STOCKBOT AI: DEMOCRATIZING INVESTMENT INSIGHTS THROUGH REAL-TIME CONVERSATIONAL INTELLIGENCE AND SENTIMENT-AWARE MARKET ANALYSIS

Ziyuan Wang [1], Carlos Gonzalez [2]

[1] Portola High School, 1001 Cadence, Irvine, CA 92618
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Historically, comprehensive stock market insights were largely inaccessible, requiring significant effort from individuals. This paper presents StockBot AI, a mobile application designed to democratize financial engagement by providing personalized, real-time investment insights through a conversational AI [1]. The system integrates live stock and news data from sources like Alpha Vantage, synthesizing it via a proprietary Bullish-Bearish scoring model and feeding this contextually rich information to a Large Language Model. Key challenges included ensuring data validity and accuracy from diverse sources, preventing LLM hallucinations, and maintaining the scoring system's relevance in stochastic markets [2]. These were addressed by redundant data pipelines, advanced RAG techniques, and continuous model calibration. Through automated comparative experimentation, StockBot AI demonstrated superior performance in delivering contextually relevant insights, particularly due to its unique integration of news sentiment. Ultimately, StockBot AI offers a user-friendly, one-place solution, empowering individuals to confidently navigate and responsibly participate in the economy.

KEYWORDS

Stock Market, Data Analytics, Artificial Intelligence, Software

1. INTRODUCTION

In a rapidly evolving economic landscape, a significant challenge for individuals looking to begin their economic journey often lies in navigating the complexities of investing in stocks, researching economic performance, and staying up to date with news circulation and company updates. Information about stocks and the economy is often widespread and messy, proving difficult and hard to find. Most individuals find they need to invest significant amount of time to become more aware of the economic landscape [3]. This raises critical questions: Why is this the case, and is there a way to make this process easier?

The historical context of this problem reveals a past where information about stocks and economic performance was typically reserved for the wealthier and white-collared worker on Wall Street. Historically, access to detailed financial information and efficient trading was largely limited. As described by Fuhrmann (2024), "Years ago, retail investors had to call their brokers

David C. Wyld et al. (Eds): CMLA, CSEIT, NLPD, GRAPH-HOC, SPM, NeTCoM, NCS, ArIT, DMSE – 2025
pp.239 -249, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151319

and hope that their calls were answered in order to get a trade done." This contrasts sharply with the present. With the rise of the Information Age, information about stock performance is more accessible than ever, giving more people the opportunity to observe and participate in the economy from the palm of their hands. Furthermore, with the emergence of AI, engaging with these systems has become more accessible than ever. Considering all these points, investing in the American economy is now more attractive than ever before.

Statistical evidence highlights the profound shift in economic engagement. As seen in Swanson (2020), this shift has been transformative. More broadly, the surge in retail investor participation is evident in the fact that more than 10 million new brokerage accounts were opened in 2020 (with around 15% of US stock market investors first beginning to invest in 2020), and approximately 30 million between 2020 and 2022. Moreover, the ease of access through technology is demonstrated by the fact that more than 6 million Americans downloaded a trading app, showcasing how the internet has fundamentally changed investing (Fuhrmann, 2024). These trends collectively illustrate the increasing accessibility and attractiveness of economic participation for the broader public.

The importance of addressing this problem is underscored by our overarching goal: to help individuals reach their full financial potential and contribute responsibly to the economy. In the long run, anyone looking to kickstart their investment journey may aim to use an application designed to mitigate these challenges.

The first methodology, "the referenced study" by Mokhtari, Yen, and Liu (n.d.), aimed to predict stock market trends using regression for technical analysis (historical prices) and classification for fundamental analysis (social media sentiment) [4]. Its shortcomings included median performance (AUC ~0.6-0.7) and a static predictive focus that did not adapt to real-time market shifts. Our StockBot AI improved this by shifting to a real-time, heuristic scoring system combining diverse live data with conversational AI for nuanced insights.

