Predicting Next-Day Stock Closing Prices using Machine Learning and Sentiment Analysis

Sara Mezuri[1] **and Alireza Golkarieh**[2]

[1]*saramezuri@oakland.edu*
[2]*golkarieh@oakland.edu*

This manuscript was compiled on December 1, 2024

Abstract

This project aims to predict the next day's stock closing price using historical stock data and sentiment analysis of financial news and tweets. We collected sentiment data from Bloomberg Terminal and used it alongside stock data to build time series and machine learning models. The baseline model, SARIMA, with sentiment as an additional factor , did not improve the prediction accuracy. It only predicted that tomorrow's stock price would be the same as today's, showing no short-term predictive power. We found that adding sentiment score did not influence the model's performance. In contrast, other models like Random Forest and LSTM showed better results predicting the next day closing price, but still not were not improved by adding the sentiment scores. This project highlights the challenges of integrating sentiment analysis into stock price predictions and suggests that more complex models and a more thorough sentiment analysis may be needed for better accuracy.

1. INTRODUCTION

Stock price prediction is crucial for informed investment decisions and profitability. Traditional methods like SARIMAX struggle with short-term predictions, even when external factors like news sentiment are considered. Previous research shows sentiment analysis can offer insights, but models often fail to capture market complexity. This project explores Auto-ARIMA, SARIMAX, Random Forest, and LSTM models to assess how sentiment data influences stock predictions and make predictions for the next day's closing price.

2. MOTIVATION

With social media's rise, information about companies spreads faster than ever, including financial news, opinions, rumors, and tweets from influential figures like CEOs. These posts can significantly impact stock prices, as investors react to real-time information. For instance, a CEO's tweet about a product launch or financial performance can cause sharp stock price movements.

As a result, predicting stock prices has become more complex. Traditional models based on historical data may overlook the effects of real-time news and social media sentiment. By integrating sentiment from news and social media, we can better understand how news events influence stock prices, leading to more accurate predictions in today's fast-paced market.

3. APPROACH

3.1. Data Collection and Preprocessing

3.1.1. Data Sources

The stock data for the Nasdaq 100 (NDX) and S&P 500 indices were collected from Yahoo Finance using the yfinance library and TC 2000, covering daily prices (Open, Close, High, Low) from September 2024 to January 2015, excluding weekends.

Sentiment data, derived from news articles and Twitter posts, was collected from the Bloomberg Terminal due to challenges and limitations with APIs and web scraping. Bloomberg provided sentiment data from January 2015, which was already preprocessed by supervised machine learning models [1]. However, Bloomberg Terminal did not provide a sentiment count and score for the indices, thus we used the sentiment counts and scores from the 15 most influential stocks (by market cap) for each index. The raw sentiment counts were converted into scores, with data available for weekends as well.

3.1.2. Preprocessing Steps

To align stock data (weekdays only) with sentiment data (including weekends), sentiment scores were merged with stock prices, ensuring correct alignment for weekdays. Missing sentiment scores were imputed using the median for each column to avoid bias. (Some stocks had more missing data than others, which is expected since popular stocks like Amazon generate more news and tweets compared to less prominent ones like Texas Instruments.) Sentiment scores for news and Twitter were normalized to a common range (-1 to 1) for consistency. These preprocessing steps ensured the data was cleaned, aligned, and ready for analysis.

3.2. Model Selection and Training

3.2.1. Algorithms

For predicting the stock closing prices, several models were considered. We chose regression models designed for time series data since stock prices are numerical variables and follow a time series structure. Our focus was on models capable of handling time series regression while learning patterns in a sequential manner.

Auto-ARIMA is suitable for time series forecasting, automatically selecting the best parameters for non-stationary data with trends and seasonality.It helps automatically determine the best parameters *(p, d, q)* and *P, D, Q* based on the given time series data and a specified criterion *(e.g., AIC, BIC)*

SARIMAX was a good choice because it handles time series data effectively, capturing both trends and seasonality, which are common in stock prices. Additionally, it allows for the inclusion of exogenous variables, enabling us to incorporate sentiment scores to evaluate their impact on stock price predictions.

