e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

STOCK ANALYSIS WEB APPLICATION: A COMPREHENSIVE TOOL FOR REAL-TIME STOCK DATA ANALYSIS AND VISUALIZATION

Shreesh Gururaj Kulkarni*1, Nagashri JA*2, Sushma C*3, Poornima B*4
*1,2,3,4Department Of IS & E, B.I.E.T., Davangere, Karnataka, India.

DOI : https://www.doi.org/10.56726/IRJMETS65834

ABSTRACT

This paper presents the development and implementation of a Stock Analysis Web Application that addresses the challenges of accessing, analyzing, and visualizing stock market data. Leveraging Python libraries such as yfinance, pandas, NumPy, matplotlib, seaborn, and Streamlit, the application enables real-time data retrieval, statistical computations, and interactive visualizations. Designed for investors and financial analysts, it simplifies complex analyses through an intuitive interface, providing actionable insights for informed decision-making. The modular architecture ensures scalability and extensibility for future enhancements. This work contributes to the intersection of financial technology and computational finance by bridging raw data with meaningful investment insights.

Keywords: Stock Analysis, Real-Time Data, Visualization, Python, Financial Technology, Web Application.

I. INTRODUCTION

The stock market serves as a cornerstone of the global economy, influencing decision-making processes for investors, corporations, and governments. Analyzing stock data, however, often requires navigating vast datasets and applying advanced techniques, which can be daunting for many users. This paper introduces a **Stock Analysis Web Application** designed to streamline this process by integrating real-time data retrieval, statistical analysis, and intuitive visualizations. Built using Python's robust ecosystem, the application caters to novice and expert users alike, offering a dynamic platform for extracting actionable insights. The focus of this project is to address key challenges, including data accessibility, complexity in analysis, and user experience.

The application has been tailored to meet diverse user requirements by incorporating robust back-end and front-end architectures. It provides real-time access to stock data, supports comprehensive statistical analyses, and delivers user-friendly visualizations. This holistic approach ensures that even users with minimal technical expertise can leverage the application for informed investment decisions [1-10].

II. METHODOLOGY

1. System Design and Architecture

A. Backend Design

The backend is designed to efficiently handle data retrieval, processing, and analysis while ensuring seamless integration with the frontend. The data retrieval module uses the yfinance library to fetch both real-time and historical stock data, including adjusted closing prices to account for stock splits and dividends [2]. Data processing is conducted using pandas to perform operations such as normalization, summarization, and descriptive statistics, while NumPy handles numerical computations including volatility and daily returns [3,8]. These operations form the foundation for statistical analysis, where metrics like mean, median, standard deviation, percentiles, correlation matrices, and risk-return trade-offs are computed [3,8]. The visualization preparation module generates professional-grade plots using matplotlib and seaborn, ensuring that outputs are both insightful and visually appealing [5,6].

B. Frontend Design

The frontend leverages Streamlit to provide a responsive and interactive user interface [1]. Users can input stock tickers and dynamically view real-time results. The interface is designed to render charts and tables seamlessly, supporting features such as time-series analysis, volatility evaluation, and correlation matrix visualization [8]. Error handling mechanisms are implemented to provide clear and actionable feedback in cases of invalid inputs or unavailable data, ensuring a smooth user experience. Additionally, the frontend's architecture allows for real-time interactivity, enabling users to update inputs and instantly observe changes in the output [1].

III. MODELING AND ANALYSIS

1. Modeling

A. Technology Stack

The application utilizes Python libraries such as yfinance for data retrieval, pandas for data manipulation and analysis, and NumPy for numerical computations [2,3]. Streamlit is employed to create a responsive and interactive web application interface, while matplotlib and seaborn are used for generating aesthetically pleasing and informative charts [1,6].

B. Workflow

The workflow begins with users providing stock tickers. The ticker symbols uniquely identify the publicly traded companies on the stock exchange. The system then retrieves the corresponding data, performs statistical computations, and prepares visualizations. The final output is displayed on the Streamlit interface, where users can dynamically explore the data and interact with the visualizations [1,7,8].

2. Analysis

A. Descriptive Statistics Analysis

The application provides a summary of stock performance through metrics such as mean, standard deviation, and percentiles. These metrics offer valuable insights into trends, stability, and the distribution of stock prices. A high standard deviation indicates greater volatility, while closely grouped percentiles suggest stability [7,8].

