# LLM Powered Financial Advisor Chatbot

*Charishma Puli*
*MPS Data Science*
*University of Maryland Baltimore County*
*Baltimore, USA*
*cpuli1@umbc.edu*

**Abstract - This study investigates whether stock prices show seasonal trends and volatility over time. For investors and financial experts, stock analysis is an essential tool for navigating the complex dynamics of the stock market. Its goal is to compare different time series analysis models to identify patterns. The goal of this financial advising's project is integration of time series models and large language models to make investors' lives easier. In addition to more conventional deep learning model like LSTM, this application utilizes time series models like Prophet, Dart, and Garch. OpenAI is integrated to address financial queries. The paper outlines the development of a chatbot application that uses Amazon Web Services (AWS). Time series forecasting and OpenAI integration are made possible by Amazon Lambda, while Amazon Lex provides chatbot interface. The model's construction uses S3, EC2 instances, and Amazon SageMaker. The project's product is a chatbot that can efficiently respond to questions about stocks and finances.**

**Keywords - Financial Chatbot, Stock Analysis, Financial Advising, Large Language Models (LLMs), Amazon Web Services (AWS), Time series forecasting**

## I. Introduction

Forecasting of stock is a major part of stock analysis, which provides the performance analysis, volatility and potential future trends of individual stocks. Many studies in various fields, including operations research, computer science, statistics, economics, and finance, have been motivated by the stock price prediction. The stock forecast helps buyers and investors take the least risk and get the most profits with well-informed decisions. Financial advising is essential for investors to understand the full picture of stock investing and finances [1]. Financial keywords, budgets, money, capitals, and other associated information are all considered financial inquiries.

The project workflow compasses two main stages. It involves comparing different models to determine the most effective approach for handling the stock inputs and outputs. To identify the optimal model for stock forecasting, the project includes comparing various time series forecasting strategies to handle stationarity, seasonality, and trends with the deep learning model of a recurrent neural network (RNN), which is used for processing sequential data. Then, the project focuses on building a chatbot application focused on user interactions. The application performs tasks according to the instructions from the user. This application serves user to financially plan the future effectively.

The paper's structure is divided into seven sections, along with acknowledgement 2 covers the history of the models and related research. section 3 outlines the project's methodologies and application flow. Section 4 presents the study's results and analysis. Section 5 highlights key findings and summarizes conclusions. Section 6 offers a final set of possible improvements for the future.

## II. Background and Related Work

### *A. AI powered Financial Chatbot*

A financial chatbot is an interactive conversational application for user engagement. Chatbots in finances especially in stock investing in viable for financial needs, which increases the business value of the model. Chatbots are used as an alternative to human customer service [2]. Large Language Models (LLMs) are incorporated into chat applications by AI-powered chatbots, which are the current era of bots. Microsoft Copilot, one of the chatbots used for business chat, is one example of how many real-time organizations constructed these chatbots [3]. The other is that Bloomberg has an AI research team that integrates AI with financial analysis and does ongoing AI [4].

### *B. Time Series Forecasting*

Time series forecasting is a powerful analytical technique used to predict future values based on historical data observations ordered chronologically. It involves analyzing the time series dataset's patterns, trends, and seasonality to make accurate predictions. The modelling in this paper includes statistical models like Autoregressive Integrated Moving Average (ARIMA), deep learning algorithms like Long Short-Term Memory (LSTM), and advanced forecasting methods like Prophet and Generalized Autoregressive Conditional Heteroskedasticity (GARCH). The Dart Python library is also employed with its libraries ARIMA and Prophet.

### *C. Large Language Models*

Large Language Models (LLMs) are artificial intelligence models that use transformers and are trained on a vast corpus of unlabeled datasets to perform NLP tasks such as question-answering text production, and summarization. One decoder-based transformer model developed by OpenAI, an AI research and deployment business, is the Chat Generative Pre-trained Transformer (ChatGPT), which revolutionizes natural

language processing jobs [5]. To execute its prompts and build its custom model programs, OpenAI offers an API. In this project, a model is created to respond to financial questions using an OpenAI model called openai using OPENAI_API_KEY [6].

