Stock Price Prediction: Time-Series Forecasting with PyTorch and LSTMs

A Project Report

Jeet Prajapati

May 30, 2025

Contents

Abstract

This project presents a comprehensive analysis and predictive model for the stock prices of five leading technology corporations: NVIDIA (NVDA), Apple (AAPL), Microsoft (MSFT), Google (GOOGL), and Amazon (AMZN). Utilizing a 15-year historical dataset (2010-2024), this report begins with an in-depth Exploratory Data Analysis (EDA) to uncover trends, volatility patterns, and comparative growth trajectories. Following the EDA, a Long Short-Term Memory (LSTM) recurrent neural network is developed using the PyTorch framework. The model is trained to forecast future closing prices based on historical sequences. The report concludes with a quantitative evaluation of the model's performance using Root Mean Squared Error (RMSE) and a qualitative assessment through detailed, interactive visualizations, demonstrating the efficacy of deep learning in the complex domain of financial forecasting.

1 Introduction and Project Setup

1.1 The Challenge of Financial Forecasting

Predicting stock market movements is a notoriously difficult task due to the market's inherent volatility and its sensitivity to a vast number of factors, from macroeconomic indicators to public sentiment. While no model can predict the future with perfect accuracy, modern machine learning techniques can identify complex, non-linear patterns in historical data that are often invisible to traditional statistical methods. This project aims to apply one such technique, a Long Short-Term Memory (LSTM) network, to this challenge.

1.2 Dataset and Environment

- **Dataset:** The analysis is based on the `15Y Stock Data: NVDA, AAPL, MSFT, GOOGL & AMZN` dataset. It contains daily trading data, including Open, High, Low, Close, and Volume for each of the five stocks from January 2010 to early 2025.
- **Environment:** The entire project is developed in Google Colab, leveraging its cloud-based computational resources. The key libraries employed include:
  - `pandas` & `numpy`: For efficient data manipulation and numerical operations.
  - `matplotlib` & `seaborn`: For static Exploratory Data Analysis visualizations.
  - `plotly`: For creating interactive, publication-quality charts.
  - `scikit-learn`: For data preprocessing, specifically data scaling.
  - `torch`: The core deep learning framework used to build and train our LSTM model.

2 Exploratory Data Analysis (EDA): Uncovering Market Behavior

Before building a predictive model, it is crucial to thoroughly understand the data. EDA helps us visualize the historical behavior of each stock, identify key characteristics, and form hypotheses about their market dynamics.

2.1 Visualizing Trends with Moving Averages

A moving average is a widely used technical indicator that helps smooth out price action by filtering out short-term noise. We plot the 100-day and 200-day Moving Averages (MA) to better identify the medium and long-term trends.

Interpretation: When the daily price is consistently above its MAs, it signals a strong bullish (upward) trend. Conversely, when it trades below its MAs, it indicates a bearish (downward) trend. Crossovers between the price and its MAs are often interpreted by traders as potential buy or sell signals.

2.2 Analyzing Daily Trading Metrics

By plotting the **Open, High, Low, Close, and Volume** on individual subplots, we gain a granular view of each stock's daily trading characteristics.

- **Price Range (High-Low):** A wider daily range suggests higher intraday volatility.
- **Volume:** Spikes in trading volume often coincide with significant price movements, earnings announcements, or major news events, indicating periods of high market interest and activity.

2.3 NVIDIA (NVDA) Analysis

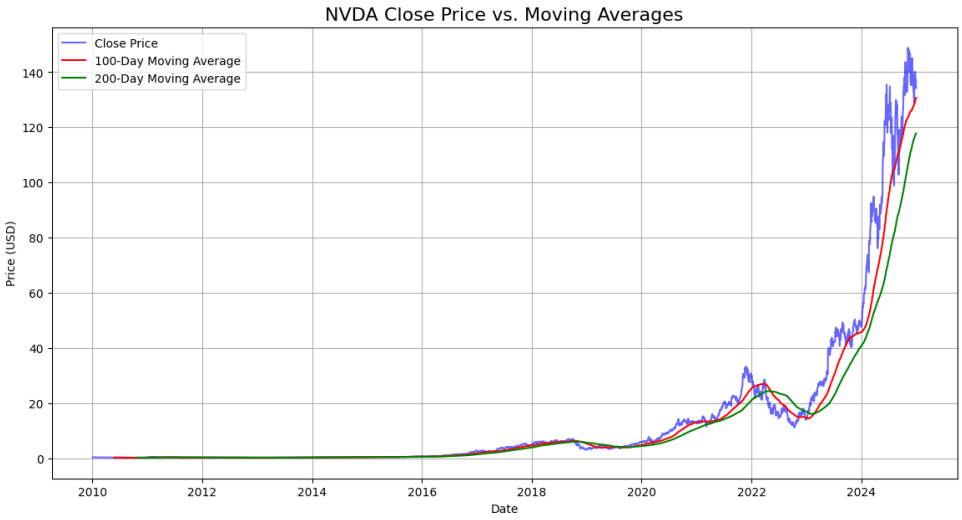

Figure 1: NVDA Close Price vs. Moving Averages.

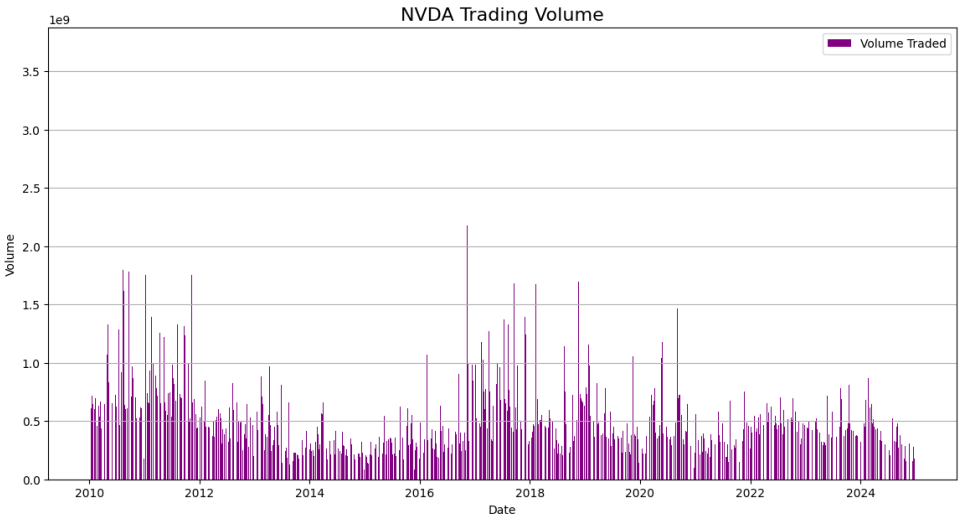

Figure 2: NVDA Daily Trading Volume.

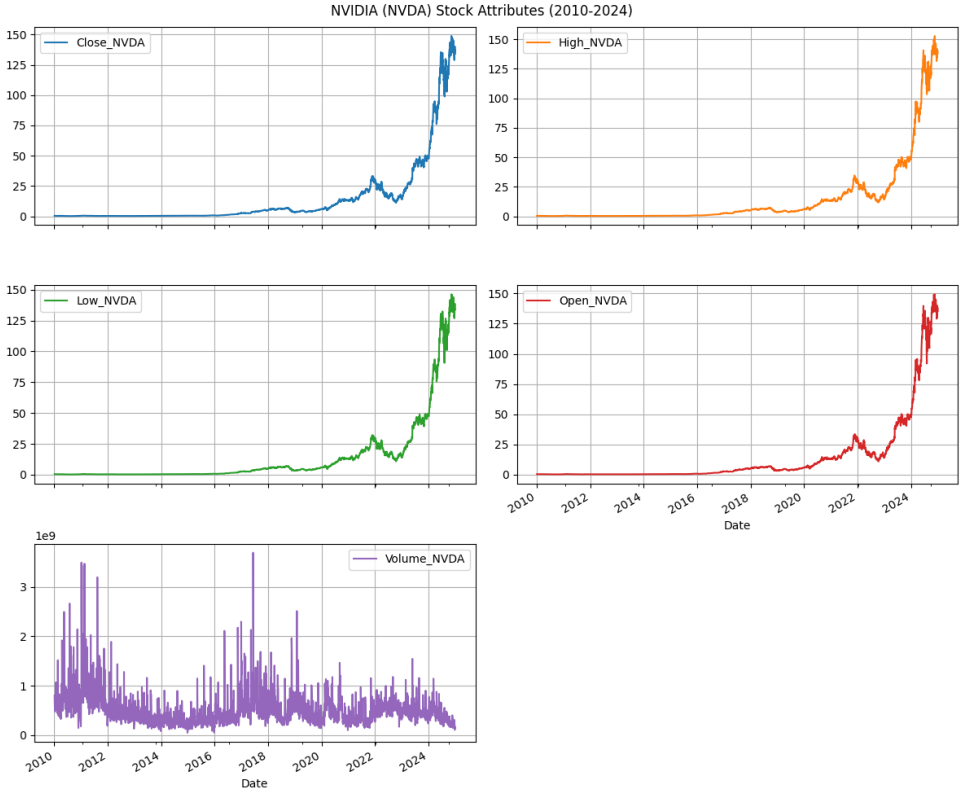

Figure 3: NVDA Daily Stock Attributes.