B. Time-Series Analysis

Time-series analysis tracks historical price trends using line charts, enabling users to identify patterns over time. This feature is particularly useful for evaluating long-term stock behavior and predicting potential investment opportunities [7,8].

C. Volatility and Correlation Analysis

Volatility analysis measures the degree of price fluctuations, aiding users in assessing the risk levels associated with specific stocks. The correlation matrix identifies relationships between stocks, helping users uncover diversification opportunities or market trends. A positive correlation between two stocks indicates that their prices move in tandem, while a negative correlation suggests the opposite [7,8].

D. Risk-Return Trade-Off

The risk-return trade-off analysis evaluates annualized returns against volatility, providing a comparative view of stock performance. Scatter plots are used to visualize these relationships, making the risk-reward dynamics easily interpretable for decision-making [7,8].

E. Comparative Analysis

Comparative analysis normalizes stock prices to a base value, allowing users to compare cumulative returns across multiple stocks. This feature helps in identifying out-performers and under-performers within a given time frame [7,8].

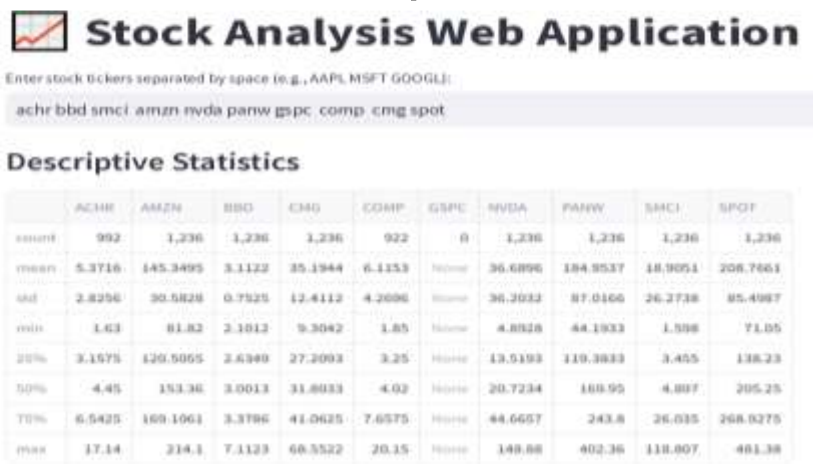
IV. RESULTS AND DISCUSSION

The application was tested on a sample data-set comprising 10 companies identified by the ticker symbols shown in the brackets. These companies are Archer Aviation Inc.(ACHR), Banco Bradesco S.A.(BBD), Super Micro Computer, Inc.(SMCI), Amazon.com, Inc.(AMZN), NVIDIA Corporation (NVDA), Palo Alto Networks, Inc. (PANW), S&P 500 Index (GSPC), NASDAQ Composite Index (COMP), Chipotle Mexican Grill, Inc. (CMG), and Spotify Technology S.A. (SPOT).

1. **Descriptive Statistics:** Descriptive statistics provided the key insights into the performance of the companies, as shown in the table below.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

Table 1: Descriptive Statistics



Stock Analysis Web Application

Enter stock tickers separated by space (e.g., AAPL MSFT GOOGL):

achr bbd smci amzn nvda panw gspc comp cmg spot

Descriptive Statistics

	ACHR	AMZN	BBD	CMG	COMP	GSPC	NVDA	PANW	SMCI	SPOT
count	992	1,236	1,236	1,236	922	0	1,236	1,236	1,236	1,236
mean	5.3716	145.3495	3.1122	35.1944	6.1153	None	36.6896	184.9537	18.9051	208.7661
std	2.8256	30.5828	0.7525	12.4112	4.2006	None	36.2032	87.0166	26.2738	85.4987
min	1.63	81.82	2.1012	9.3042	1.85	None	4.8928	44.1933	1.598	71.05
25%	3.1575	120.5065	2.6349	27.2093	3.25	None	13.5193	110.3833	3.455	138.23
50%	4.45	153.36	3.0013	31.8033	4.02	None	20.7234	168.95	4.807	205.25
75%	6.5425	169.1061	3.3796	41.0625	7.6575	None	44.6657	243.8	26.035	268.0275
max	17.14	214.1	7.1123	68.5522	20.15	None	148.88	402.36	118.807	481.38

From the results, it can inferred that stocks like NVDA and AMZN exhibited high mean prices, indicative of their status as large-cap companies attracting long-term investors. Conversely, stocks such as COMP, ACHR and SMCI demonstrated significant standard deviations, appealing to risk-tolerant traders due to their higher volatility and is depicted the figure1 below.