Random Forest is an effective model for stock price prediction due to its ability to capture nonlinear relationships in complex data, making it well-suited for the highly volatile and noisy nature of stock markets. By averaging multiple decision trees, it reduces the impact of noise, prevents overfitting, and provides more stable predictions. It can handle large datasets with many features, offers insights into feature importance, and adapts to both regression (price prediction) and classification tasks. Additionally, its ensemble approach improves generalization, and it can manage missing data with minimal preprocessing.

LSTM (Long Short-Term Memory) networks are well-suited for stock price prediction due to their ability to capture sequential and long-term dependencies, handling issues like vanishing gradients through their unique architecture. They excel in extracting nonlinear

patterns from complex financial data and are resistant to noise, focusing on underlying trends. LSTMs also process multi-feature inputs, integrating factors like trading volume and sentiment, and adapt well to non-stationary data, continuously learning over time. Additionally, they can predict both single-step and multi-step forecasts. These strengths enabled our LSTM model to outperform traditional statistical models like ARIMA in capturing stock market dynamics.

3.2.2. Training Process

Since we are dealing with time series model, random splits were unsuitable due to the importance of temporal order. Instead, we used a sliding window approach, splitting the data based on time: the first 80% of days were used for training, and the final 20% for testing. This approach ensures the model learns from past data to predict future outcomes.

The SARIMAX model was chosen using Auto-ARIMA, which selected the optimal parameters based on the AIC criterion.

For the Random Forest and LSTM models, k-fold cross-validation and hyperparameter tuning was used to validate the model on different subsets of the training data. This helps ensure the model generalizes well to unseen data.

4. EXPERIMENTAL EVALUATION

4.1. Methodology

Our initial goal for this project was to predict the next day's closing price and analyze how effectively sentiment data from news and social media improve stock price predictions for the Nasdaq 100 and S&P 500 indices. Our initial hypothesis was that the sentiment scores positively correlate with stock price movements and the models incorporating sentiment data perform better than those based solely on historical prices.

The independent variables are the historical stock prices (Close), and the sentiment scores (news and Twitter), while the dependent variable is the predicted stock closing prices. We evaluated four different models Auto-ARIMA, SARIMAX (with exogenous sentiment data), Random Forest, and LSTM, using Python libraries including **Scikit-learn, TensorFlow, and Statsmodels** (for the time series).

Then, we compare all our models without sentiment and with sentiment by comparing the accuracy we get from each of them and the other metrics.

Models were compared to determine the impact of including sentiment data, with results presented using graphs and metrics, and also to choose the best-performing model from all of them (the model that gave us the best next day close price prediction).

4.2. Evaluation Metrics

The time series models and the Random Forest model were evaluated using Mean Absolute Error (MAE), R^2 Score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to provide a comprehensive assessment of its performance. MAE measures the average magnitude of prediction errors, offering an intuitive sense of model accuracy. MSE penalizes larger errors more heavily, highlighting significant prediction inaccuracies. RMSE, by providing error measurements in the same units as the target variable, makes the model's performance easier to interpret for stock prices. The R^2 Score evaluates how well the model explains variance in the data, giving insight into the overall fit of the model. Together, these metrics ensure a balanced evaluation of accuracy, error magnitude, and model reliability.

Meanwhile MSE was chosen as the sole evaluation metric for the LSTM model because it quantifies the prediction error. Since stock prices are continuous and can have varying magnitudes, MSE is effective in assessing the model's overall accuracy, especially when capturing subtle changes in stock trends. Moreover, LSTMs are designed to capture long-term dependencies in time series data, and MSE helps in identifying how well the model predicts these temporal patterns without being influenced by less important evaluation metrics.

4.3. Results and Discussion

Results were visualized using line graphs for predicted vs. actual prices and error distributions. (See appendix for all the graphs and charts).

For the time series analysis, SARIMAX$(2,1,2)$x$(1,0,[1],5)$ was identified as the best model using the AIC criterion. This model effectively captured stock price movements and trends in the long-term but struggled with short-term predictions.

Adding sentiment scores to the model did not result in any changes to the selected parameters or improve the predictions. This indicates that the sentiment scores had no measurable impact on the model's performance, contradicting our hypothesis that sentiment data would enhance prediction accuracy.

The evaluation metrics for the SARIMAX model are as follows:

	Metric Name	Metric Value
Metrics	MAE	2714.69
	MSE	7,606,569.32
	RMSE	2758.00

These metrics suggest limited predictive power for short-term forecasting and demonstrate that the model failed to leverage the sentiment data effectively.