## III. Methodology

### A. Data

Yahoo Finance (yfinance) is a popular Python library developed by Ran Aroussi to scrap the data available on Yahoo Finance. The yfinance API provides financial information about stocks, including historical market data, financial statements, company news, etc. It interests developers and analysts as it's an open-source library that is easy to access using API calls [7]. In a project, the closing price is used to forecast the stock. Using the closing price data from stock data allows one to assess a stock's performance over a specific period. The closing price is the last price at which a stock trades during the trading session, and it is considered an important indicator of a stock's value and market sentiment.

### B. Data Preprocessing and Exploratory Data Analysis

#### i. Augmented Dickey fuller test

Augmented Dickey-Fuller test checks whether the time series is stationary or not. Stationarity is crucial for forecasting models. The existence of a unit root in a first-order AR model is the null hypothesis of the DF test. When DF fails to anticipate a higher-order autocorrelation in the model. To capture the higher-order autocorrelation, Dicky and Fuller augment the equation with higher-order lags. This technique is known in the time series literature as the Augmented Dickey Fuller (ADF) test [8]. The ADF statistic of -2.058 and the high p-value of 0.261 indicate non-stationarity. Further support comes from the plotting in Fig. 1, where the critical values for 1%, 5%, and 10% do not align with the stock data, confirming its non-stationary nature.

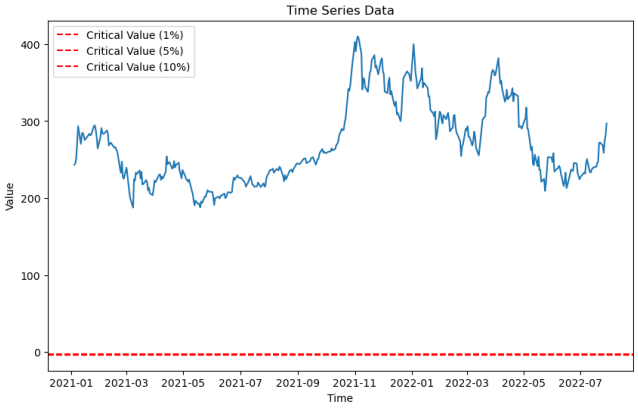


Fig. 1. ADF test results of stock data

#### ii. Stock Returns

Analysis the plot of stock data visually depects for any Seasonality, trend and noise in the data. That is handeled by the particular models. We can calculate stock returns from pct_change() function of data frames, which is "Fractional change between the current and a prior element, useful for comparing the fraction of change in a time series of elements" [9]. Which is useful for detecting the seasonality, trends and volatility. This approach proves valuable in the Fig, 2 ADF test, where the returns data demonstrates stationarity.

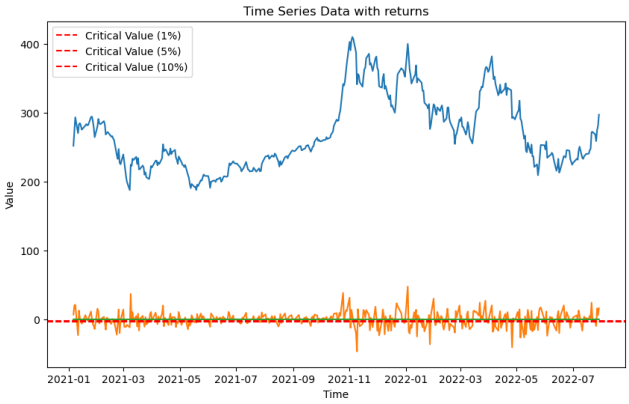


Fig. 2. ADF test results of stock data with returns

#### iii. Differencing

In time series analysis, the technique known as "differencing is used to stabilize a mean and variance of the time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality by deducting successive observations" [10]. First-order differencing is subtracting the current value from its prior value making the data stationary. In this project, first differencing is used to calculate the change of the stocks. After conducting the ADF test for the first difference data plotted in Fig .3, the results indicate an ADF Statistic of -6.107 and a p-value of approximately 0.000000095. These values reject the null hypothesis, proving that the time series is stationary with the first differencing.