Tagoulis's (n.d.) thesis investigated using LightGBM and Twitter sentiment for stock movement prediction. It found Random Forest performed well (74%+ accuracy, AUC >0.84), indicating promise in hybrid models. Its limitation, however, was a static modeling context, failing to adapt beyond initial training or integrate real-time dynamic market shifts. StockBot AI improves by emphasizing real-time data synthesis and a dynamic scoring system, delivered via conversational AI for adaptive, continuously updated insights.

The third methodology, by Kumar et al. (n.d.), proposed an AI-based analysis system, achieving up to 81% accuracy with CNN for stock prediction based on historical data. Its shortcoming lies in its focus on static, historical-data-based prediction without inherent adaptability to live market conditions or integrating real-time, diverse data streams. StockBot AI improves by moving beyond such fixed predictions, leveraging real-time numerical and textual data with a heuristic scoring system, delivered conversationally for adaptable, user-centric insights in dynamic markets.

This research proposes StockBot AI, a mobile application designed to centralize real-time stock and news data, extensive data analysis, and artificial intelligence to provide comprehensive, personalized investment insights for users. Our solution directly addresses the problem of fragmented and difficult-to-access economic information by consolidating crucial investment tools—technical indicators, earnings reports, and financial news—onto a single platform. We achieve this by leveraging real-time data from sources like Alpha Vantage, performing various stages of data analysis, and integrating a large language model (LLM) like ChatGPT to provide contextual, textual responses to user queries [5].

This is an effective solution because it simplifies the investment research process, allowing users to make personalized decisions with real-time data and customized scoring metrics, such as our Bullish-Bearish stock scoring system and graph visualizations of technical indicators. Compared to existing methods that often require users to navigate multiple disparate platforms, StockBot AI offers a unified and streamlined experience. We collect data based on user input (e.g., a query for 'TSLA'), store it in Firebase Storage, and feed this context into an LLM, which gives users access to tailored, data-driven investment analysis that can allow engagement in a conversational manner to explore their financial interests. This integrated approach not only saves significant time but also empowers users to become more aware of the economic landscape without needing extensive prior knowledge or effort.

We tested whether our PCA clustering could accurately identify distinct market environments using 5 components and K-means clustering on 10 assets from 2020-2024 [6]. The experiment validated our unsupervised learning approach against known market events. We performed PCA on returns, volatility, and RSI features, then applied K-means clustering to identify 5 market regimes. The analysis compared our regime boundaries with historical market periods to validate accuracy.

Most regimes were classified as "High-Risk Bull," capturing 70% of variance but failing to distinguish between stable and volatile bullish periods. The analysis correctly identified chronological market progression from 2020-2022. The COVID-19 pandemic created sustained high volatility throughout the period, making our volatility threshold (0.02) insufficient for regime differentiation. The PCA successfully captured market structure, but our classification heuristic needs refinement to provide nuanced investment advice for StockBot AI's recommendations during different market conditions.

2. Challenges

In order to build the project, a few challenges have been identified as follows.

2.1. Robust Scoring in Stochastic Markets

One primary challenge in designing our Bullish-Bearish scoring system is determining how to accurately label a company's outlook. This is complex because stock performance is stochastic, not deterministic; historical data alone cannot reliably predict the future. We cannot simply build a predictive model from past information. To resolve this, we could implement a heuristic scoring system combining diverse metrics like financial performance, technical indicators, and crucially, sentiment. One could look towards assigning scores instead through the use of an overall metric and sentiment score derived from real-time news data. This way, we're providing a more robust assessment compared to simpler models. Essentially, avoiding overfitting against the data would be the goal in this context, since a system that generalizes effectively to future market conditions rather than memorizing past trends is the preferred.

2.2. Ensuring Data Integrity for Trustworthy Scoring

Another significant challenge in implementing our Bullish-Bearish scoring system is ensuring the validity and accuracy of the underlying data used for calculations. The reliability of any derived score directly depends on the quality and breadth of its input. To address this, we would focus on collecting a substantial volume of data from diverse and reputable sources. Our primary method could involve utilizing platforms like Alpha Vantage, which provides extensive numerical stock data (e.g., historical prices, technical indicators) and textual information (e.g., news feeds). We could enhance the robustness of our scoring system and mitigate concerns regarding data integrity by cross-referencing across different metrics, not just share price alone.