While Random Forest performed well, stock market prediction remains difficult due to market efficiency and randomness. For our specific problem, which involved time-sensitive predictions, the Long Short-Term Memory (LSTM) network proved more suitable. Interestingly, adding sentiment data to the Random Forest model actually worsened its performance, likely due to the challenge of incorporating sentiment data effectively.

For the Random Forest model applied to the Nasdaq 100, the performance metrics were:

	Metric Name	Metric Value
Metrics	MAE	782.66
	R^2	0.78
	MSE	1,755,111.75
	RMSE	1,324.81

For the Random Forest model applied to the S&P 500, the metrics were:

	Metric Name	Metric Value
Metrics	MAE	191.18
	R^2	0.67
	MSE	109,933.04
	RMSE	331.56

As the best model, was chosen to be the LSTM model with the following characteristics:

Architecture	Hyperparameters
Input layer dimension = 7	Sequence length = 30
Number of layers = 2	Split ratio = 0.8
Hidden dimension = 64	Number of epochs = 20
Output dimension = 1	Optimizer = Adam
	Learning rate = 0.001

The evaluation criteria (MSE) was as follows:

- Test loss for Nasdaq: 0.005
- Test loss for S&P 500: 0.0007

5. LIMITATIONS

There are several limitations in this project. First, assuming that sentiment directly affects stock prices might oversimplify how markets behave, as many other factors also influence stock prices. Also, using sentiment data from just the top 15 stocks to represent the broader indices like Nasdaq 100 and S&P 500 may not fully capture the overall market sentiment.In addition, the sentiment data we showcased a lot of noise.

Data accessibility was another challenge, and we didn't have weighted sentiment scores to reflect the importance of each news article or tweet.

For the Random Forest model, one major limitation was the high computational cost, especially with large datasets, which made training slow and resource-intensive. Additionally, Random Forest doesn't naturally account for the sequential nature of time-series data. To make it work for stock price predictions, extra feature engineering and time-series techniques were needed.

When using LSTM networks, we faced the challenge of selecting the right architecture. We tested different configurations, but some models only learned the average stock price, which didn't help in capturing meaningful patterns. Finding the right LSTM structure for accurate forecasting was difficult.

For the SARIMAX model, it struggled to capture the nonlinear relationships in the data, especially when sentiment was included. Despite adding sentiment as an external factor, the model didn't show any improvement. SARIMAX was also harder to tune, and sometimes it either overfitted or underfitted the data, making its performance less reliable.

6. CONCLUSIONS AND FUTURE WORK

This project demonstrated that incorporating sentiment analysis into stock price prediction models does not always lead to improved accuracy. While sentiment data from financial news and social media can provide valuable insights, traditional models like SARIMAX did not benefit from sentiment as expected. More advanced models, such as Random Forest and LSTM, showed better performance, but adding sentiment data did not consistently enhance their predictive power. The findings highlight the complexity of stock price prediction and the need for more sophisticated models and better handling of sentiment data to achieve accurate forecasts in real-time financial markets.

Future work could focus on exploring alternative ways of integrating sentiment data, such as weighting sentiment scores based on relevance, filtering the noise or using more granular data from diverse news sources and social media platforms. It may also be beneficial to experiment with other advanced machine learning techniques, such as attention-based models, reinforcement learning, or Transformers to capture the dynamic nature of stock market behavior. Additionally, improving the quality and scope of sentiment data, possibly by incorporating more diverse factors and events, could help better capture the true impact of news and social media on stock prices.

7. DATA AVAILABILITY

The data and the codes are available at GitHub Repository.

8. ACKNOWLEDGEMENTS

- Bloomberg Terminal for providing the sentiment data.
- Yahoo Finance and TC 2000 for offering historical stock data for the Nasdaq 100 and S&P 500 indices.
- Open Source Libraries for their invaluable contributions to data science and machine learning.
- ChatGPT for assisting in editing, proofreading, and refining the project's documentation and report.
- Dr. Wardat and TA Anas Raza for their guidance, support, and valuable feedback throughout this project and the entire semester. Their expertise and encouragement were essential in shaping this work.