2.4 Apple (AAPL) Analysis

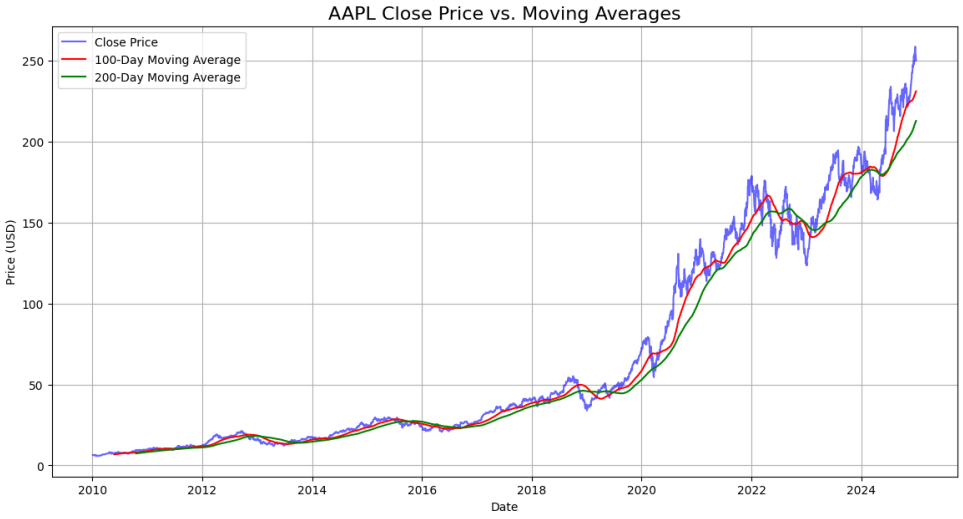

Figure 4: AAPL Close Price vs. Moving Averages.

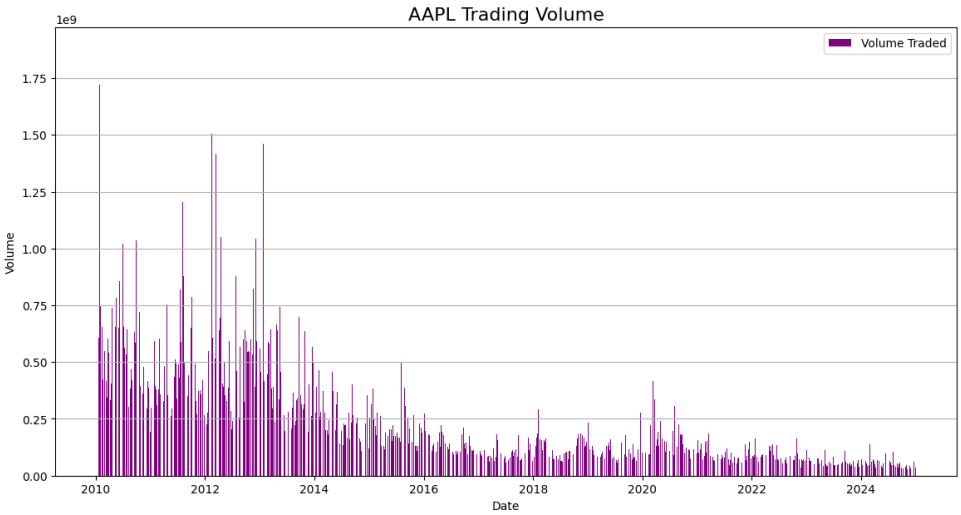

Figure 5: AAPL Daily Trading Volume.

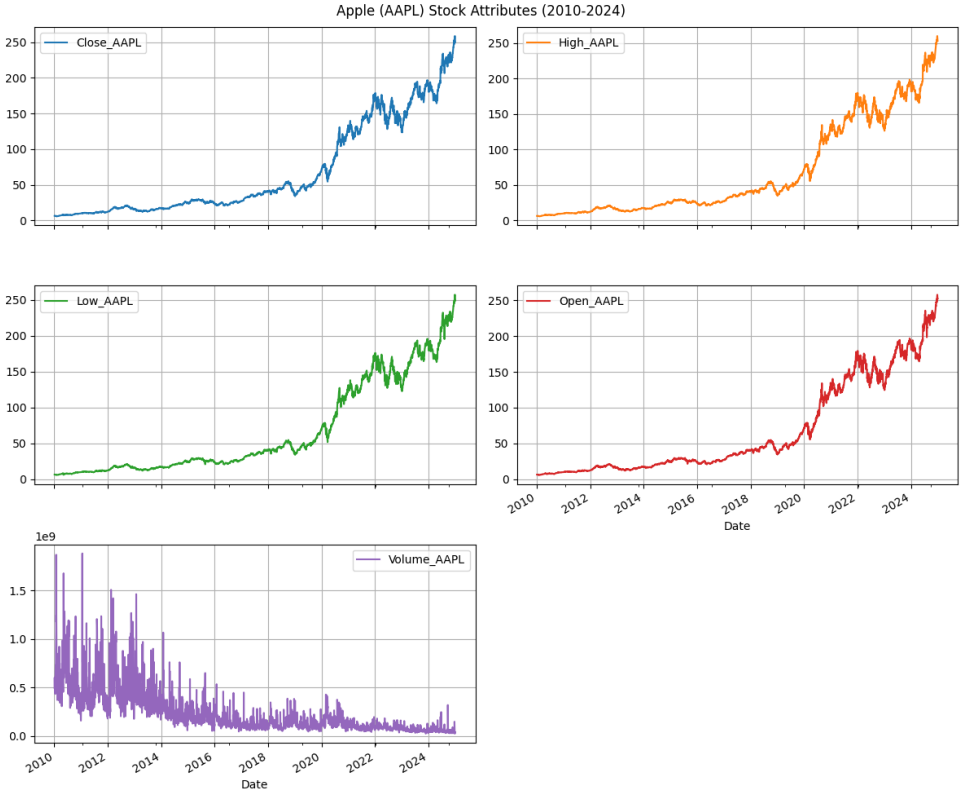

Figure 6: AAPL Daily Stock Attributes.

2.5 Microsoft (MSFT) Analysis

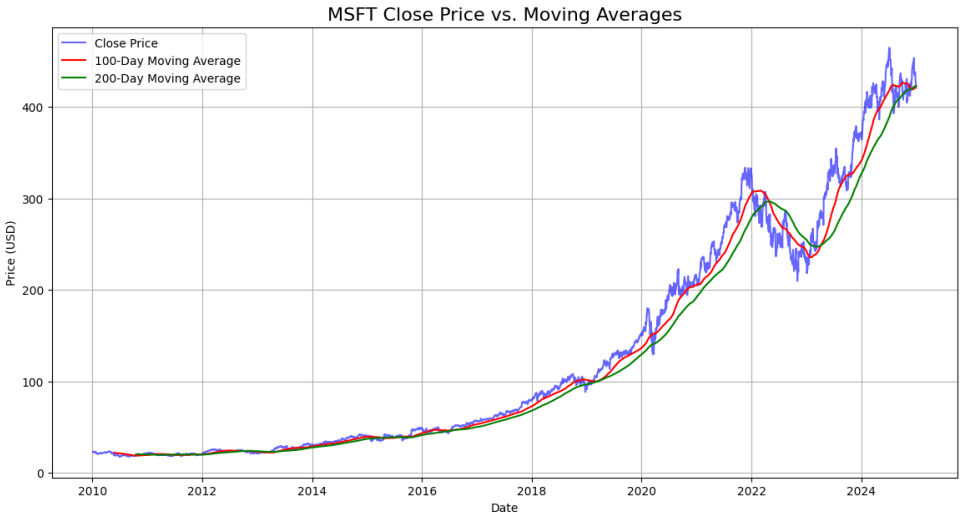

Figure 7: MSFT Close Price vs. Moving Averages.