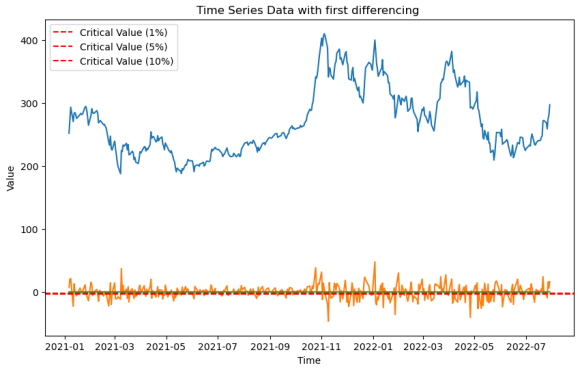


Fig. 3. ADF test results of stock with first order differencing

#### iv. Quantile-Quantile plot

The QQ plot, or Quantile-Quantile plot, is a scatter plot used to assess whether a dataset follows a particular probability distribution such as a normal or exponential. A scatterplot made by plotting two sets of quantiles is known as a QQ plot [11]. The data is in the same probability distribution if the data points form a roughly straight line. Deviation from the diagonal straight line says the data points are not in the specified probability distribution. In Fig 4, the Q-Q plot is from the plotting of stock returns to exhibit a normal distribution. It is visible that data points are in a nearly straight diagonal line, proving the data is distributed normally.

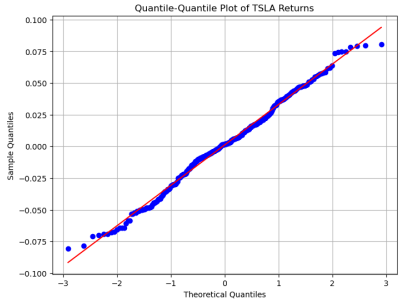


Fig. 4. Quantile-Quantile plot

### v. Autocorrelation function and Partial autocorrelation function

The autocorrelation function (ACF) plot and partial autocorrelation function (PACF) plot help identify the auto-regressive (AR) parameter (p) and Moving Average (MA) parameter (q). Autocorrelation implies serial dependence. It happens specifically when there is a strong correlation between a time series and its lag variant. The ACF plot explains the correlation between a time series and its lagged version. The PACF plot explains the additional correlation between times series and successive lagged terms. The ACF plot and PACF plot illustrate that the lags and a series have greater dependence the longer the bar in the ACF plot. In Fig 5 and Fig 6, The ACF plot using plot_acf and PACF plot using plot_pacf both exhibit a spike at the lags 7th and 9th, we also have larger peaks. The 9th lag is chosen for the model because larger lags are used for better seasonality checks.

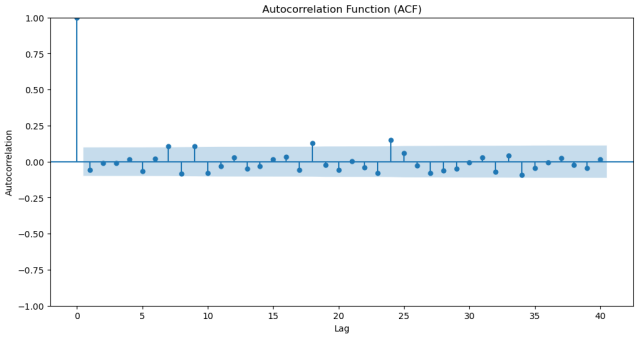


Fig. 5. ACF of stock data

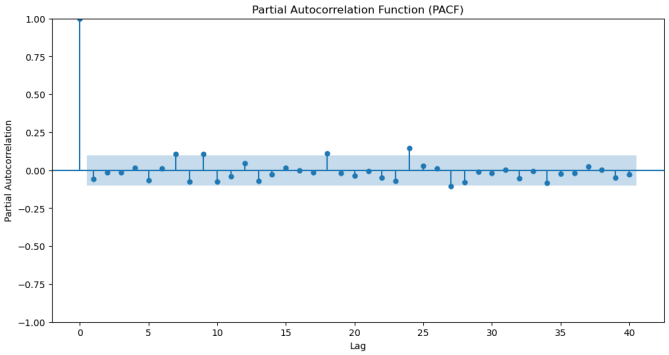


Fig. 6. PACF of stock data

## B. Data Modelling

### i. Long Short-Term Memory (LSTM)

LSTM is a gradient-based method to address the vanishing error. This model comes under the umbrella of deep learning algorithms, under RNN, a neural network with recurrent connections [12]. LSTM is used to process the sequential data. LSTM models can capture long-term dependencies and comples patterns. This model is trained using the stock returns and performing the MinMaxScaler standardization technique, to normalize the time series data before training. The scaled data is used for the LSTM modelling and resulted the plot Fig. 7 showing the actual stock prices and forecasted the stock prices are in upward increasing order with the starting prediction been way less than previous stock price.