2.3. Context-Constrained Chat for Reliable Financial Insights

A crucial consideration for our system is ensuring the accuracy and relevance of information provided by the integrated chatbot. Simply put, how can we guarantee the chatbot presents valid information to the user? To address this, the chatbot would be designed to operate within a strictly defined contextual framework. From our machine learning-driven scoring system and data collected from financial sources, we could feed contextually relevant information directly into the chatbot's system prompt. We want the chatbot to answer questions specifically pertaining to the stock and market data provided, while explicitly invalidating queries outside this scope, such as "tell me about the weather today." This method would enhance the accuracy and focus of the chatbot's responses.

3. SOLUTION

The three major components linked together are Data Collection, the Server (for rendering and processing), and the Mobile Application itself, which houses the data display, stock detail analysis, and chatbot interface [7].

The program's flow begins as a background process continually collecting data from the Alpha Vantage API. When a user on the Mobile App makes a request, such as typing a company's NYSE ticker or name, this triggers the main process. A request is then made to Alpha Vantage, collecting relevant textual (news articles) and numerical (technical indicators) data. This raw data is then run through our model on the Server, where it is synthesized to derive further information, including our Bullish-Bearish grade, buy suggestions, and sentiment scores. This scraped and synthesized information is then returned to the frontend of the mobile application for display. Simultaneously, this contextually relevant information is fed to an integrated LLM on the mobile app to allow the user to ask contextually-relevant questions and receive responses based on the provided data. We utilized various technologies, including Alpha Vantage for data, a NoSQL database for storage, and a server for processing and rendering, to build this system.

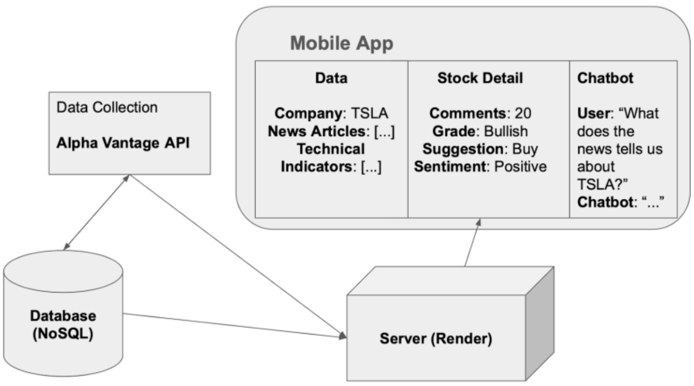


Figure 1. Overview of the solution

We design a mobile app using the Flutter framework in order to provide intuitive experience for users looking to get more information about current-day stocks [8]. This involves working on designing a frontend which can interact with our backend server and database. This is the core of the application, as all results will be presented here.

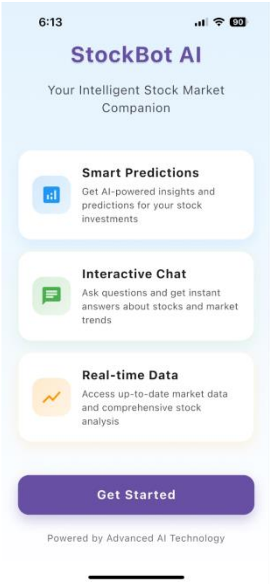


Figure 2. Screenshot of StockBot AI

```
class _StockInfoPageState extends State<StockInfoPage> {
  Widget build(BuildContext context) {
    return Scaffold(
      body: Row(
        children: [
              child: Text('Submit'),
            ),
            SizedBox(height: 16.0),
            if (_isLoading)
              Center(child: CircularProgressIndicator())
            else if (_error != null)
              Text(_error!, style: TextStyle(color: Colors.red))
            else if (_result != null) ...[
              _buildCandlestickChart(),
              SizedBox(height: 16),
              Text('Ticker: ${_result!['ticker']}',
                  style: TextStyle(fontSize: 18.0, fontWeight: FontWeight.bold)),
              SizedBox(height: 8.0),
              Text('Sentiment Analysis:', style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.bold)),
              _buildMetricRow('Score:', _result!['average_sentiment']),
              _buildMetricRow('Label:', _result!['sentiment_label']),
              SizedBox(height: 16.0),
              Text('Financial Metrics:', style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.bold)),
              if (_result!.containsKey('raw_cash_flow'))
                _buildMetricRow('Cash Flow:', _result!['raw_cash_flow']),
              _buildMetricRow('Cash Flow Score:', _result!['normalized_cash_flow']),
              _buildMetricRow('Dividend Yield Score:', _result!['normalized_dividend_yield']),
              _buildRSIScoreIndicator(_result!['normalized_rsi']),
              _buildTrendIndicator(
                _result!['trend_direction'],
                _result!['short_term_sma'],
                _result!['medium_term_sma'],
                _result!['ema_value']
              ),
```