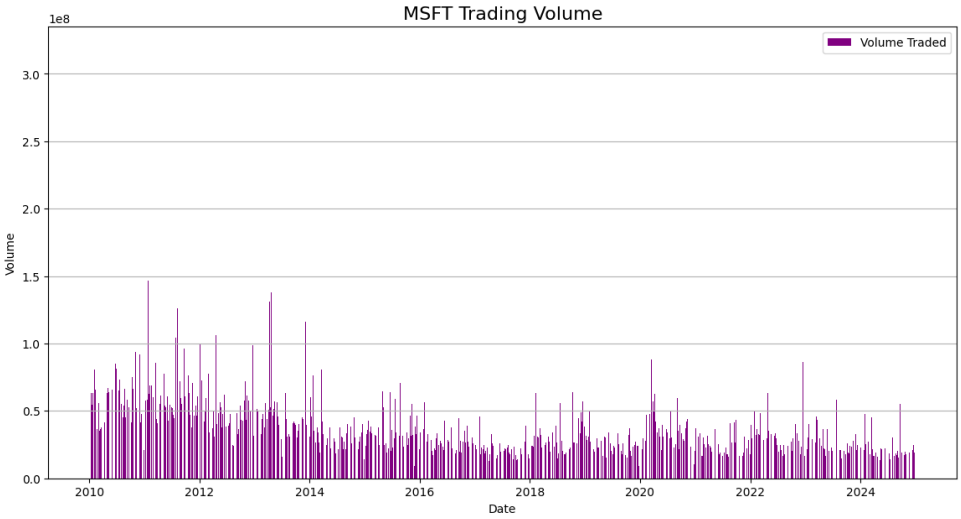

Figure 8: MSFT Daily Trading Volume.

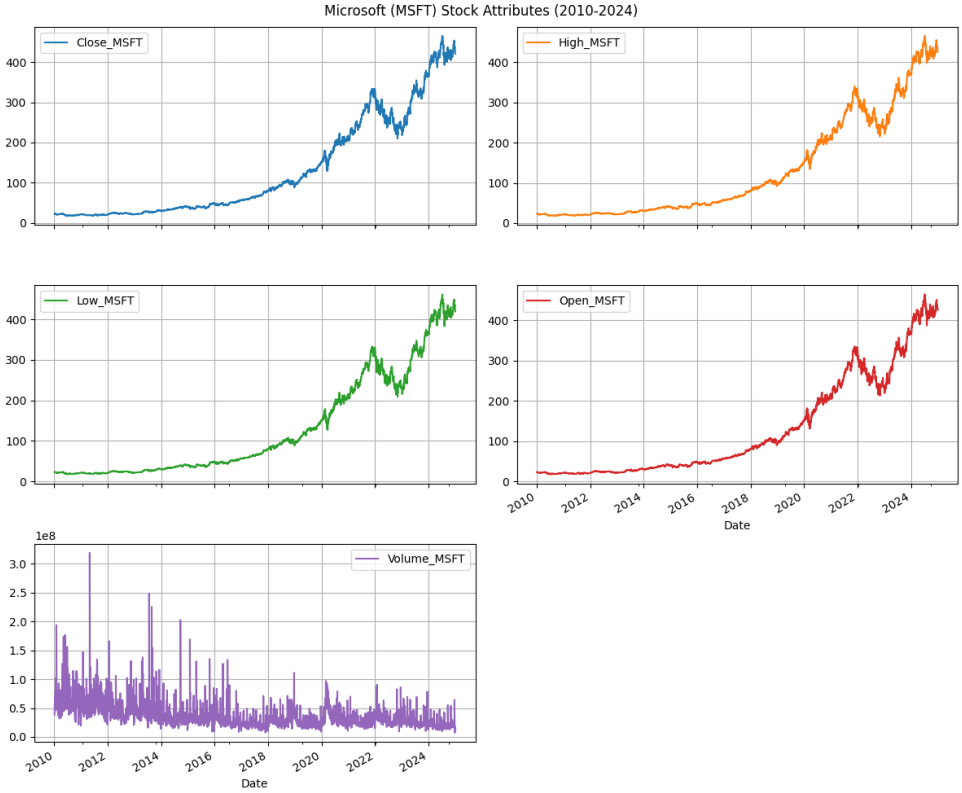

Figure 9: MSFT Daily Stock Attributes.

2.6 Google (GOOGL) Analysis

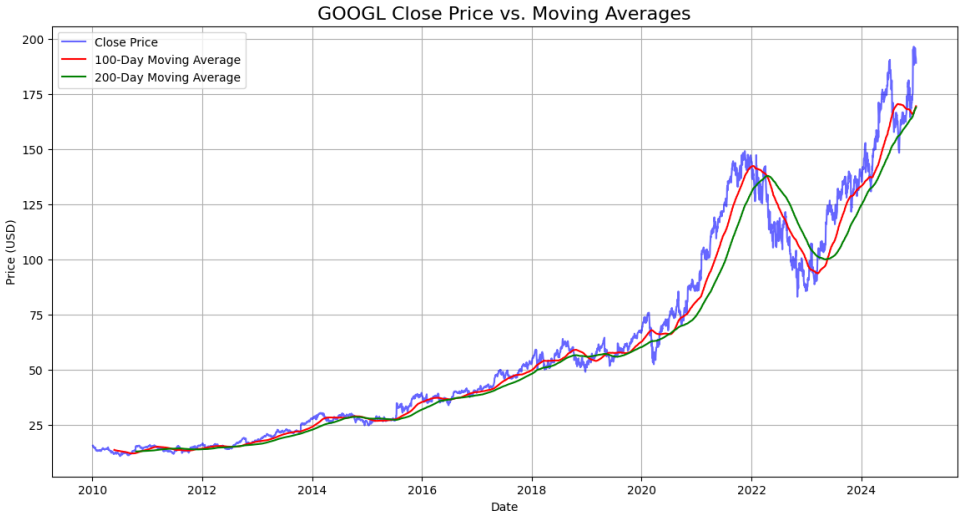

Figure 10: GOOGL Close Price vs. Moving Averages.

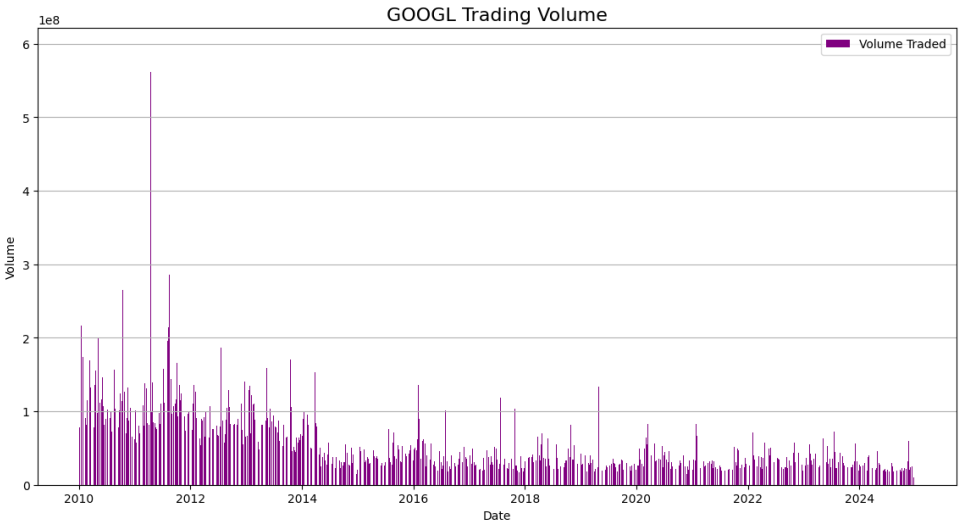

Figure 11: GOOGL Daily Trading Volume.