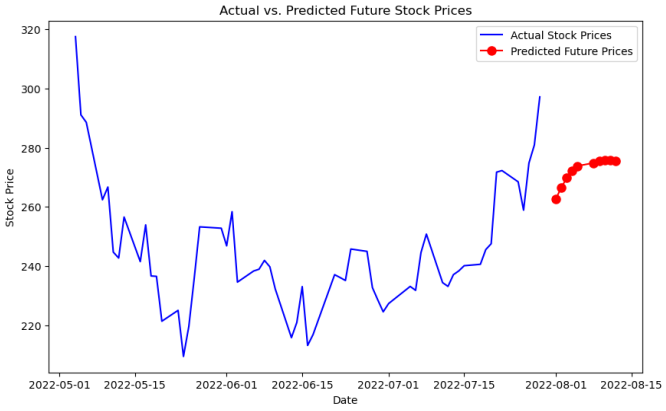


Fig. 7. LSTM forecasting stock prices

### ii. Autoregressive Integrated Moving Average (ARIMA)

The ARIMA statistical analysis model uses time series data to find future trends. A model is considered auto regressive if it predicts future values and uses lagged moving averages [13]. This model predicts future values with the past values chronologically. ARIMA consists of auto regressive (AR), integrated (I) and moving average (MA) parameters. Auto regressive is a variable that changes on its own using the lagged or prior value. Integrated(I) is for the differencing term

for the stationarity. Moving average(MA) uses lagged observations, the relationship between an observation and a residual error from a moving average model. In this project, we have effectively trained the ARIMA model using the returns of the closing price with p =7 and q = 7 for predictions over long horizons. The autoregressive term in our model depicts the linear relationship between the current observation and its preceding 7 observations, while the Moving Average term captures the errors from the previous 7 observations. The volatility capturing of the forecasted stock prices, as shown in Fig. 8, demonstrates ARIMA model in predicting stock prices with volatility.

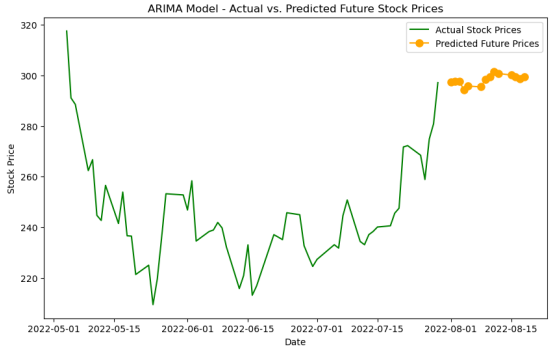


Fig. 8. ARIMA model stock forecast

### *iii. Prophet*

Prophet is the time series forecasting model developed by Facebook, used for handling time series data with strong seasonal patterns. Prophet is based on an additive model with daily, weekly, and annual seasonality with holiday effects that fits non-linear tendencies [14]. Prophet is trained using the first differences of the stock price with the parameters changepoint_prior_scale=0.1, holidays_prior_scale=0.01, seasonality_prior_scale=0.01. The dataframe with columns ds, yhat, yhat_lower, yhat_upper are forecasted. The yhat will be our forecasted values. Fig 9 shows the forecasting range if the prophet models, the shaded blue part shows the forecasting interval range between yhat_upper and yhat_lower and forecasted stock price yhat is colored blue line, which is in decreasing order.

