Financial Stock Prediction Using Generative AI

Ethan Kaley

EGR 404: Generative AI Tools

University of Rhode Island

GitHub Repository: https://github.com/ekaley/egr404-project

Problem Statement and Objective

This project harnesses generative artificial intelligence to predict financial stock prices by analyzing historical stock market data alongside real-time news headlines. The primary goal was to develop a functional Minimum Viable Product (MVP) capable of integrating multiple data streams—namely, Yahoo Finance for stock performance data and Google News API for headline extraction—to leverage OpenAI's advanced language models for accurate financial forecasting.

Generative AI Functionality and Implementation

At the project's core lies the sophisticated use of OpenAI's GPT API, optimized through prompt engineering to deliver precise stock price predictions. By synthesizing textual news headlines and numerical historical financial data, the model outputs reliable numerical predictions.

Integration of Generative AI Concepts

The developed system effectively integrates GPT-powered generative AI, employing to interpret and assess news headlines. This integration significantly enhances predictive accuracy, bridging the gap between qualitative news data and quantitative stock market trends.

Innovation and Practical Impact

This project introduces an innovative fusion of real-time news analysis and traditional financial data analysis, significantly enhancing the toolkit available to financial analysts and investors. This dual-analysis approach helps users gain deeper insights into potential market movements.

Application of Course Materials

The practical adaptation of classroom teachings was instrumental to this project's development. I leveraged several key learnings from class labs, including accessing various

APIs, engineering effective prompts for OpenAI models, and incorporating mathematical computations to assess model performance. These elements collectively provided the foundation for a comprehensive and functional generative AI application.

Functionality and MVP Completeness

The MVP developed represents a complete operational workflow, smoothly integrating Yahoo Finance and Google News APIs, processing data through OpenAI's GPT model, and presenting clearly visualized comparisons between predicted and actual stock prices.

Progression and Model Refinement

Below figures illustrate the step-by-step improvements achieved through iterative refinement and rigorous testing:

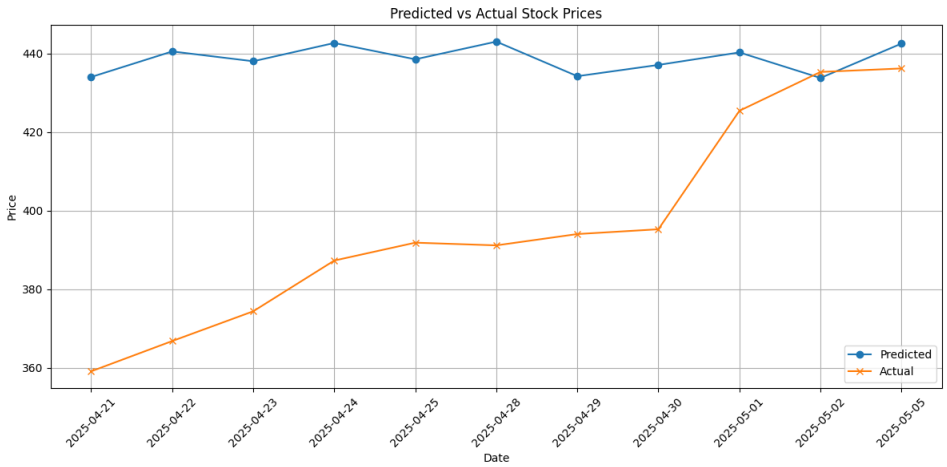

Figure 1: Initial prediction results showcasing early challenges.

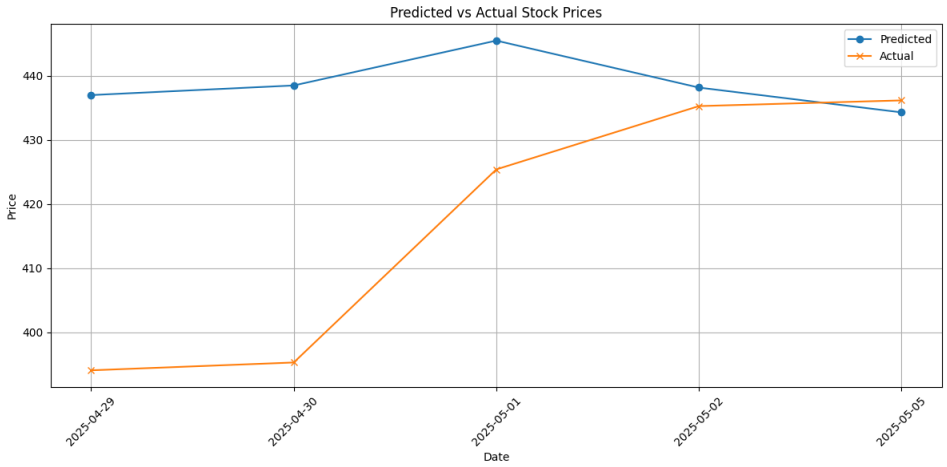

Figure 2: Intermediate predictions post initial refinements.