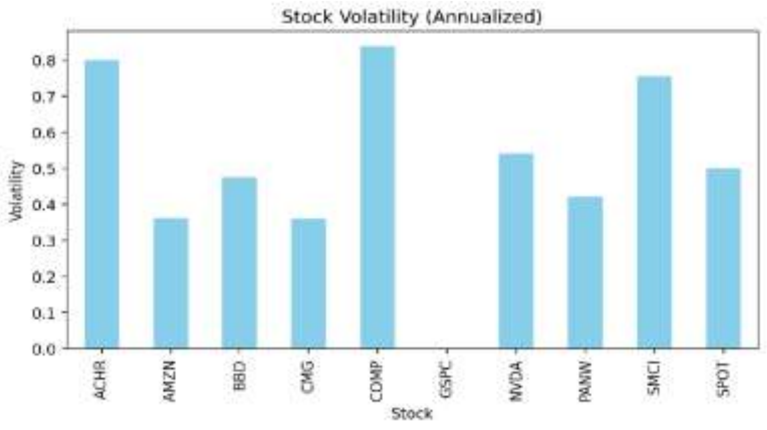


Figure 1: Bar Graph - Stocks Volatility Analysis

2. **Time Series Analysis:** The time-series analysis highlighted distinct trends and is shown in the figure 2 below. From the graph it can be inferred that, NVDA and PANW showed consistent upward trajectories, reflecting strong market confidence. In contrast, indices like GSPC exhibited stability with narrower price ranges, making them suitable for risk-averse investors.

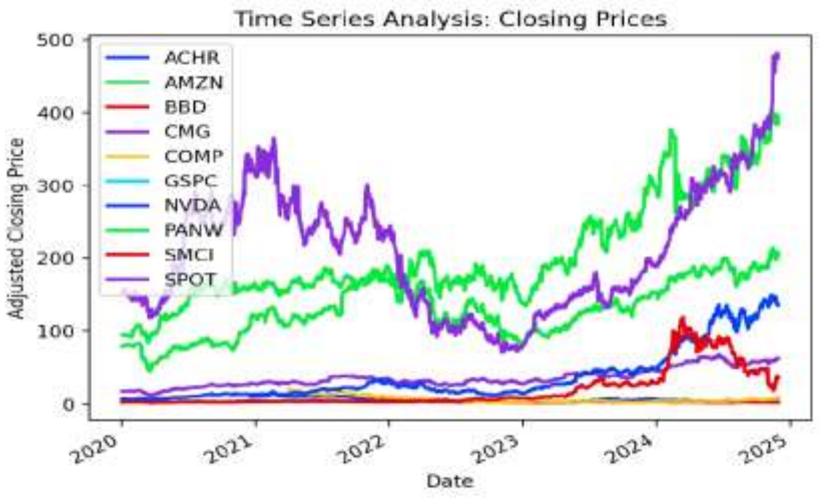


Figure 2: Time Series Plot of Adjusted Closing Price

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

3. **Correlation Matrix:** Figure 3 presents the details of the correlation matrix of stocks. The correlation analysis revealed that technology stocks like NVDA and AMZN had strong positive correlations, while diversification opportunities were evident between sectors such as technology and consumer goods.

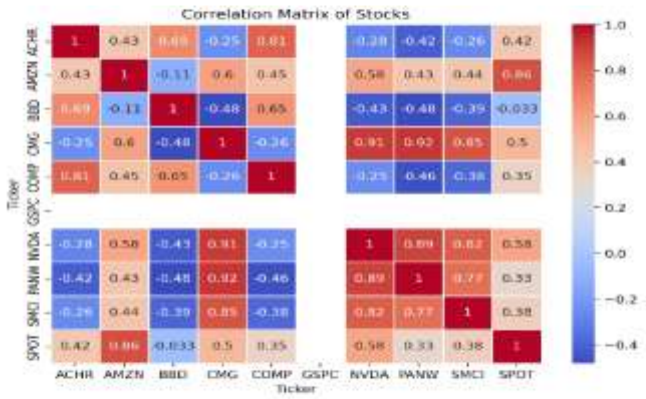


Figure 3: Correlation Matrix of Stocks

4. **Risk-Return Trade Off:** The risk-return trade-off analysis provides actionable insights for the stock analysis and is depicted in figure 4.

