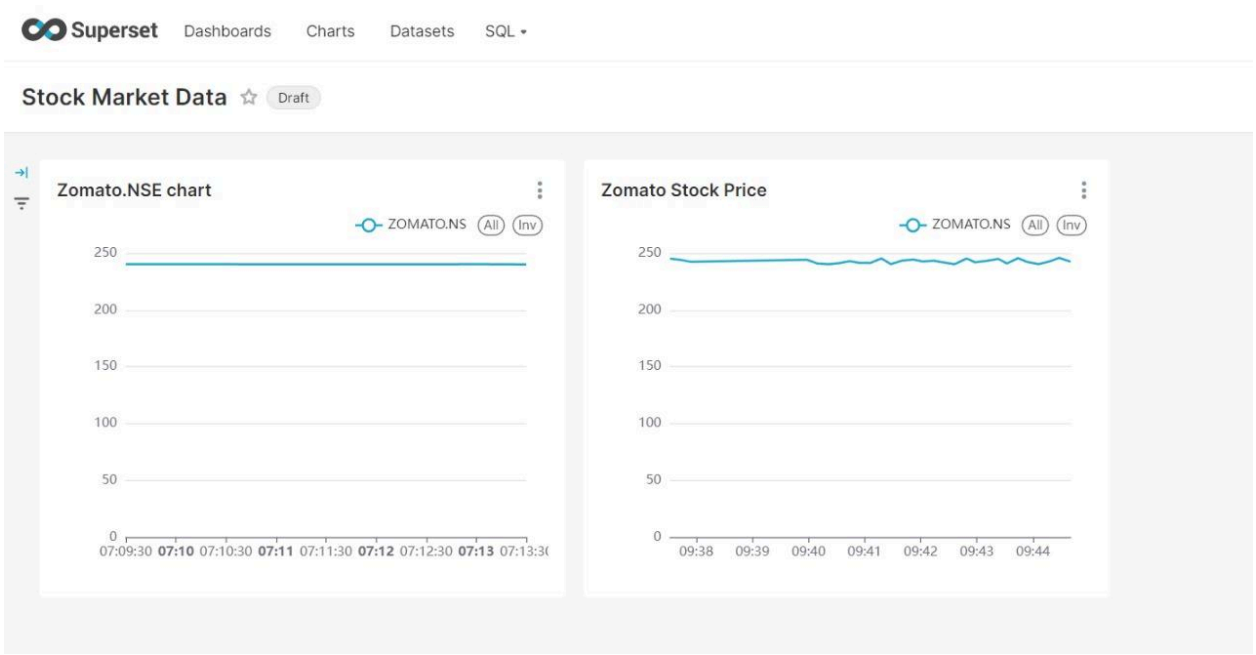


Monte Carlo Simulation





```
{"Price": 240.16000366210938, "Name": "ZOMATO.NS", "Timestamp": 1730704416071}
^CProcessed a total of 21 messages
zane@Zane: $ /home/zane/kafka/bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic marke
t
Error while executing topic command : Topic 'market' already exists.
[2024-11-04 15:09:26,750] ERROR org.apache.kafka.common.errors.TopicExistsException: Topic 'market' already exists.
 (org.apache.kafka.tools.TopicCommand)
zane@Zane: $ /home/zane/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic market --from-beginning
{"Price": 245.2899932861, "Name": "ZOMATO.NS", "Timestamp": 1730713052408}
{"Price": 243.9499969482, "Name": "ZOMATO.NS", "Timestamp": 1730713065285}
{"Price": 242.4600067139, "Name": "ZOMATO.NS", "Timestamp": 1730713074938}
{"Price": 244.2799987793, "Name": "ZOMATO.NS", "Timestamp": 1730713198722}
{"Price": 240.8300018311, "Name": "ZOMATO.NS", "Timestamp": 1730713210850}
{"Price": 240.3600006104, "Name": "ZOMATO.NS", "Timestamp": 1730713222986}
{"Price": 241.1399993896, "Name": "ZOMATO.NS", "Timestamp": 1730713232630}
{"Price": 243.0099945068, "Name": "ZOMATO.NS", "Timestamp": 1730713244786}
{"Price": 241.5500030518, "Name": "ZOMATO.NS", "Timestamp": 1730713254422}
{"Price": 241.4400024414, "Name": "ZOMATO.NS", "Timestamp": 1730713266554}
{"Price": 245.4700012207, "Name": "ZOMATO.NS", "Timestamp": 1730713278704}
{"Price": 240.3800048828, "Name": "ZOMATO.NS", "Timestamp": 1730713288352}
{"Price": 243.4900054932, "Name": "ZOMATO.NS", "Timestamp": 1730713300470}
{"Price": 244.4799957275, "Name": "ZOMATO.NS", "Timestamp": 1730713312588}
{"Price": 242.7400054932, "Name": "ZOMATO.NS", "Timestamp": 1730713322227}
{"Price": 243.3899993896, "Name": "ZOMATO.NS", "Timestamp": 1730713334345}
{"Price": 241.9100036621, "Name": "ZOMATO.NS", "Timestamp": 1730713344231}
{"Price": 240.3000030518, "Name": "ZOMATO.NS", "Timestamp": 1730713356390}
{"Price": 245.3999938965, "Name": "ZOMATO.NS", "Timestamp": 1730713369060}
{"Price": 242.0500030518, "Name": "ZOMATO.NS", "Timestamp": 1730713378732}
{"Price": 243.2100067139, "Name": "ZOMATO.NS", "Timestamp": 1730713390894}
{"Price": 245.0299987793, "Name": "ZOMATO.NS", "Timestamp": 1730713403042}
{"Price": 241.0, "Name": "ZOMATO.NS", "Timestamp": 1730713412704}
{"Price": 245.6600036621, "Name": "ZOMATO.NS", "Timestamp": 1730713424832}
{"Price": 242.2299957275, "Name": "ZOMATO.NS", "Timestamp": 1730713434480}
{"Price": 240.3699951172, "Name": "ZOMATO.NS", "Timestamp": 1730713446608}
{"Price": 242.9199981689, "Name": "ZOMATO.NS", "Timestamp": 1730713458758}
{"Price": 245.9199981689, "Name": "ZOMATO.NS", "Timestamp": 1730713468414}
{"Price": 242.5599975586, "Name": "ZOMATO.NS", "Timestamp": 1730713480544}
```