A. Bloomberg Terminal Graphs and Data

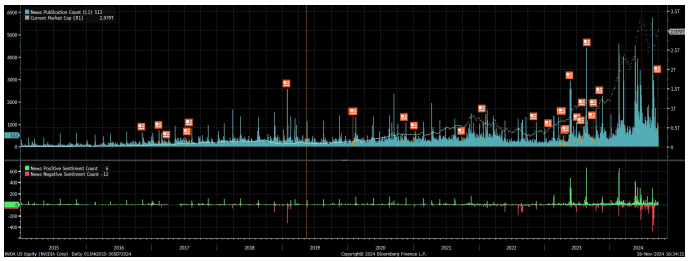

Figure 1. Nvidia News Sentiment Data Bloomberg Terminal

Figure 2. Nvidia News Sentiment Data Bloomberg Terminal

Figure 3. Nvidia News Sentiment Data as seen in Bloomberg Terminal

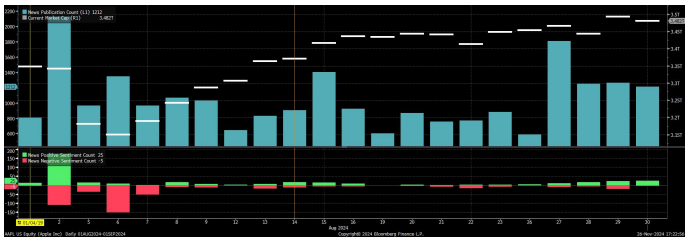

Figure 4. Apple News August 2024 Sentiment Data Bloomberg Terminal

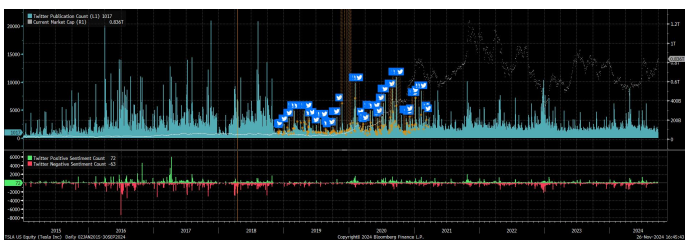

Figure 5. Tesla Twitter Sentiment Data Bloomberg Terminal

B. Stock and Sentiment Data

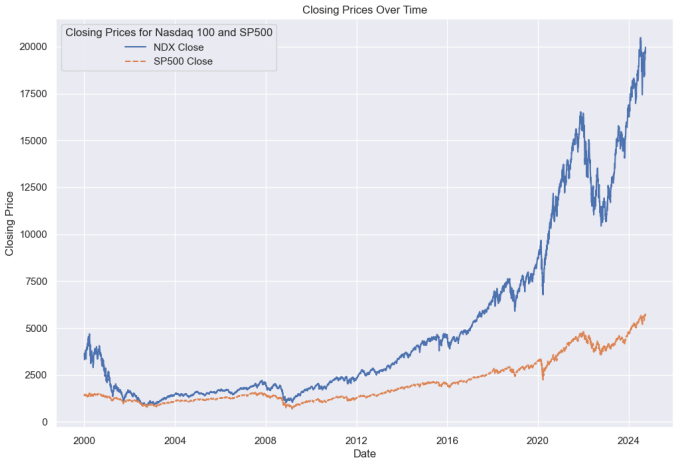

Figure 6. Closing Price over Time for both indices

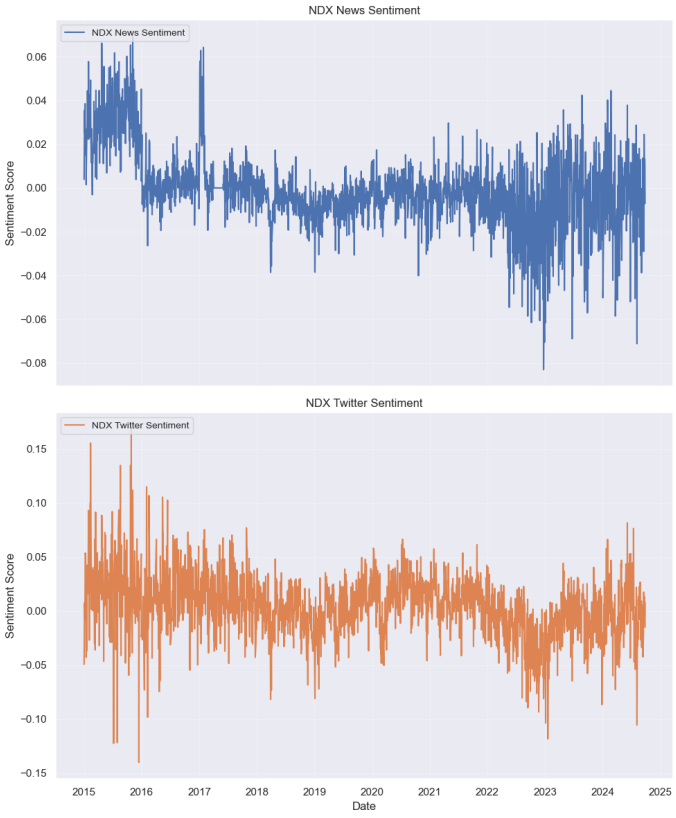

Figure 7. Sentiment Scores for Nasdaq 100

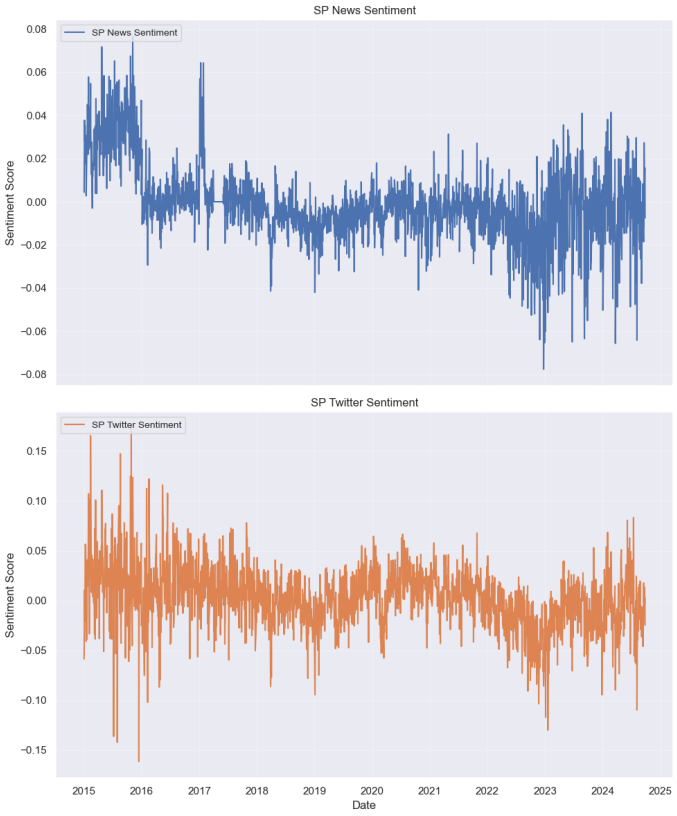

Figure 8. Sentiment Scores for S&P 500

```
                               SARIMAX Results
==========================================================================================
Dep. Variable:                                 y   No. Observations:                 1960
Model:             SARIMAX(0, 1, 1)x(0, 0, 1, 5)   Log Likelihood              -12160.702
Date:                           Sun, 01 Dec 2024   AIC                          24327.404
Time:                                   23:56:11   BIC                          24344.145
Sample:                                        0   HQIC                         24333.558
                                            - 1960
Covariance Type:                             opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1          0.1335      0.013      9.934      0.000       0.107       0.160
ma.S.L5       -0.0348      0.015     -2.328      0.020      -0.064      -0.006
sigma2      1.442e+04    245.354     58.760      0.000    1.39e+04    1.49e+04
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):              2419.70
Prob(Q):                              0.92   Prob(JB):                         0.00
Heteroskedasticity (H):               0.06   Skew:                             0.46
Prob(H) (two-sided):                  0.00   Kurtosis:                         8.37
===================================================================================
```