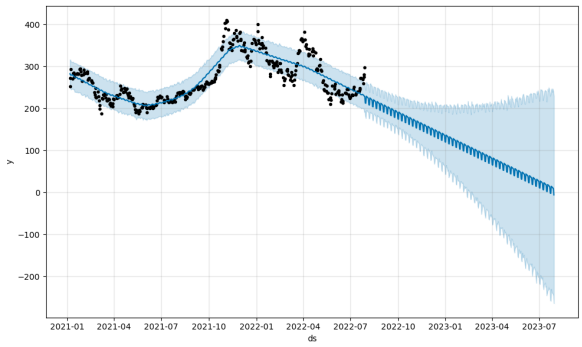


Fig. 9. Prophet forecasting stock prices

### *iv. Darts*

Darts is a Python library for user-friendly time series forecasting and anomaly detection. It offers various forecasting models, from traditional statistical methods like ARIMA to state-of-the-art forecasting models like Prophet. Dart provides similar functionalities like scikit learn [15]. It makes model easy to train, validate, and make predictions with various time series models, including handling multivariate data. ARIMA model in Dart library is trained with stock returns and Fig. 7 shows the decreasing order forecasted values of the stocks. Prophet model in Dart library is also trained using the stock returns and observed that the forecasted values are almost the same of the actual stock values.


Fig. 10. forecasting stock prices using Dart ARIMA model.

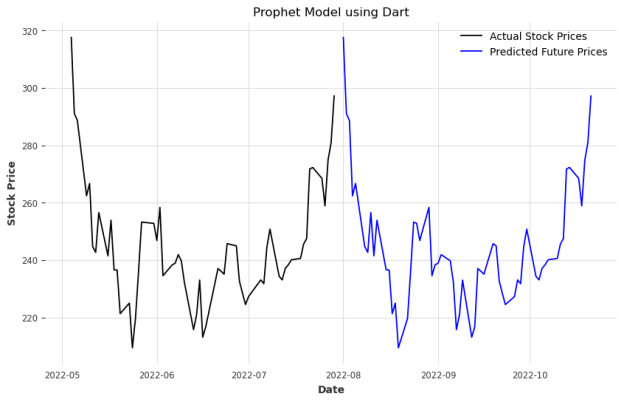


Fig.11. forecasting stock prices using Dart Prophet model.

### *v. Generalized Autoregressive Conditional Heteroskedasticity (Garch)*

GARCH is a time series forecasting models which is volatility dependent. This model uses for past conditional variances in the current conditional variance equation. Stationarity and autocorrelation conditions handled using this volatility model. GARCH data modeling is trained using stock returns with the parameters p = 5 and q =7, from the autocorrelation factors. This model captures the long term dependencies while taking care of volatility. In Fig. 12, rolling forecast of time series data is compare with the stock returns. The 14 days horizon forecast is shown in Fig 13, splitly

decreasing at the starting and incrased and consistant at the end.

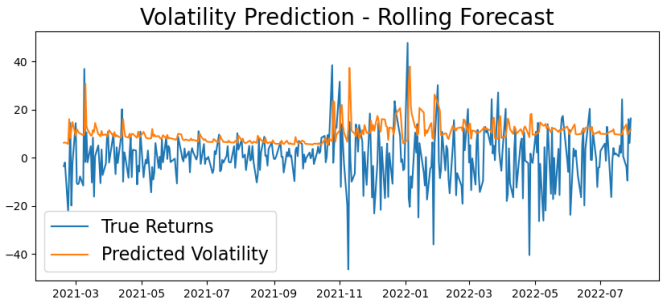


Fig. 12. Rolling forecasting stock prices using GARCH model.

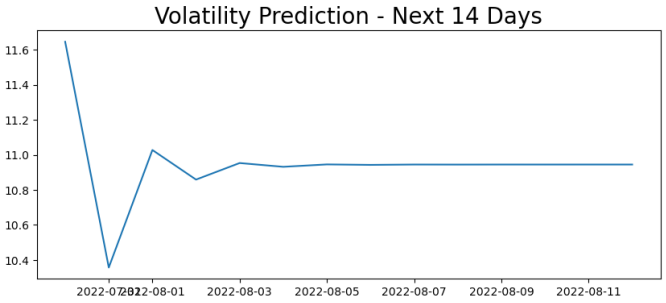


Fig. 13. forecasting stock prices using GARCH model.