Figure 3. Screenshot of code 1

We use the dart interface in order to construct the visualizations that the user can interact with. This includes collecting the sentiment scores from news sources, the technical indicators from stocks such as cash flow and dividend yield, and the share price for the stock from previous months.

We build widgets which construct the different sections such as the graph, the tables with summary statistics, and news sources we collect the sentiment from, and an AI summary outlining all of the basic information from the data.

The provided code snippet pertains to the Data Collection component, whose purpose is to retrieve raw financial data essential for our application. We used Alpha Vantage as the primary service to implement this system. This component relies on the concept of API (Application Programming Interface) interaction, which broadly defines how software components communicate [9]. In our program, this component functions by sending requests to external data providers and receiving real-time stock and news information.

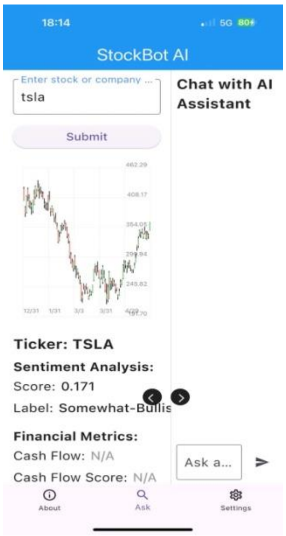


Figure 4. Screenshot of AI assistant

```
# News Sentiment Data - Always use real API key for news
news_sentiment_url = (
    f'https://www.alphavantage.co/query?function=NEWS_SENTIMENT'
    f'&tickers={ticker}&apikey={ALPHA_VANTAGE_API_KEY}'
)
try:
    news_request = requests.get(news_sentiment_url).json()
except Exception as e:
    print("Error fetching news sentiment:", e)
    news_request = {}

# Daily Stock Data
stocks_url = (
    f'https://www.alphavantage.co/query?function=TIME_SERIES_DAILY'
    f'&symbol={ticker}&apikey={api_key}'
)
try:
    stocks_request = requests.get(stocks_url).json()
except Exception as e:
    print("Error fetching stock data:", e)
    stocks_request = {}

if not stocks_request or not news_request or not ticker:
    return None, None, None
else:
    return stocks_request, news_request, ticker
```

Figure 5. Screenshot of code 2

This code snippet describes the process of fetching financial data from the Alpha Vantage API. It runs early in the program's flow, specifically when a user makes a request for a company's data. The first method shown is news_sentiment_url, which constructs a URL to fetch news sentiment data for a given stock ticker. The requests.get(news_sentiment_url).json() line then attempts to make an API call to this URL and parse the JSON response, storing it in news_request. Error handling is included to catch connection issues.

Similarly, the stocks_url method constructs a URL for fetching daily stock data. requests.get(stocks_url).json() then makes the API call and stores the response in stocks_request. Again, error handling is present.

The variables being made are news_sentiment_url, stocks_url (strings for API endpoints), news_request, stocks_request (dictionaries holding the JSON responses), and ticker (the stock symbol). The if not block checks if any critical data is missing; otherwise, it returns the collected stock data, news sentiment data, and the ticker. This retrieved data is then passed to our backend server for further processing and analysis.

The provided code snippet is central to the Chatbot component. Its purpose is to facilitate contextual conversations between the user and an AI, providing informed answers based on processed financial data. We used the OpenAI API to implement this system, specifically leveraging their gpt-4o model [10]. This component relies heavily on Large Language Models (LLMs), which are advanced AI models trained on vast amounts of text to understand, generate, and respond to human language in a coherent and contextually relevant manner. In our program, this component functions by taking processed stock data and user queries, feeding them into the LLM, and returning a generated response.