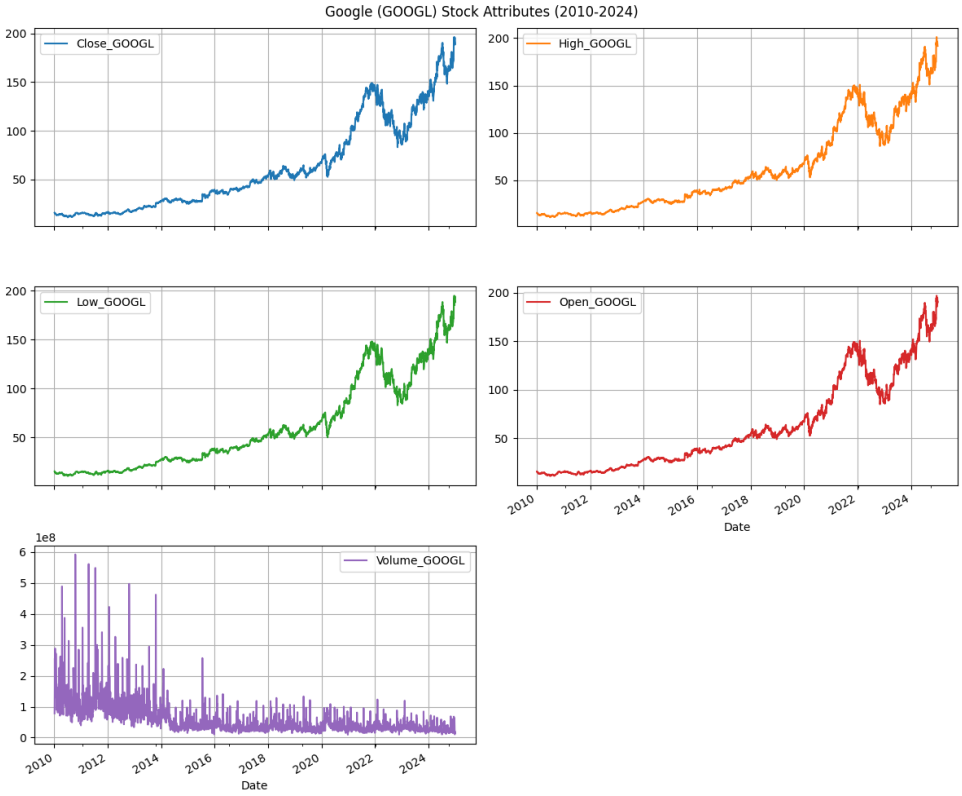

Figure 12: GOOGL Daily Stock Attributes.

2.7 Amazon (AMZN) Analysis

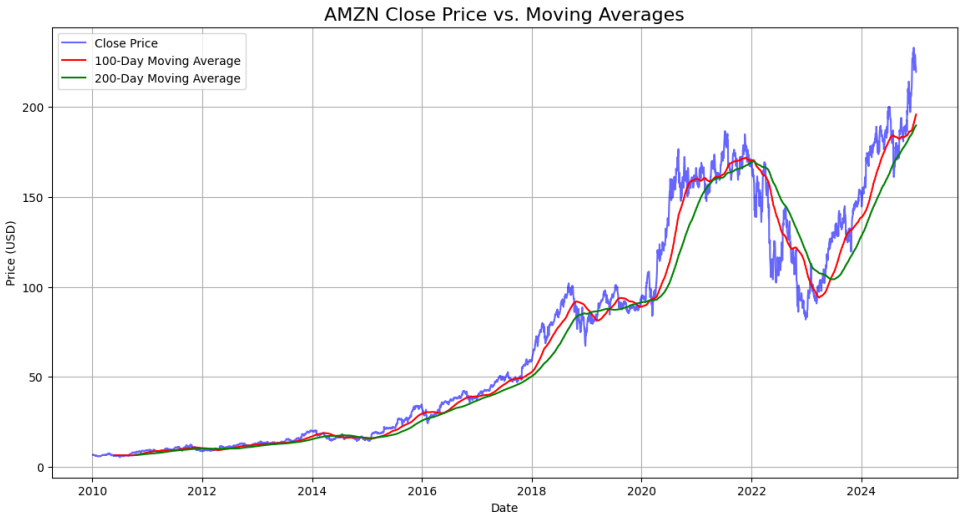

Figure 13: AMZN Close Price vs. Moving Averages.

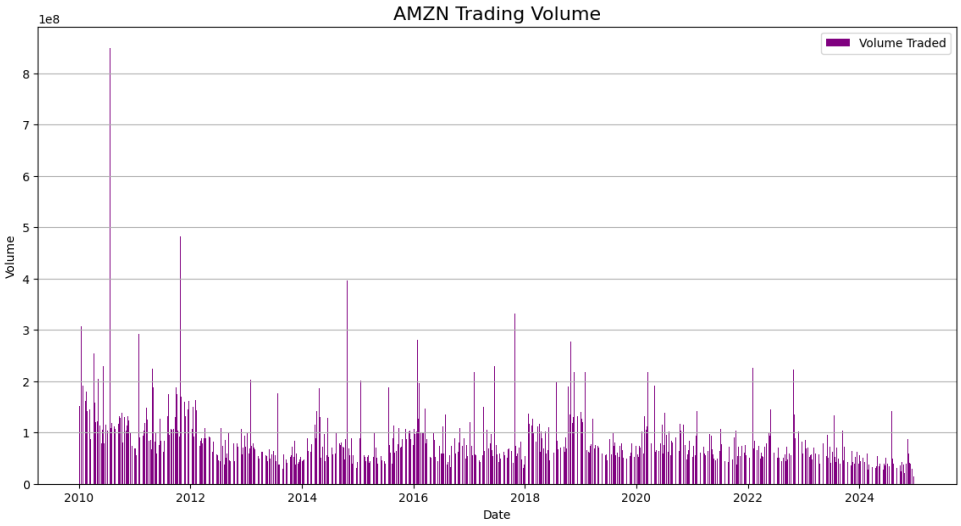

Figure 14: AMZN Daily Trading Volume.

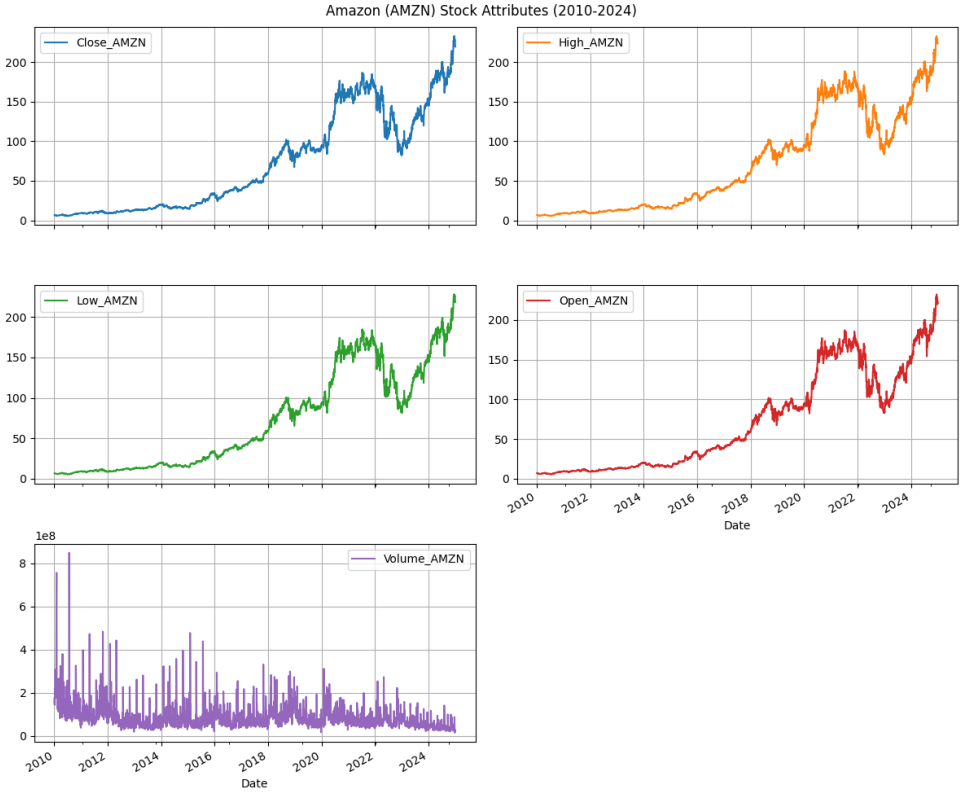

Figure 15: AMZN Daily Stock Attributes.

3 Comparative Analysis: Benchmarking the Tech Giants

While individual analysis is insightful, comparing the stocks directly reveals market leadership and relative performance.

3.1 The Need for Normalization

A direct plot of stock prices can be deceiving due to their different price scales (e.g., NVIDIA's stock price is much higher than Apple's after splits). To create a fair comparison of growth, we **normalize** the prices. By setting the starting 'High' price of every stock to a common baseline of 100, we can effectively visualize their percentage growth over time. This method clearly answers the question: "If you had invested $100 in each of these stocks in 2010, which would have performed best?"