### *vi. Prophet and Garch Combined time series forecasting model*

To get the better model, the idea is to use the prophet model to capture the underlying trends and the garch model to capture the conditional variability of the predicted trends. The residual from the prophet model is used to train the garch model and find the volatility structure. In Fig. 14, the residual graph is shown from prophet model trained on the stock returns. The residual graph shows all the percentage changes of the stock price. Volatility predicted using garch model is shown with the Fig. 15, comparing to the Fig 13, the volatility is captured well with prophet model and garch model combining.

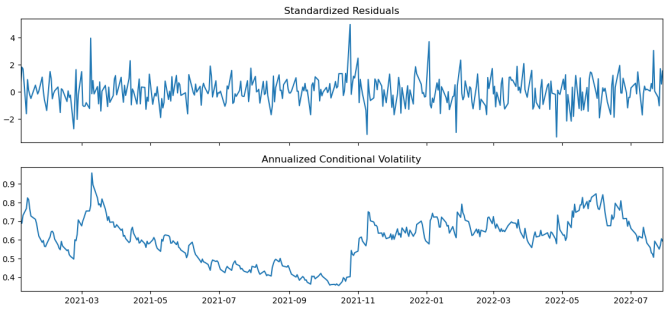


Fig. 14. Residuals from Prophet model.

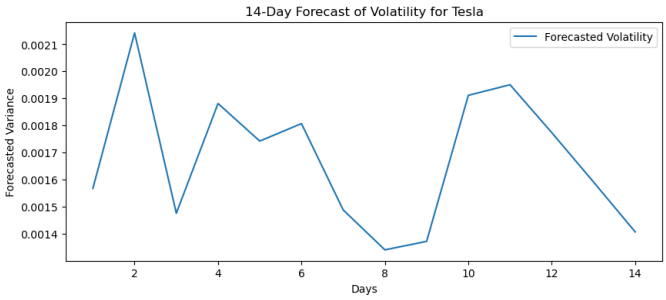


Fig. 15. Volatility from GARCH model.

### *vii. Model Evaluation and Selection*

The plots of forecasted closing prices in Fig. 16, say that the LSTM model starts aligning with the sample testing stock prices. Still, the volatility is not defined, with the arima mode start well and exhibited differet volatility. The prophet showed the graph with most volatility differences and is in the increasing order of the model. As the garch model can be used for the volatility check and first underestimates the data and then starting to get the higher values for the predictions. The combined prophet and garch model is selected for the chat bot application as it starts with slightly correct predictions and ends with almost the same predictions. It shows the stock decreasing trend.

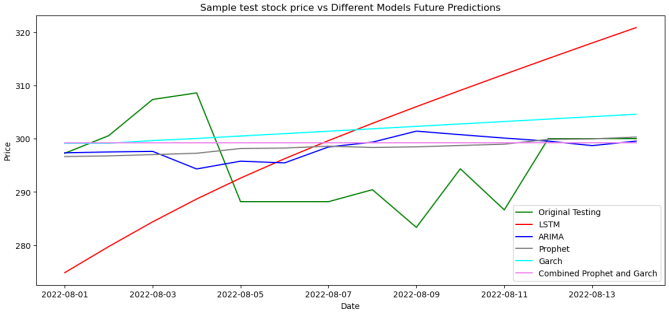


Fig. 16. Model Comparison using forecasted prices.

## *C. Application Flow*

The chatbot application in Fig. 17 is based on the flow of the user, providing any options for the chatbot, such as stock investment and financial questions. Stock investment query uses the function with a time series forecasting model and OpenAI with Langhain to extract the stock names. Financial Questions triggers the function with Openai and gets the financial questions answered. The chatbot application is built using Amazon Web Services (AWS). AWS is a cloud computing platform from Amazon. The first step is creating an AWS account and starting the Amazon Lex chatbot application. The workflow of the chatbot application is shown in Fig 19.

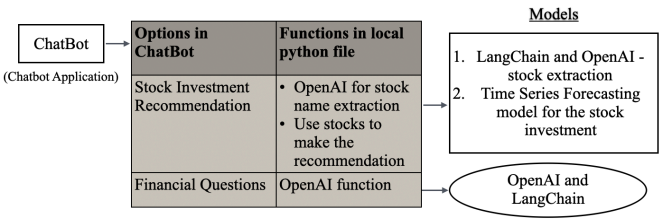


Fig. 17. Model Comparison using forecasted prices

### *i. Chatbot - Amazon Lex*

Amazon Lex is a component of Amazon Web Services (AWS) that builds conversational interfaces called chatbots. Lex consists of intents. The initial intent shown in Fig. 18 with the 'hi' statement utters the card group with two options: Financial queries and stock investment. The next is the financial query intent, which connects to the lambda functions of open. The other intent connects to the lambda function of stock recommendation with openai init and it connected to the amazon sagemaker model.