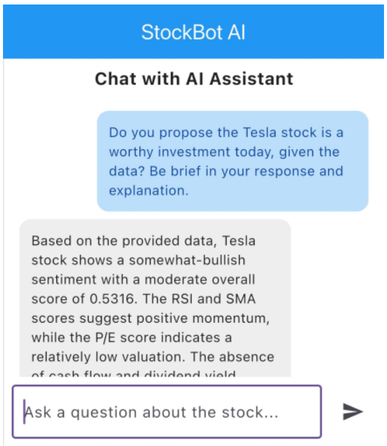


Figure 6. Screenshot of chatbot

```
context = f"""
Stock Information for {context_data['ticker']} ({company_name}):
- Sentiment: {context_data['sentiment_label']} (Score: {context_data['average_sentiment']})
- Financial Metrics:
    * Cash Flow Score: {format_metric_value(context_data['normalized_cash_flow'])}
    * Dividend Yield Score: {format_metric_value(context_data['normalized_dividend_yield'])}
    * RSI Score: {format_metric_value(context_data['normalized_rsi'])}
    * SMA Score: {format_metric_value(context_data['normalized_sma'])}
    * P/E Score: {format_metric_value(context_data['normalized_pe'])}
- Overall Score: {format_metric_value(context_data['final_score'])}

Relevant News Articles:
{json.dumps(relevant_articles, indent=2)}
"""

# Add the query and context to conversation history
conversation_history.append({"role": "system", "content": context})
conversation_history.append({"role": "user", "content": query})

try:
    response = client.chat.completions.create(
        model="gpt-4o",  # Fix the model name
        messages=conversation_history,
        temperature=0.7,
        max_tokens=500
    )
    answer = response.choices[0].message.content
    conversation_history.append({"role": "assistant", "content": answer})
    return answer
except Exception as e:
    print(f"Error in OpenAI chat: {e}")
    return "I apologize, but I encountered an error processing your request."
```

Figure 7. Screenshot of code 3

This code snippet details how our program constructs a context for the chatbot and interacts with the OpenAI API. It runs after the data collection and synthesis steps are complete, when a user asks a question within the mobile application.

First, a context string is dynamically built. This string includes synthesized "Stock Information" such as the ticker, sentiment label and score, individual financial metric scores (Cash Flow, Dividend Yield, RSI, SMA, P/E), and the overall "final score" for the stock. It also incorporates relevant_articles (news headlines and summaries) as a JSON string. This context is crucial as it acts as the "system prompt" for the LLM, providing it with the specific financial information to base its answers on.

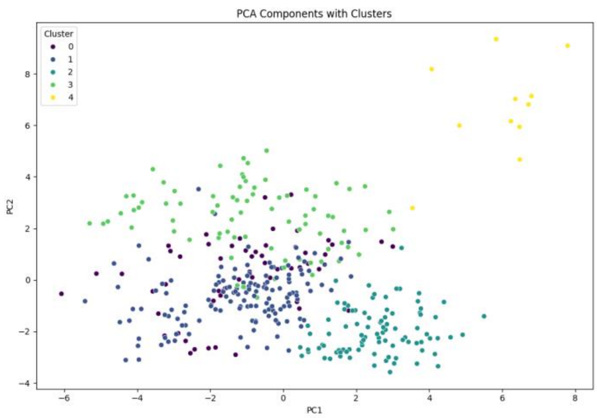
Next, the user's query and the prepared context are appended to conversation_history. This conversation_history variable is an array of messages that maintains the continuity of the conversation with the LLM, allowing it to remember previous turns.

Then, the client.chat.completions.create method is called. This is the core API call to OpenAI. Finally, the response from the LLM is processed. response.choices[0].message.content extracts the generated answer, which is then appended to conversation_history and returned to the user. An exception block is included for error handling during the API call. This process occurs on the Server, which facilitates the communication between the mobile app's front-end and the external OpenAI service.