Figure 10. Nasdaq 100 SARIMAX Summary

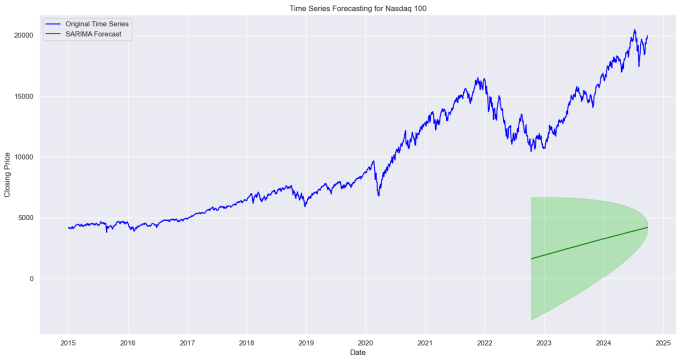

Figure 11. Nasdaq 100 SARIMAX Forecast

C. Time Series Analysis

Figure 9. Time Series Analysis for Nasdaq 100

Figure 12. Time Series Analysis for S&P 500

SARIMAX Results

Dep. Variable:	y	No. Observations:	1960
Model:	SARIMAX(2, 1, 2)x(1, 0, [1], 5)	Log Likelihood	-9746.363
Date:	Sun, 01 Dec 2024	AIC	19506.727
Time:	23:53:34	BIC	19545.788
Sample:	0	HQIC	19521.084
	- 1960		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-1.7466	0.021	-82.125	0.000	-1.788	-1.705
ar.L2	-0.8747	0.021	-42.442	0.000	-0.915	-0.834
ma.L1	1.6640	0.028	58.769	0.000	1.609	1.719
ma.L2	0.7673	0.028	27.160	0.000	0.712	0.823
ar.S.L5	0.7654	0.087	8.757	0.000	0.594	0.937
ma.S.L5	-0.8095	0.080	-10.136	0.000	-0.966	-0.653
sigma2	1227.0407	20.010	61.322	0.000	1187.822	1266.259

Ljung-Box (L1) (Q):	0.73	Jarque-Bera (JB):	3931.24
Prob(Q):	0.39	Prob(JB):	0.00
Heteroskedasticity (H):	0.11	Skew:	0.86
Prob(H) (two-sided):	0.00	Kurtosis:	9.72