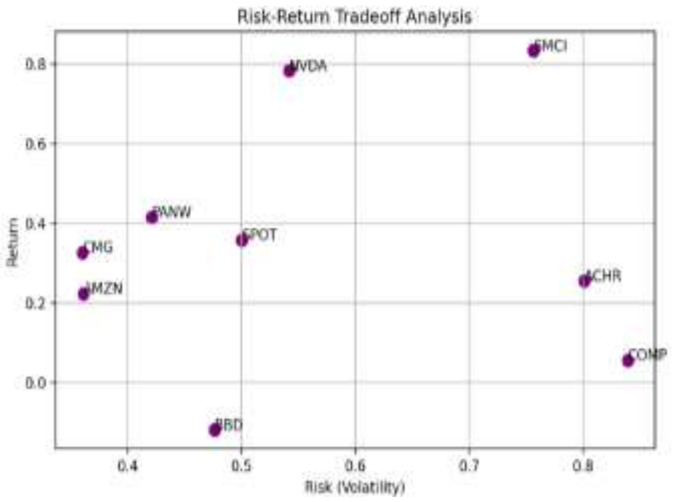


Figure 4: Return Vs Risk Plot

NVDA offers superior annualized returns, despite its high volatility, positioning it as a high-risk, high-reward investment. Indices like GSPC, with lower volatility and modest returns, were identified as safer options.

5. **Comparative Analysis:** Comparative Analysis further highlighted that AMZN consistently outperformed other stocks in cumulative returns, while ACHR showed significant price fluctuations within a short period. The results are shown in figure 5.

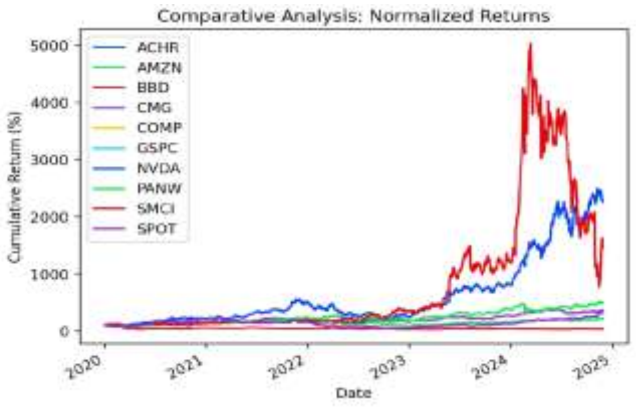


Figure 5: Normalized Cumulative Returns of Stocks for the Period 2020 to 2024

User feedback emphasized the application's effectiveness in simplifying complex analyses. Users appreciated the intuitive interface and real-time visualizations, which made it easier to interpret data and make informed investment decisions. The application's modular design also received praise for its scalability and potential for future enhancements.

V. CONCLUSION

The Stock Analysis Web Application bridges the gap between raw financial data and actionable insights by combining advanced analytical techniques with user-friendly design. Its modular and extensible architecture ensures adaptability to evolving user needs. Plans to incorporate more data sources and support for advanced visualisation techniques are underway

ACKNOWLEDGEMENTS

The authors express gratitude to the Department of Information Science and Engineering, BIET, Davangere, for their invaluable support and resources during the project's development.

VI. REFERENCES

[1] Streamlit Official Documentation. Streamlit: The fastest way to build and share data apps. Retrieved from https://docs.streamlit.io/.

[2] yfinance GitHub Repository. yfinance: Yahoo! Finance market data downloader. Retrieved from https://github.com/ranaroussi/yfinance.

[3] Pandas Official Documentation. pandas: Powerful Python data analysis toolkit. Retrieved from https://pandas.pydata.org/.

[4] NumPy Official Documentation. NumPy: The fundamental package for scientific computing in Python. Retrieved from https://numpy.org/.

[5] Matplotlib Official Documentation. Matplotlib: Python plotting. Retrieved from https://matplotlib.org/.

[6] Seaborn Official Documentation. Seaborn: Statistical data visualisation. Retrieved from https://seaborn.pydata.org/.

[7] Investopedia. Statistical Analysis in Finance. Retrieved from https://www.investopedia.com/.

[8] Financial Modeling Prep. Time Series Analysis and Stock Performance. Retrieved from https://www.financialmodelingprep.com/.

[9] Yahoo Finance API Documentation. Yahoo Finance: Historical and real-time stock market data. Retrieved from https://finance.yahoo.com/.

[10] Bloomberg Markets. Bloomberg: Business and Market News. Retrieved from https://www.bloomberg.com/markets.