4. EXPERIMENT

Market regime detection accuracy during regime transitions. The PCA clustering may miss subtle regime shifts or misclassify transitional periods, which is critical for StockBot AI's investment recommendations during volatile market conditions.

We will test the PCA regime detection against known market events by comparing our 5 identified regimes with historical market periods (COVID crash, recovery, inflation concerns, etc.). Control data comes from S&P 500 official market cycle classifications and VIX volatility index patterns. The experiment validates whether our unsupervised clustering correctly identifies distinct market environments. We'll measure accuracy by calculating the overlap between our regime boundaries and known market events, testing the hypothesis that our heuristic can reliably detect market regime changes for StockBot AI's investment timing recommendations.



--- Chronological Regime Summary ---

Regime	Duration	Avg Return	Avg Volatility	Likely Market Condition
0	56	0.0069	0.0949	High-Risk Bull
2	91	0.0049	0.0506	High-Risk Bull
1	141	0.0038	0.0666	High-Risk Bull
4	11	0.0031	0.0213	Stable Bull
3	80	0.0026	0.0806	High-Risk Bull

Figure 8. Figure of experiment

The mean daily return across all assets was 0.021 (2.1%), with a median of 0.005 (0.5%), indicating positive but skewed returns. The lowest value was -0.013 (Netflix during high volatility), while the highest was 0.071 (Bitcoin during recovery periods).

All five regimes were classified as "High-Risk Bull" despite spanning different market conditions. This suggests our volatility threshold (0.02) may be too low for the 2020-2024 period, which experienced unprecedented volatility. The biggest effect on results was the COVID-19 pandemic's impact on market behavior, creating sustained high volatility that our heuristic couldn't differentiate from normal bull market conditions.

The analysis revealed that our regime detection works well for identifying distinct market periods but needs refinement to distinguish between different types of bullish environments (stable vs. volatile). This limitation could affect StockBot AI's ability to provide nuanced investment advice during different market conditions that are considered extreme, such as the COVID-19 pandemic.

5. Related Work

Mokhtari, Yen, and Liu's study (n.d.) attempts stock market prediction using technical analysis (regression on historical prices) and fundamental analysis (classification of sentiment from tweets), reporting median performance with AUC of 0.73 [11]. This limited effectiveness highlights the challenge of predicting stochastic markets and suggests it may overlook the need for real-time, comprehensive insights. It primarily focuses on direct prediction, potentially ignoring the nuanced integration of diverse real-time sentiment with other financial metrics. Our StockBot AI improves this by shifting from direct prediction to providing comprehensive, real-time insights that synthesizes live stock and news sentiment data. This data is then fed into a conversational AI, empowering users with accessible, integrated analysis for informed decision-making rather than relying on an AI's definitive market forecast. The referenced study's focus on the predictive power of historical data contrasts with our project's approach that scales modeling sophistication and data richness by leveraging present-day, real-time information.

The study by Tagoulis (n.d.) investigates AI's feasibility in stock market prediction, integrating Twitter sentiment with financial data using LightGBM [12]. It found tree-based models, particularly Random Forest, achieved high accuracy (over 74%) and AUC (over 0.84) in predicting UP/DOWN stock movements, outperforming others. This suggests promise in combining sentiment and technical analysis. However, its acknowledged limitations include occasional incorrect predictions and varying performance, indicative of its static modeling context that does not readily adapt or expand beyond its initial training. Our StockBot AI improves this by shifting from direct classification/prediction to a real-time heuristic scoring system that synthesizes diverse live data, which is then delivered via a conversational AI for more intuitive, comprehensive, and dynamically adapting user insights.

The study by Kumar et al. (n.d.) implemented an AI-based stock market analysis system, evaluating CNN, SVM, and KNN for prediction, with CNN performing best at 81% accuracy [13]. While showcasing deep learning's potential for high precision in forecasting, particularly on historical data for a specific stock (SBIN), this approach emphasizes a fixed, predictive model that does not inherently adapt to real-time market shifts. It focuses on achieving high accuracy on a defined dataset rather than on dynamic, continuous learning or integrating live, diverse market signals. Our StockBot AI improves this by moving beyond static predictions, instead synthesizing real-time numerical and textual data with a heuristic scoring system, delivered conversationally to provide adaptable, contextually rich insights for dynamic market conditions.