3.2 Combined High Price Comparison

A direct plot of the 'High' price for all five stocks shows their absolute price movements over time.

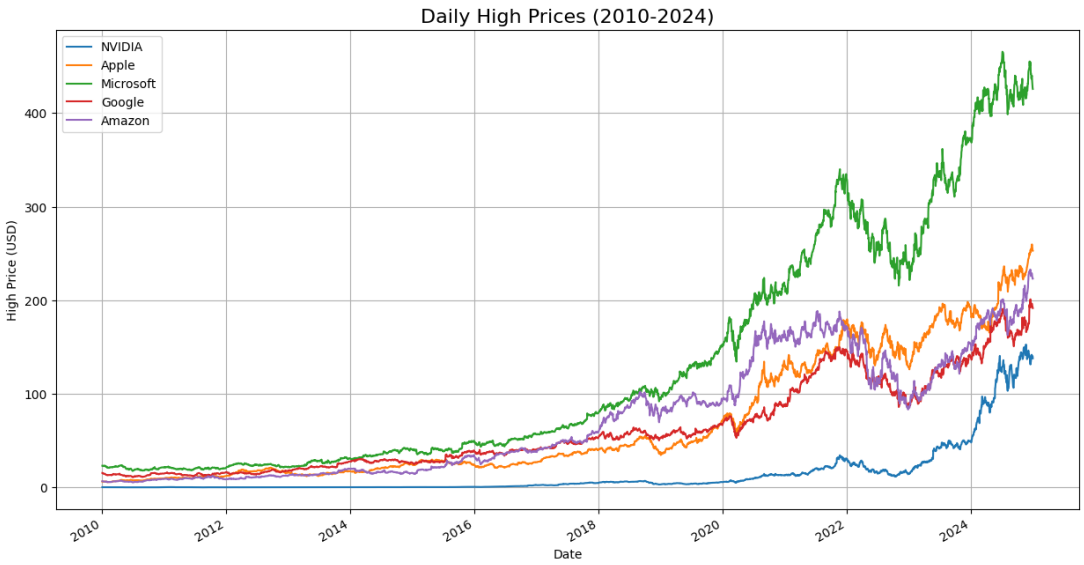

Figure 16: Daily High Prices Comparison for all five stocks.

3.3 Normalized Growth Comparison

To create a fair comparison of growth, we normalize the prices to a common baseline of 100.

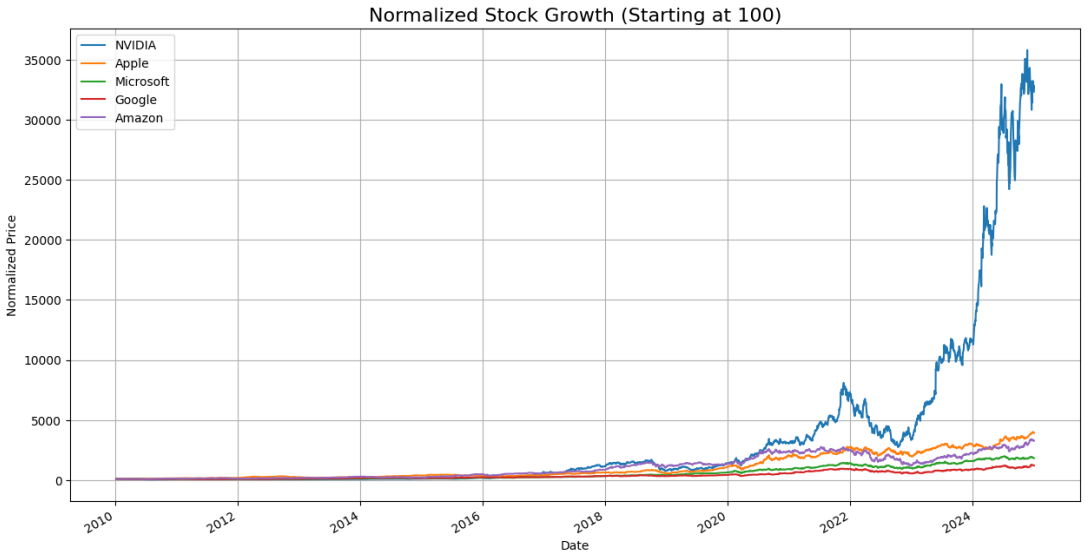

Figure 17: Normalized stock growth comparison.

3.4 Expanding Windows: A Look at Cumulative Performance and Risk

The expanding window analysis provides a dynamic view of a stock's history.

- **Expanding Mean:** This is a cumulative average that shows the long-term momentum of the stock. A steadily increasing expanding mean indicates strong and consistent historical performance.
- **Expanding Standard Deviation:** This is a measure of cumulative volatility, or **risk**. A rising line indicates that the stock's price fluctuations have been increasing over time, suggesting it has become a more volatile and potentially riskier asset.

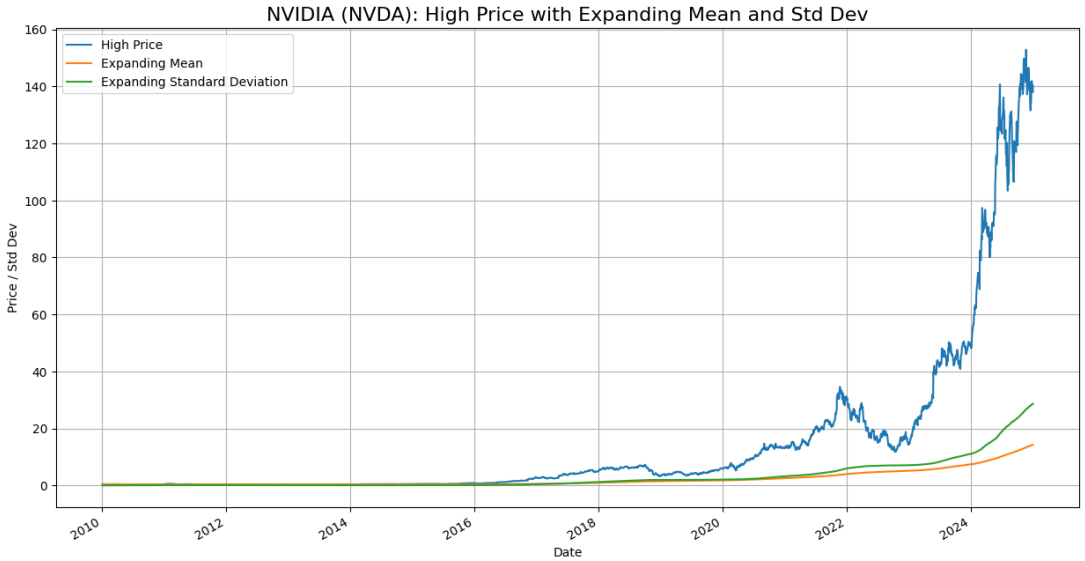

Figure 18: NVIDIA Expanding Mean and Standard Deviation.

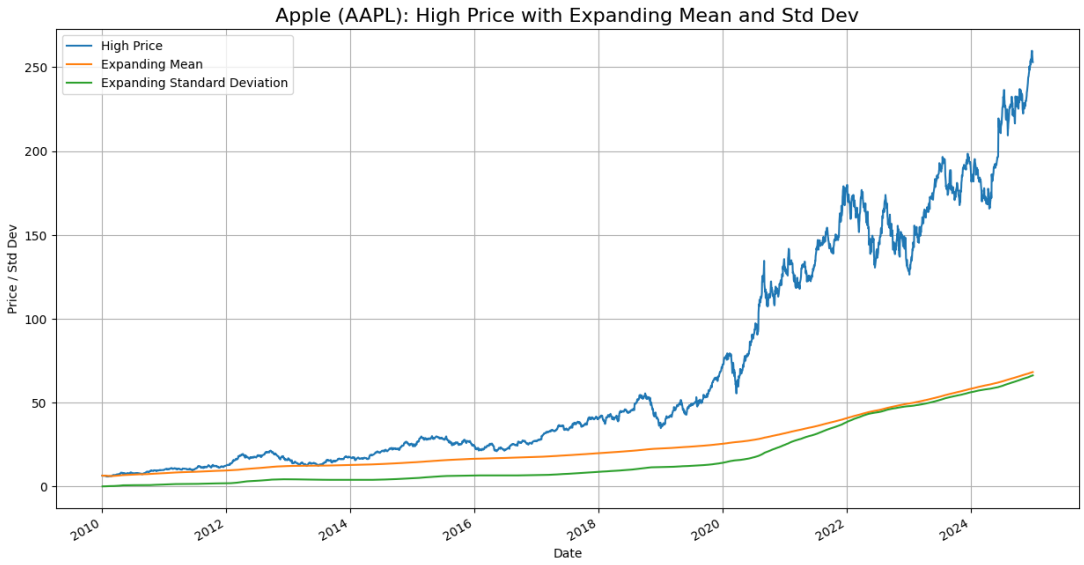

Figure 19: Apple Expanding Mean and Standard Deviation.