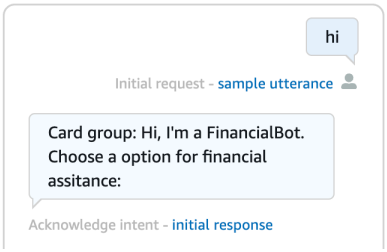


Fig. 18. Initial intent of chatbot

### *ii. Financial Queries Option - Amazon Lambda*

Amazon Lambda is a developer service that runs the code serverless. First, the Python language lambda function is created, and with Amazon lex by checking the "cloud watch logs" API call format and response to the lambda function output using the format of the API call. The libraries to run the lambda function are downloaded with the docker container to get all the Python dependencies of the code. The financial queries are answered with the Openai and Langchain layers downloaded by docker, added to the lambda function, and connected with the event API of Amazon Lex.

### *iii. Stock Investment Option - Amazon Lambda and Amazon Sagemaker*

Amazon Sagemaker is AWS's machine learning modeling service. In this service, we can train the model using notebook instance and store the model using endpoints, which endpoints are called using rest API and are used from the lambda function. The critical step here is to give the IAM role administration access, as this will make the developer eligible to make the models and deploy them in the s3 bucket. Another critical step is to start the EC2 instance before starting the notebook instance. In the project, the combined prophet and dart model is trained by stock data, and the model is stored in the Sagemaker models, deploying the model in the s3 bucket. Use rest API calls to the endpoints of the model and use that endpoint to connect with the Amazon lambda function. For stock investment options, the Sagemaker lambda function is connected to the Openai lambda function to retrieve the stock names and make predictions on the stock names to give the investment advice generated by the model.

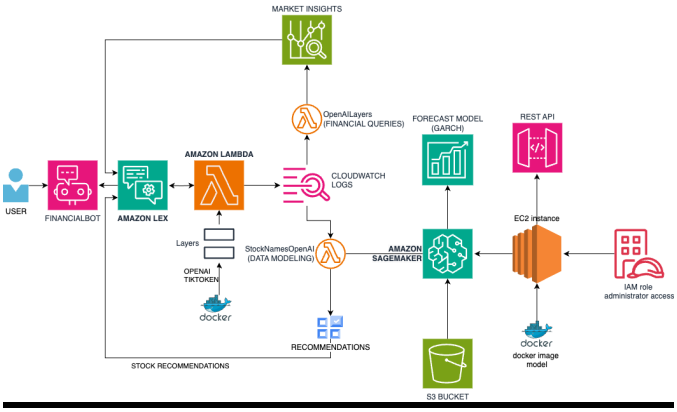


Fig. 19. Initial intent of chatbot

## IV. Results and Discussions

The project results in a user-interactable AWS chatbot. The final chatbot contains two options to choose from: stock investments and financial queries. Stock Investments provides stock investment advice using the LLM and time series forecasting model with Prophet and Garch. The combined model is selected to capture the trend and the volatility of the model. Financial queries provide financial information using the OpenAI and Langchain models. The resulting chatbot is shown in Fig. 19.

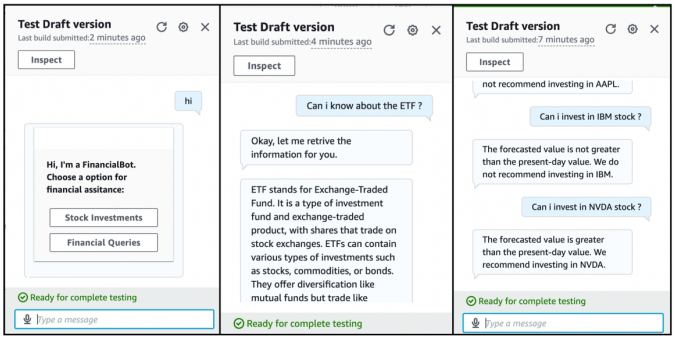


Fig. 19. Chatbot Application

## V. Conclusion

In conclusion, the project provides a well-performed chatbot utilizing the LLM and the time series analysis. The time series data is analyzed for stationarity using the Augmented Dickey-Fuller Test. Normalizing using the Differencing and stock returns, the Autocorrelation factor for the seasonality and trend was examined using ACF and PACF plots. Then, the LSTM, ARIMA, Prophet, and Garch models are compared. The dart library is explored with the arima and prophet models. The