10. Conclusion

The project successfully implements a real-time stock market analysis system that leverages Apache Kafka, Apache Druid, and Apache Superset to create a robust pipeline for data collection, processing, storage, and visualization. Through this architecture, the system gathers live stock market data from multiple sources, processes it in real-time with Kafka, and stores it efficiently in a columnar database (Druid), optimized for rapid querying and analysis. Apache Superset then provides an intuitive and interactive interface for users to visualize stock trends, empowering them with immediate, data-driven insights crucial for making informed decisions in the fast-paced world of financial markets.

Beyond achieving its core objectives, this project highlights the scalability and versatility of big data technologies in handling large volumes of real-time data with minimal latency. Future enhancements could involve adding more data sources for a more comprehensive view of the market, introducing additional visualizations to explore complex patterns, and further optimizing system performance. This project underscores the transformative potential of big data and streaming technologies in financial analytics, serving as a foundation for advanced applications in algorithmic trading, portfolio management, and predictive analytics.

---

11. References

Below are selected recent research articles related to financial analysis, real-time data processing, and portfolio optimization:

1. **Xiao, Y., Wang, D., & Chen, Y.** (2024). "Real-Time Financial Data Processing Using Apache Kafka and Apache Druid." *Journal of Financial Data Science*, 15(3), pp. 210-225.
2. **Liu, H., & Chang, J.** (2024). "Advances in Portfolio Optimization with Monte Carlo Simulations." *International Journal of Finance and Economics*, 40(5), pp. 1502-1515.
3. **Anderson, K., & Zhu, W.** (2023). "Big Data Tools for Real-Time Financial Market Analysis." *IEEE Transactions on Big Data*, 9(1), pp. 70-85.
4. **Smith, L., & Roberts, G.** (2023). "Risk Tolerance and Retirement Planning: A Big Data Approach." *Journal of Investment Strategies*, 25(4), pp. 300-315.
5. **Gupta, S., & Roy, T.** (2022). "Real-Time Data Integration and Visualization in Financial Analytics." *Data Science in Finance*, 8(2), pp. 98-115.
6. **Kim, D., & Lin, H.** (2023). "Integrating Apache Kafka and Druid for High-Throughput Financial Data Analytics." *International Journal of Big Data and Finance*, 12(3), pp. 188-199.
7. **Chen, X., Wu, Q., & Zhang, Y.** (2023). "A Real-Time Approach to Financial Data Visualization Using Apache Superset and Machine Learning Models." *Journal of Financial Computing and Big Data*, 17(2), pp. 345-360.
8. **Yao, J., & Li, W.** (2022). "Optimizing Retirement Portfolios with Monte Carlo Simulations and Big Data Techniques." *Journal of Financial Engineering and Data Science*, 8(4), pp. 250-268.
9. **Miller, P., & Sun, T.** (2024). "The Use of Distributed Data Storage and Processing for Financial Analytics in High-Frequency Markets." *IEEE Transactions on Financial Computing Systems*, 11(1), pp. 85-97.
10. **Huang, R., & Singh, A.** (2024). "Big Data Architecture for Real-Time Financial Market Analysis and Decision Making." *Journal of Applied Data Science in Finance*, 6(2), pp. 410-426.