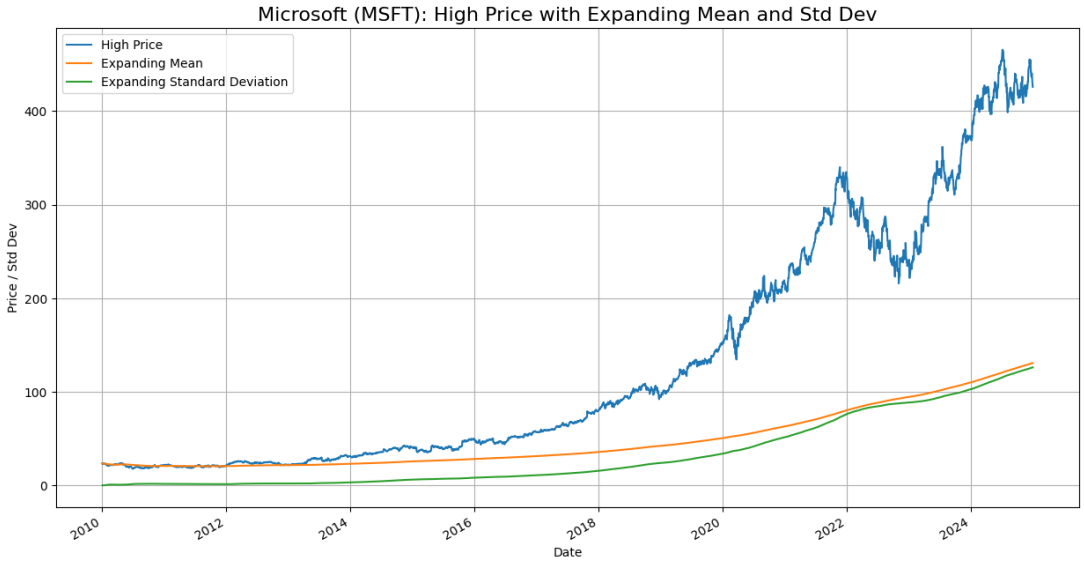

Figure 20: Microsoft Expanding Mean and Standard Deviation.

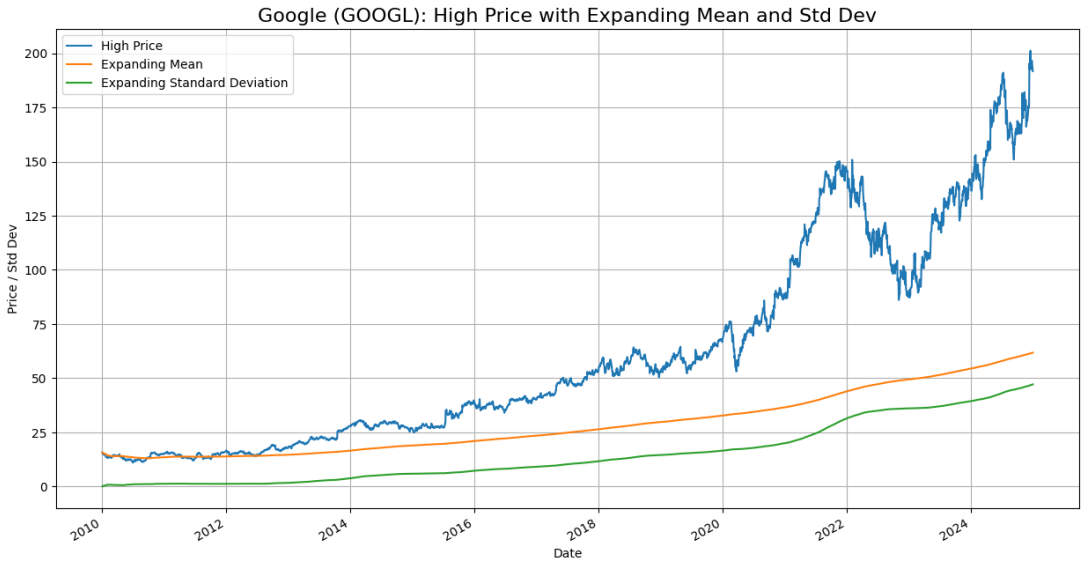

Figure 21: Google Expanding Mean and Standard Deviation.

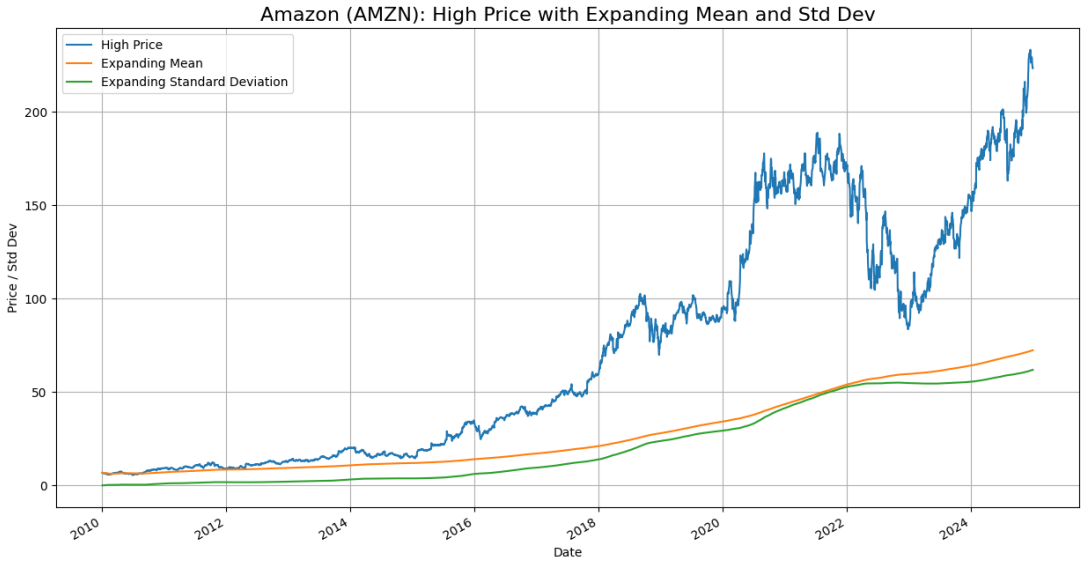

Figure 22: Amazon Expanding Mean and Standard Deviation.

4 Statistical Decomposition of Time-Series Data

To statistically validate the patterns observed during EDA, we decompose the time-series data. This technique separates the stock price into three fundamental components, allowing us to understand its underlying structure.

- **Trend:** This component captures the long-term trajectory of the stock price. For all the tech stocks in our dataset, we expect to see a strong upward trend, reflecting their growth over the last decade.
- **Seasonality:** This component isolates any repeating, fixed-period patterns. For stock data, this might capture annual or quarterly cycles related to business performance or market psychology.
- **Residuals:** This component represents the ”noise” or random, unpredictable fluctuations that remain after the trend and seasonality have been removed. A high degree of residuals indicates significant market irregularity.

This analysis confirms that our data is not random; it possesses a strong, learnable trend, which justifies the use of a sophisticated time-series model like an LSTM.