project gives the idea of time series analysis on stock data using yfinance. The chatbot application will be helpful in user engagement and simplify life for buyers and investors.

## VI. Future Enhancements

The project's future enhancements involve exploring different forecasting models to do a time series on stock prices and integrating other external factors for the sudden change in the stock trend. The chatbot application will upgrade to integration with a frontend.

## Acknowledgement

I express my gratitude to Dr. Tony Diana for his outstanding direction and consistent support during this study project. His constant encouragement and insightful remarks have been crucial in helping us shape our concepts and improve the quality of this work. I greatly appreciate his leadership and the time he contributed to helping me.

## References

[1] D. Shah, H. Isah, and F. Zulkernine, "Stock Market Analysis: A Review and Taxonomy of Prediction Techniques," *International Journal of Financial Studies*, vol. 7, no. 2, p. 26, May 2019, doi: https://doi.org/10.3390/ijfs7020026.

[2] "Chatbots in consumer finance," *Consumer Financial Protection Bureau*, Jun. 06, 2023. Available: https://www.consumerfinance.gov/data-research/research-reports/chatbots-in-consumer-finance/chatbots-in-consumer-finance/

[3] J. Spataro, "Introducing Microsoft 365 Copilot – your copilot for work," *The Official Microsoft Blog*, Mar. 16, 2023. Available: https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/

[4] "Artificial Intelligence (AI): Latest Coverage From Bloomberg - Bloomberg," *Bloomberg.com*, Feb. 06, 2024. Available: https://www.bloomberg.com/ai

[5] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[6] OpenAI, "OpenAI API," *platform.openai.com*. Available: https://platform.openai.com/docs/api-reference/introduction

[7] R. Aroussi, "yfinance: Yahoo! Finance market data downloader," *PyPI*, 2023. Available: https://pypi.org/project/yfinance/

[8] "Dickey-Fuller Test - an overview | ScienceDirect Topics," *www.sciencedirect.com*. Available: https://www.sciencedirect.com/topics/economics-econometrics-and-finance/dickey-fuller-test

[9] "pandas.Series.pct_change — pandas 2.2.2 documentation," *pandas.pydata.org*. Available: https://pandas.pydata.org/docs/reference/api/pandas.Series.pct_change.html. [Accessed: May 04, 2024]

[10] "Stationarity and differencing of time series data," *Duke.edu*, 2019. Available: https://people.duke.edu/~rnau/411diff.htm

[11] "Q-Q plots," *onlinestatbook.com*. Available: https://onlinestatbook.com/2/advanced_graphs/q-q_plots.html

[12] R. Staudemeyer and E. Morris, "-Understanding LSTM - a tutorial into Long Short-Term Memory Recurrent Neural Networks," 2019. Available: https://arxiv.org/pdf/1909.09586. [Accessed: May 04, 2024]

[13] A. HAYES, “Autoregressive Integrated Moving Average (ARIMA) Prediction Model,” *https://www.investopedia.com/*. Available: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp#:~:text=An%20autoregressive%20integrated%20moving%20average%2C%20or%20ARIMA%2C%20is%20a%20statistical,values%20based%20on%20past%20values.

[14] S. J. Taylor and B. Letham, “Forecasting at scale,” Sep. 2017, doi: https://doi.org/10.7287/peerj.preprints.3190v2. Available: https://peerj.com/preprints/3190/

[15] “Time Series Made Easy in Python — darts documentation,” *unit8co.github.io*. Available: https://unit8co.github.io/darts/

[16] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity,” *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, Apr. 1986, doi: https://doi.org/10.1016/0304-4076(86)90063-1