Figure 13. S&P 500 SARIMAX Summary

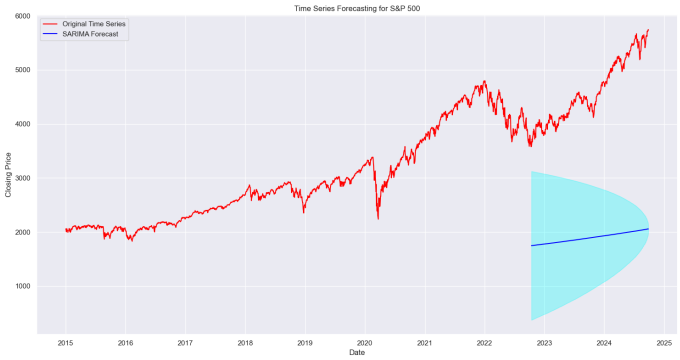

Figure 14. S&P 500 SARIMAX Forecast

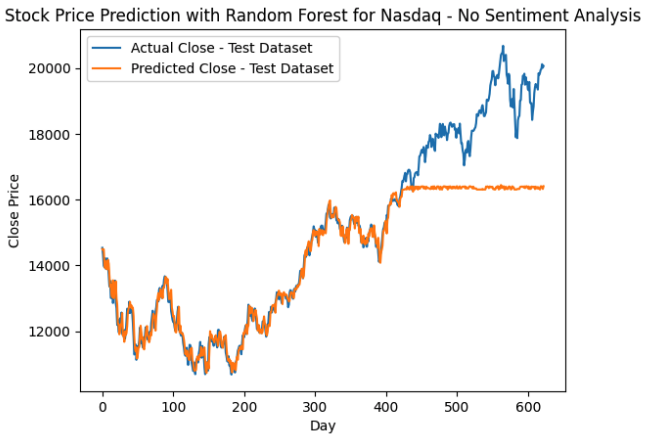

Figure 15. Nasdaq 100 Random Forest Model

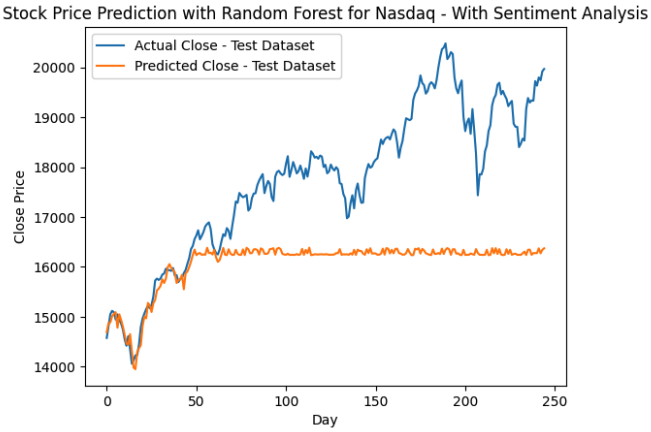

Figure 16. Nasdaq 100 Random Forest Model with Sentiment Score

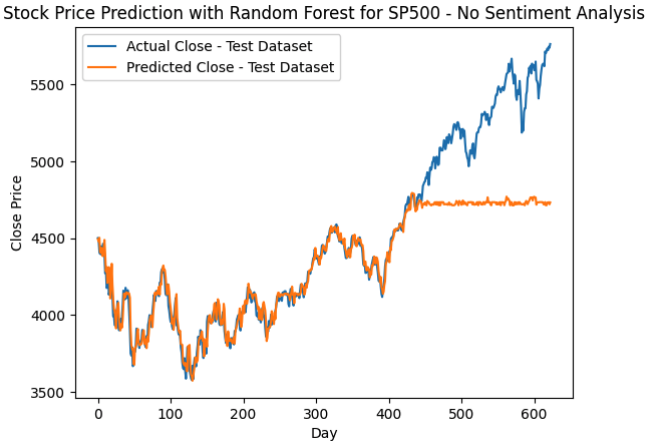

Figure 17. S&P 500 Random Forest Model

D. Random Forest Model

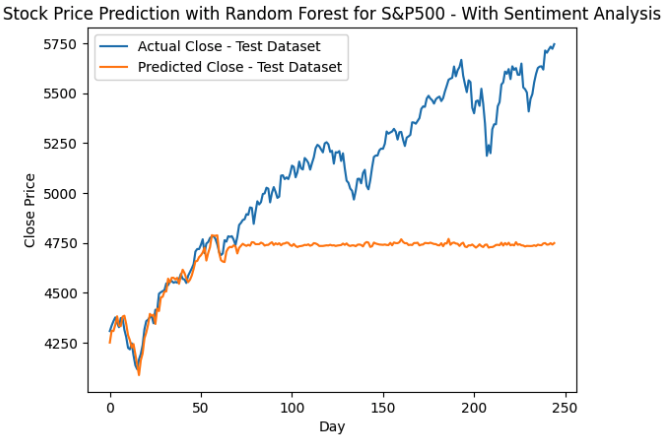

Figure 18. S&P 500 Random Forest Model with Sentiment Score

E. LSTM Model

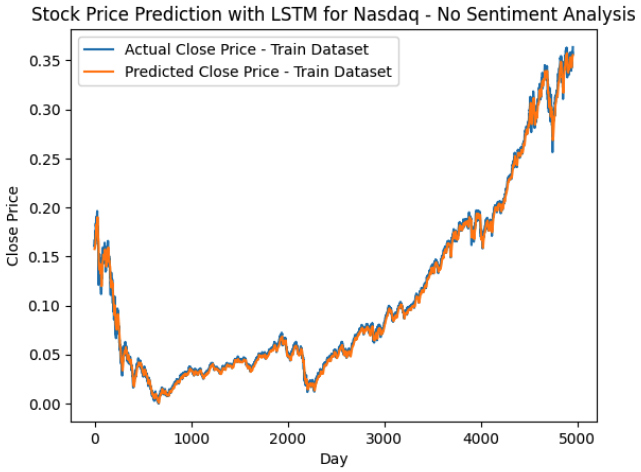

Figure 19. Nasdaq 100 LSTM Model on Training Data

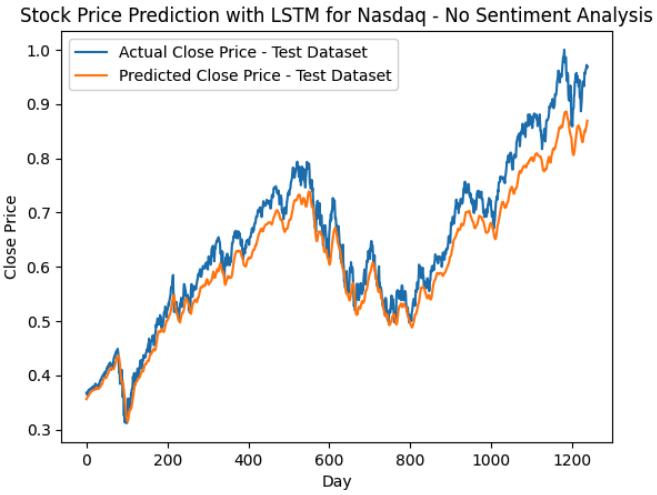

Figure 20. Nasdaq 100 LSTM Model on Testing Data

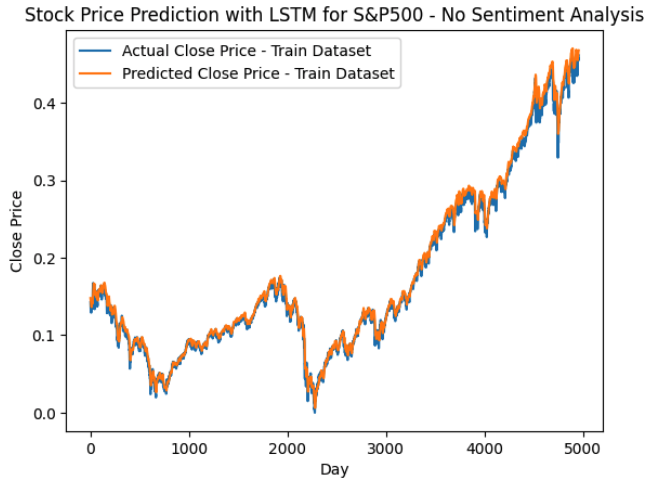

Figure 21. S&P 500 LSTM Model on Training Data

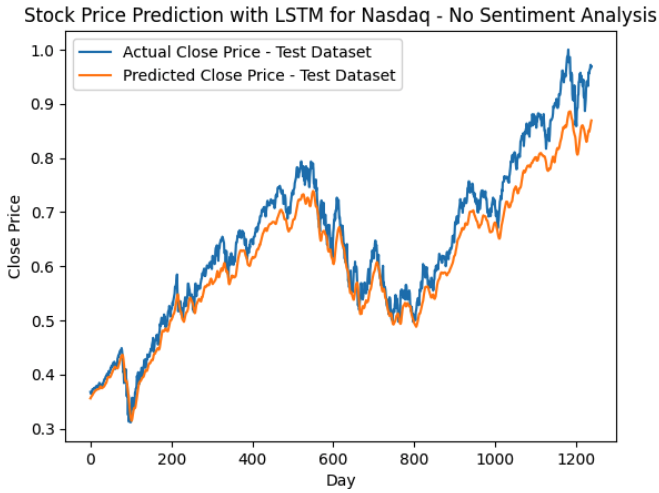

Figure 22. S&P 500 LSTM Model on Testing Data

References

[1] Xin Cui, Daniel Lam, Arun Verma. "Embedded Value in Bloomberg News & Social Sentiment Data" *Bloomberg L.P.*

[2] Jason Brownlee. *Deep Learning for Time Series Forecasting*. Machine Learning Mastery, 2020. https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/