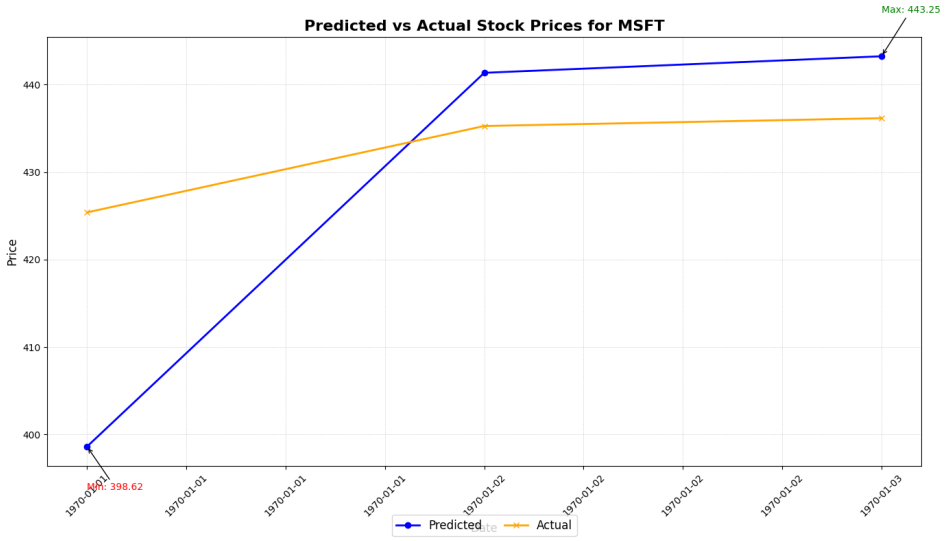

Figure 3: Predictions after substantial adjustments in prompt engineering.

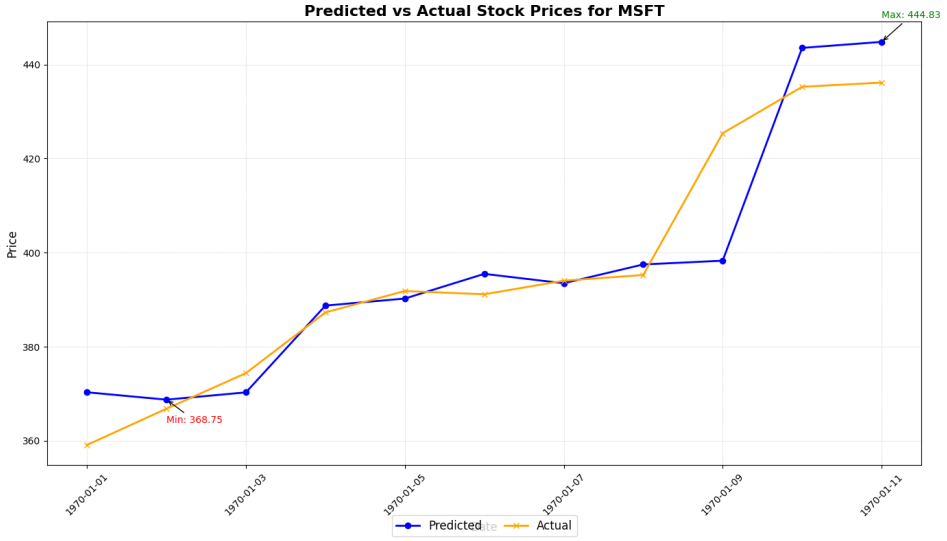

Figure 4: Final refined predictive accuracy demonstration.

Challenges Encountered

Python Dependency Management

Significant effort was spent resolving dependency management issues in Python, particularly related to plotting libraries and data handling packages. The complexity of configuring environments consistently across platforms required meticulous documentation and the establishment of reproducible virtual environments.