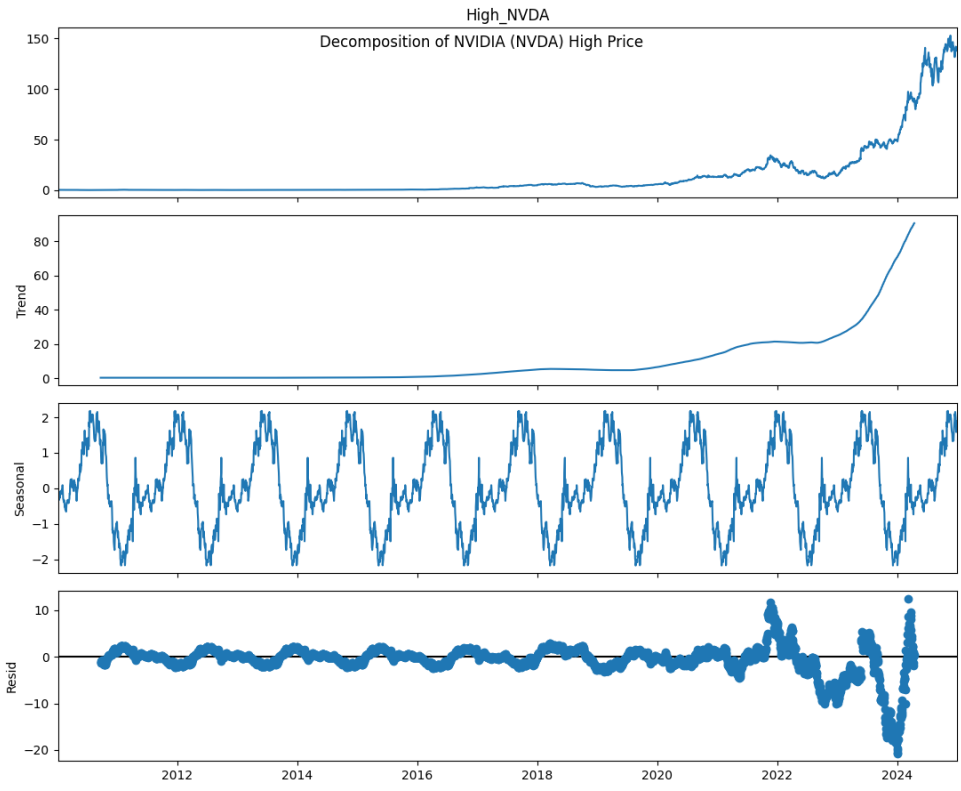

Figure 23: Decomposition of NVIDIA High Price.

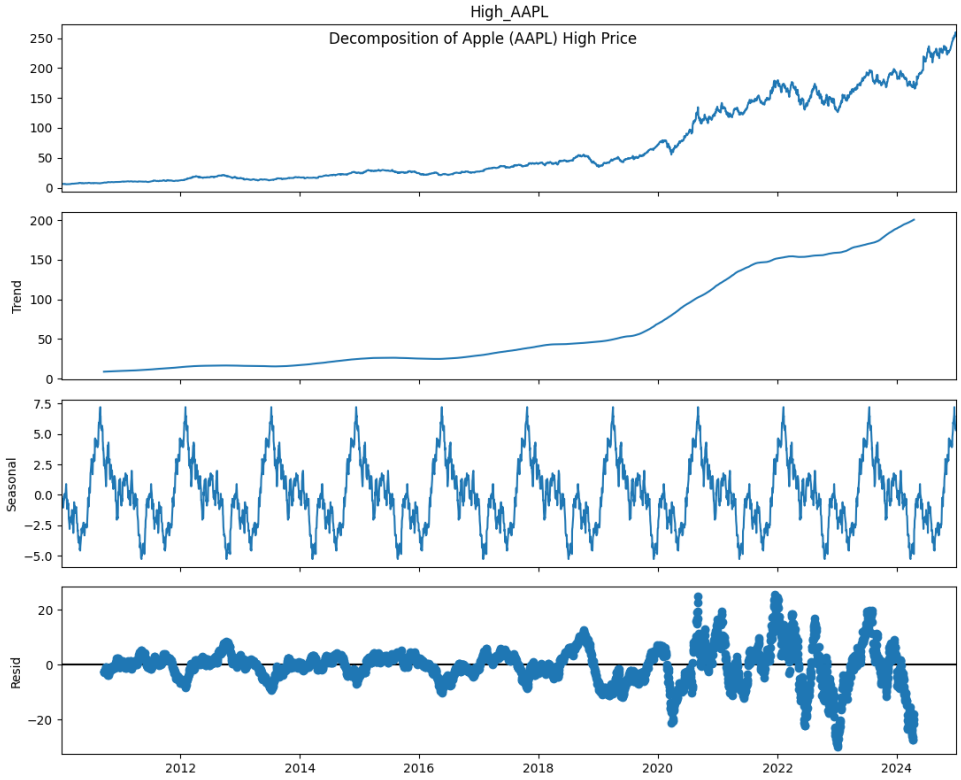

Figure 24: Decomposition of Apple High Price.

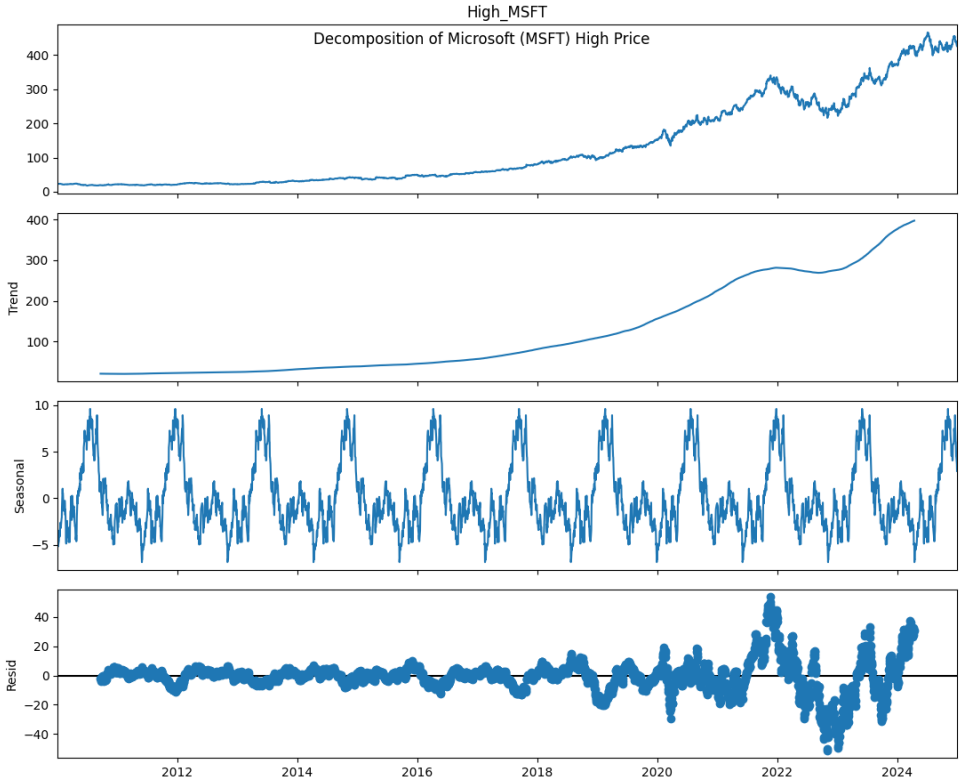

Figure 25: Decomposition of Microsoft High Price.

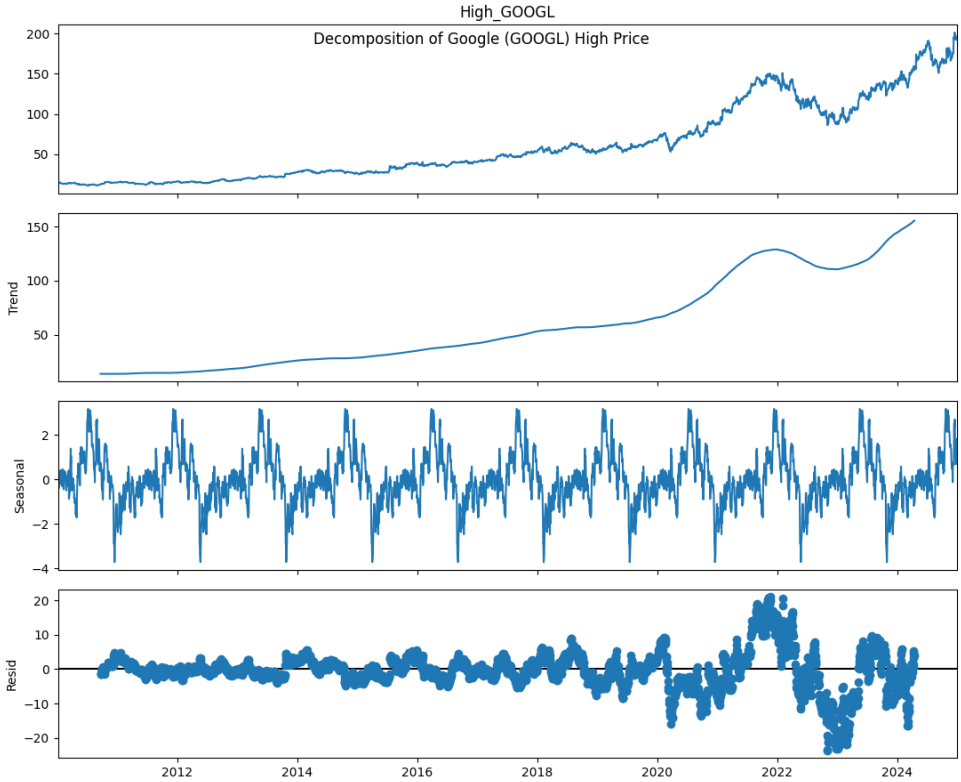

Figure 26: Decomposition of Google High Price.

