Stock Price Prediction based on News Sentiment Analysis

By Musaddiq Kamal Khan (22k- 44320) and Ahmed Yoshay (22k-4396)

Contents

[Project Objective 2](#_Toc197971990)

[Dataset Used 2](#_Toc197971991)

[Models Used 2](#_Toc197971992)

[Results 3](#_Toc197971993)

[Discussion on results 4](#_Toc197971994)

# Project Objective

The objective of this project is to develop a predictive model that forecasts stock price movements based on the sentiment of news articles related to financial markets and specific companies. By leveraging natural language processing (NLP) techniques to analyze the tone and context of news content, the model aims to classify news as positive, negative, or neutral, and correlate these sentiments with corresponding stock price trends.

# Dataset Used

The dataset we used was created by us by combining data from different sources. First, we used the yfinance library to get the list of companies in the S&P 500, which is an index of 500 large companies listed in the US stock market. Then, for each company, we used the GNews library to collect news articles related to them. GNews is a simple tool from GitHub that lets you get news without needing any API key. To analyze the sentiment of these news articles, we used a model called ProsusAI/finbert, which is specifically made for financial news. It gave us a sentiment label (positive, neutral, or negative) and a sentiment score for each article. After that, we combined the stock price data (Open, High, Low, Close, Volume) with the average sentiment of news articles published before each stock price date. This final dataset, which includes both price and sentiment data, was then used to train our LSTM model.

# Models Used

For this project, we used two main models: one for analyzing sentiment and another for predicting stock prices. For sentiment analysis, we used the ProsusAI/finbert model. It's a pre-trained BERT model designed specifically for financial texts, which helps us understand whether the news about a company is positive, neutral, or negative. It also gives a sentiment score that shows how strong the sentiment is.

For stock price prediction, we built an LSTM (Long Short-Term Memory) model using Keras. Our model has the following structure:

* **First LSTM layer with 64 units**: This layer looks at sequences of stock and sentiment data and learns patterns over time. We set return\_sequences=True so it outputs the full sequence to the next LSTM layer.
* **Dropout layer (0.2)**: Dropout helps prevent overfitting by randomly turning off 20% of the neurons during training, so the model doesn’t rely too heavily on any one part of the data.
* **Second LSTM layer with 64 units**: This layer continues learning from the previous one but doesn’t return a sequence—it outputs just one vector.
* **Another Dropout (0.2)**: Again used to reduce overfitting and improve generalization.
* **Dense layer with 1 unit**: This is the final output layer, and it predicts the closing stock price.

We compiled the model using the **Adam optimizer**, which is widely used for training deep learning models as it adapts the learning rate during training. The loss function is **mean squared error (MSE)**, which is common for regression tasks like predicting stock prices

# Results

To get the best performance from the LSTM model, I wrote a script that automatically tested different combinations of sequence lengths and training epochs. The sequence length controls how many previous days the model looks at when making a prediction, and the number of epochs determines how many times the model goes over the training data.

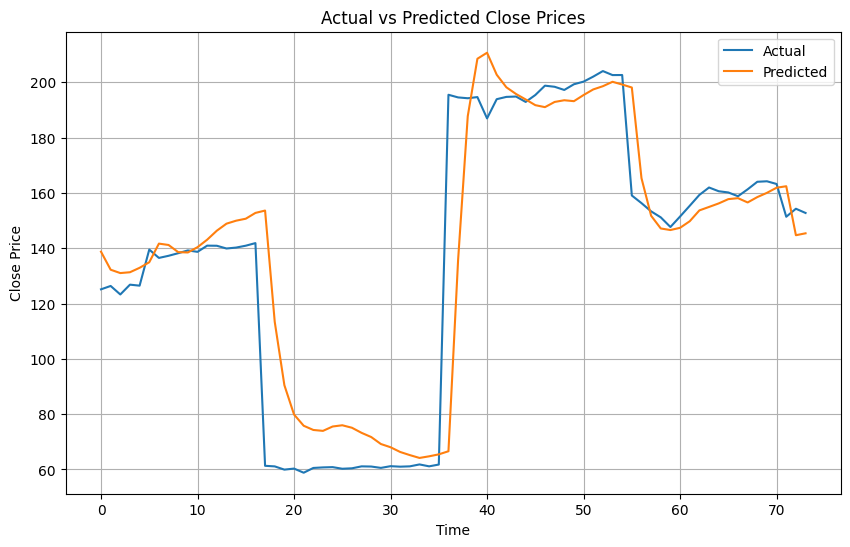
After trying out several combinations, the best results came from using a **sequence length = 10** and training the model for **100 epochs**. This setup gave the most accurate predictions based on the evaluation metrics.

Here are the final performance results:

* **Mean Absolute Error (MAE):** 11.68
* **Root Mean Squared Error (RMSE):** 22.82
* **R² Score:** 0.7951
* **Directional Accuracy:** 70.59%

These results show that the model was able to capture trends in the stock prices quite well, especially considering we included sentiment data, which adds another layer of complexity. The R² score of 0.7951 means the model explains about 80% of the variation in stock prices, and the Directional Accuracy of 70.59% is also very good, indicating that the model successfully predicts the direction of price movements more than 70% of the time.

Below is a plot comparing the **actual vs. predicted closing prices** over time. As you can see in the graph, the model was able to closely follow the real price movements.



# Discussion on results

Overall, the model performed quite well, achieving a good level of accuracy in predicting stock prices based on both historical price data and news sentiment. The R² score of 0.7951 demonstrates that the model was able to explain about 80% of the variation in stock prices, while the Directional Accuracy of 70.59% indicates that the model correctly predicted the direction of price movements over 70% of the time. These metrics show that the model effectively captured the underlying trends in the stock prices, with sentiment information adding valuable context beyond just the price history.

However, there are a few limitations to consider. One major factor is that the model was trained on data from only a limited number of companies. This means the model might not generalize as well to other companies or industries with different behavior or news patterns. Training on a broader dataset could improve performance and robustness.

Another important point is that while we included sentiment scores from financial news, we didn't check whether the news was actually true or misleading. Sometimes, false or sensational headlines can skew sentiment without reflecting reality. Adding a feature that verifies the factual accuracy of news articles could help filter out noise and improve the model’s predictions by focusing on more reliable information.

# Links

**CODE FILES AND DATASET:**

<https://drive.google.com/drive/folders/11CJmWIVL8M1oH7reY13WvUAKzZ_Hq1F3?usp=sharing>