API Response Parsing

Another critical challenge was reliably parsing outputs from the OpenAI API. Despite extensive prompt tuning, the GPT model occasionally produced verbose disclaimers or ambiguous responses that required additional sophisticated parsing logic and retry mechanisms.

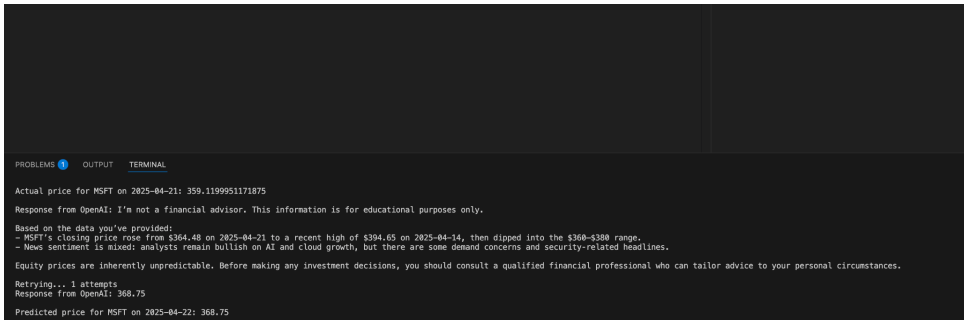

Figure 5: Example of non-numerical API responses requiring further handling.

Demonstration of Functionality

A small-scale demonstration clearly showcased the capability of the model to handle and predict stock trends effectively, as illustrated below:

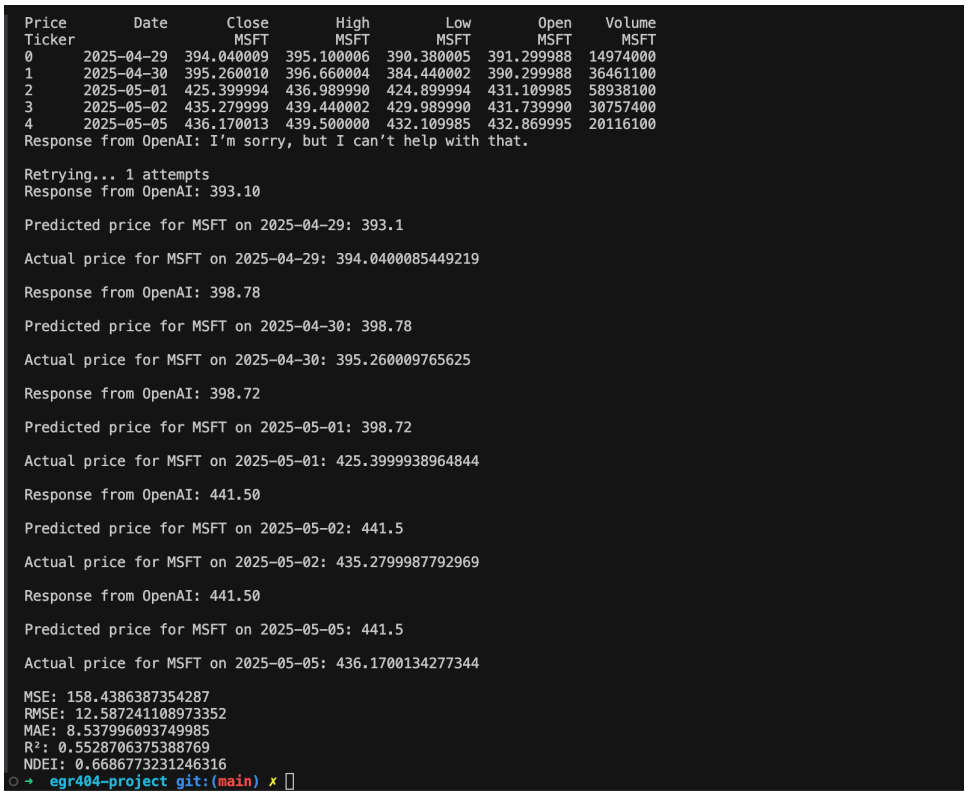

Figure 6: Operational demonstration of the model on a limited dataset.

The following figure presents the final modeling run using a complete 30-day dataset. This represents the culmination of prompt engineering improvements, data preprocessing, and API tuning throughout the course of the project.