[3] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2019. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

[4] Bing Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012. https://www.cs.uic.edu/~liub/FBS/Sentiment-Analysis-and-Opinion-Mining.pdf

[5] Various Authors. "Financial News Predicts Stock Market Volatility Better Than Close Price." *Journal of Financial Economics*, 2019.

[6] Various Authors. "An Empirical Analysis of Stock Price Prediction Using ARIMA and LSTM." *Journal of Economic Dynamics and Control*, 2021.

[7] Various Authors. "Sentiment Analysis in Financial Texts." *Finance and Data Science*, 2022.

[8] Medium Author. "Sentiment Analysis with Python - A Beginner's Guide." *Medium Article*, 2023. https://medium.com/@eleanor.watson/a-beginners-guide-to-performing-sentiment-analysis-on-text-with-python-3ce80dcac22e

[9] Jason Brownlee. "Introduction to Time Series Forecasting with Python." *Machine Learning Mastery Blog*, 2023. https://machinelearningmastery.com/time-series-forecasting/

[10] Bloomberg. "Bloomberg Python API: blpapi." 2023. https://bloomberg.github.io/blpapi-python/

[11] Yahoo Finance. "Yahoo Finance API Documentation." 2023. https://www.yahoofinanceapi.com/

[12] News API. "News API Documentation." 2023. https://newsapi.org/docs/