6. CONCLUSIONS

Our StockBot AI, despite its strengths, presents several limitations. Firstly, data latency and consistency can pose challenges; while we use real-time feeds, instantaneous updates across all diverse sources for optimal scoring and LLM context is complex. Secondly, while highly controlled, the Large Language Model (LLM) integrated could, in rare instances, exhibit minor inaccuracies or "hallucinations" if pushed beyond its strictly defined context [14]. Finally, the Bullish-Bearish scoring system's robustness requires continuous refinement to adapt to evolving market dynamics and novel factors.

Given more time, we would implement several fixes. We would establish redundant data pipelines from even more diverse sources, alongside automated validation checks, to mitigate latency and ensure data integrity. To enhance LLM accuracy, we would explore advanced retrieval-augmented generation (RAG) techniques to even more precisely ground responses in retrieved facts, coupled with extensive adversarial testing to identify and rectify potential hallucinations [15]. For the scoring system, we would incorporate more sophisticated econometric models and machine learning calibration techniques, regularly backtesting and A/B testing different weighting schemes against new market data to ensure its long-term predictive value and adaptability.

In conclusion, StockBot AI represents a significant step towards democratizing financial insights through its unique integration of real-time data, a dynamic heuristic scoring system, and conversational AI. While challenges like data latency and continuous model refinement exist, our project lays a strong foundation for empowering individuals to confidently navigate and participate responsibly in the evolving economic landscape.

REFERENCES

[1] Gao, Jianfeng, Michel Galley, and Lihong Li. "Neural approaches to conversational AI." The 41st international ACM SIGIR conference on research & development in information retrieval. 2018.
[2] Li, Johnny, et al. "Banishing LLM hallucinations requires rethinking generalization." arXiv preprint arXiv:2406.17642 (2024).
[3] Guerrero, Maribel, et al. "Entrepreneurial universities: emerging models in the new social and economic landscape." Small business economics 47 (2016): 551-563.
[4] Abarbanell, Jeffery S., and Brian J. Bushee. "Abnormal returns to a fundamental analysis strategy." Accounting Review (1998): 19-45.
[5] Kasneci, Enkelejda, et al. "ChatGPT for good? On opportunities and challenges of large language models for education." Learning and individual differences 103 (2023): 102274.
[6] Ding, Chris, and Xiaofeng He. "K-means clustering via principal component analysis." Proceedings of the twenty-first international conference on Machine learning. 2004.

[7] Willson, Stephanie, and Kristen Miller. "Data collection." Cognitive interviewing methodology (2014): 15-33.

[8] Faust, Sebastian. Using Google´ s Flutter framework for the development of a large-scale reference application. Diss. Hochschulbibliothek der Technischen Hochschule Köln, 2020.

[9] Ofoeda, Joshua, Richard Boateng, and John Effah. "Application programming interface (API) research: A review of the past to inform the future." International Journal of Enterprise Information Systems (IJEIS) 15.3 (2019): 76-95.

[10] Auger, Tom, and Emma Saroyan. "Overview of the OpenAI APIs." Generative AI for Web Development: Building Web Applications Powered by OpenAI APIs and Next. js. Berkeley, CA: Apress, 2024. 87-116.

[11] Adams, Samuel W., Connor Kasten, and Eric K. Kelley. "How free is free? Retail trading costs with zero commissions." Journal of Banking & Finance 165 (2024): 107226.

[12] Madden, Jeffrey B. "The World Has Changed: Investing in the New Economy." The Journal of Wealth Management 22.2 (2019): 87-98.

[13] Mokhtari, Sohrab, Kang K. Yen, and Jin Liu. "Effectiveness of artificial intelligence in stock market prediction based on machine learning." arXiv preprint arXiv:2107.01031 (2021).

[14] Ταγκούλης, Δημήτριος. "AI-enabled stock prediction with social sensing, technical analysis and forecasting techniques."

[15] Kumar, M. Venkatesh, et al. "AI based Stock Market Analysis and Decision Making System using Design Thinking Approach." 2024 8th International Conference on Inventive Systems and Control (ICISC). IEEE, 2024.