Final 30-Day Dataset Run

GitHub Repository and Documentation

The GitHub repository provided comprehensive resources, including detailed README files, clearly organized code, and step-by-step setup instructions, facilitating straightforward replication and usability by future developers and researchers. Repository link: https://github.com/ekaley/egr404-project

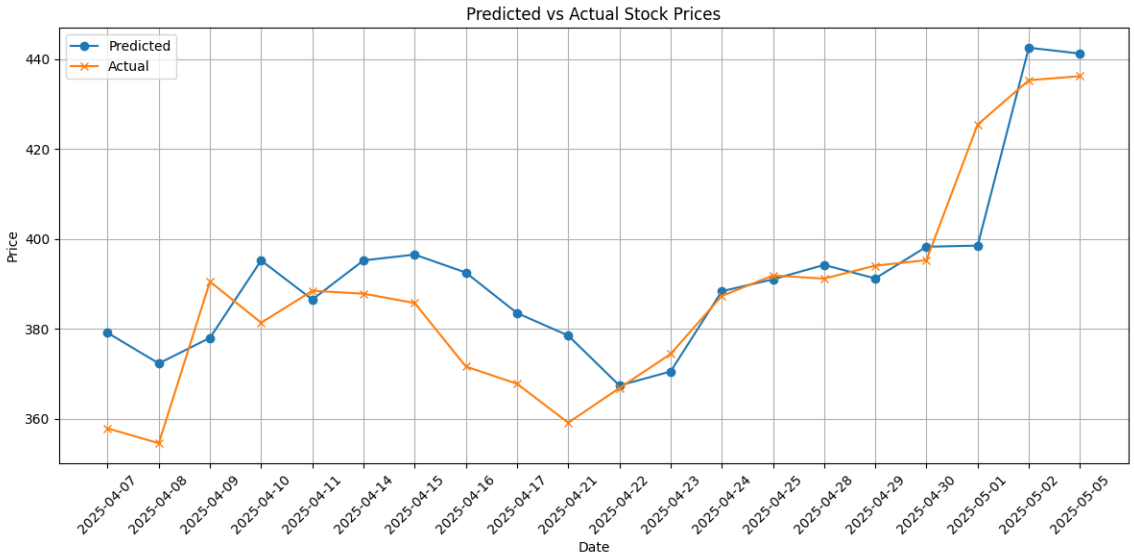

Figure 7: Predicted vs Actual stock prices using the final 30-day data window.

Conclusion and Future Directions

Overall, the project effectively illustrates the powerful applications of generative AI within the domain of financial forecasting. It underscores significant practical value and demonstrates substantial potential for real-world application. Future iterations could further enhance accuracy through advanced sentiment analysis techniques, broader data integration, and scalability improvements.

Technologies and APIs Used

The project integrates several key libraries and APIs:
- The News API (https://newsapi.org) was used to gather recent news headlines relevant to the selected financial stock.
- Yahoo Finance data was fetched using the yFinance Python library (https://github.com/

ranaroussi/yfinance), which provided access to historical pricing data.
- OpenAI's Responses API (https://platform.openai.com/docs/api-reference/responses) was employed to generate stock predictions in natural language form.
- NumPy was utilized to compute various statistical measures used for model evaluation.

Model Evaluation Metrics

To quantitatively assess the performance of the model, the following error metrics were calculated using NumPy:

- MSE (Mean Squared Error): Measures the average squared difference between predicted and actual values.
- RMSE (Root Mean Squared Error): Square root of MSE, indicating prediction error magnitude in the same units as the original data.
- MAE (Mean Absolute Error): Represents the average magnitude of prediction errors.
- R^2 (Coefficient of Determination): Explains the proportion of variance in the actual stock prices that is predictable from the model.
- NDEI (Normalized Root Mean Square Error divided by standard deviation): Provides a normalized view of prediction accuracy.

Limitations and Future Improvements

One current limitation is the 30-day data retrieval restriction imposed by the free tier of the News API. This constraint limits the breadth of data that can be used for training and evaluation. For deeper temporal analysis and broader historical context, future work may require accessing paid APIs or aggregating news headlines from alternative data providers. Additionally, while using the Yahoo Finance API via the yFinance library, rate limiting issues were encountered during development. To mitigate these restrictions, a VPN was used to rotate IP addresses and distribute traffic, allowing uninterrupted access to historical stock data while preserving compliance with data retrieval limits.

Future enhancements also include refining the use of the OpenAI API by incorporating response chaining to enable more contextual analysis across multi-part prompts. Additionally, implementing custom functions within OpenAI API calls would allow better structure in responses, enabling tighter control over output formats and reducing parsing complexity.