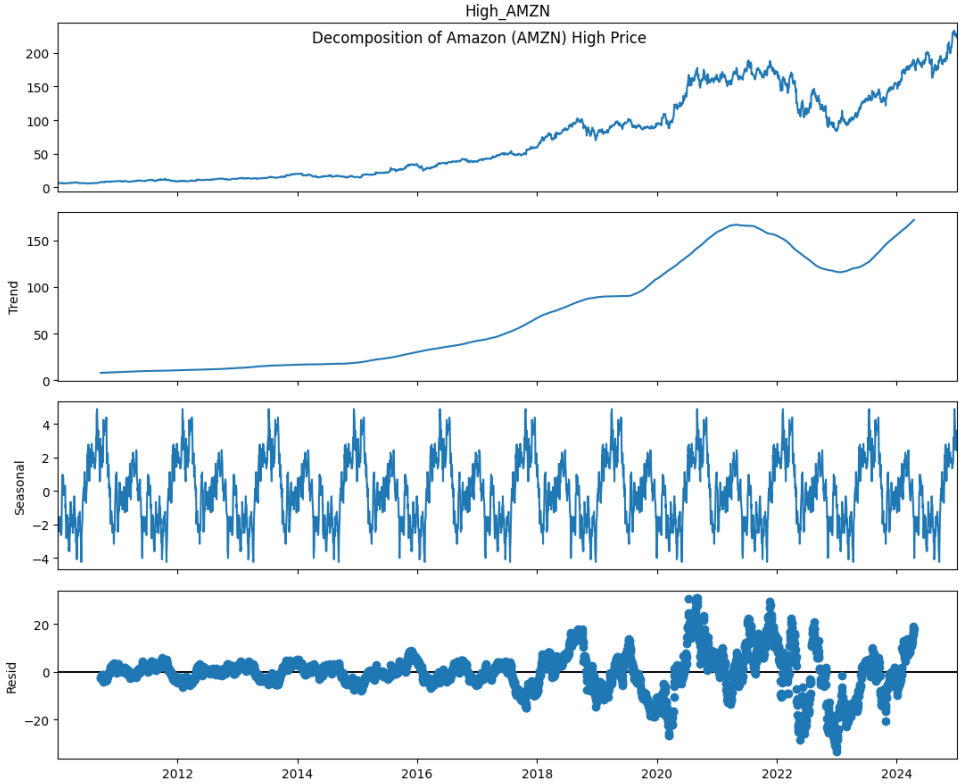

Figure 27: Decomposition of Amazon High Price.

5 Deep Learning Model for Price Forecasting

This section details the construction, training, and evaluation of our predictive model.

5.1 Why Long Short-Term Memory (LSTM)?

Simple Recurrent Neural Networks (RNNs) struggle to remember information over long periods, a phenomenon known as the **vanishing gradient problem**. LSTMs are a specialized type of RNN designed to overcome this limitation. They employ a sophisticated architecture with internal "gates" (an input gate, a forget gate, and an output gate) that regulate the flow of information. This allows the network to selectively remember important patterns from the distant past (like long-term trends) while ignoring irrelevant noise, making them exceptionally powerful for financial time-series forecasting.

5.2 Model Architecture and Training

- **Architecture:** Our model is built in PyTorch and uses a stacked architecture with two LSTM layers followed by two Dense (fully connected) layers. Stacking LSTM layers allows the model to learn patterns at different time scales, creating a more hierarchical understanding of the data.
- **Data Preparation:**
  1. **Scaling:** We use `MinMaxScaler` to scale the 'Close' prices to a range of [-1, 1]. This is a critical step that ensures the gradients during training remain stable and the model converges efficiently.
  2. **Sequencing:** We use a `lookback` window of 20 days. The model is presented with a sequence of 19 consecutive closing prices and is trained to predict the price on the 20th day.
- **Training Process:** The model is trained for 105 epochs.
  - **Loss Function:** We use **Mean Squared Error (MSE)**, which measures the average squared difference between the predicted and actual prices.
  - **Optimizer:** We use **Adam**, an efficient and popular optimization algorithm that adapts the learning rate during training to achieve faster convergence.

5.3 Results: Evaluation and Interpretation

The model's performance is assessed from multiple perspectives:

- **Quantitative Analysis (RMSE):** We calculate the **Root Mean Squared Error** on both the training and testing data. The **Test Score RMSE** is the most important metric, as it indicates how well the model generalizes to new, unseen data. Comparing the Train and Test scores also helps us check for **overfitting** (when the model performs well on training data but poorly on test data).
- **Qualitative Analysis (Visualization):**

– **Diagnostic Plots:** The Seaborn plots provide insight into the training process. The ”Training Prediction” plot shows how well the model learned to fit the data it was trained on, while the ”Training Loss” curve should show a consistent downward trend, indicating that the model was learning successfully during each epoch.

– **Interactive Plotly Chart:** This is the final and most critical output. It visualizes the model's predictions against the actual stock prices for the entire dataset. This chart allows us to visually inspect where the model excels and where it struggles, providing a clear and intuitive measure of its real-world forecasting capability.

5.3.1 NVIDIA (NVDA) Forecast

Figure 28: NVDA Training Fit and Loss Curve.

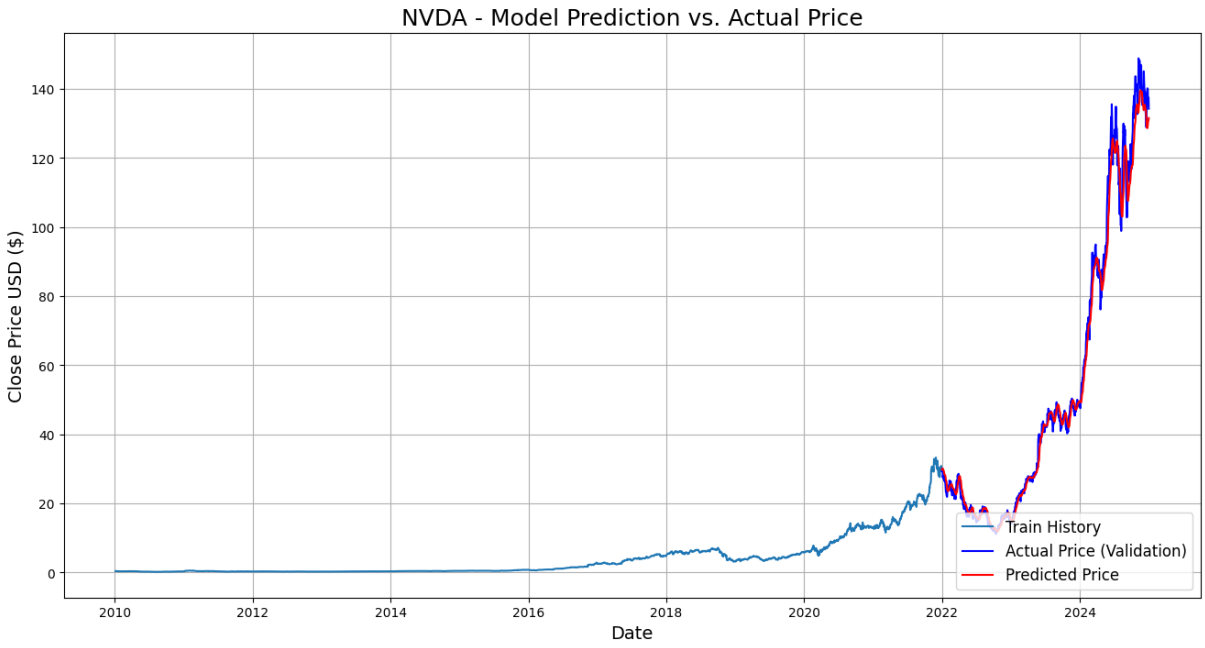

Figure 29: NVDA Model Prediction vs. Actual Price.

5.3.2 Apple (AAPL) Forecast

Figure 30: AAPL Training Fit and Loss Curve.

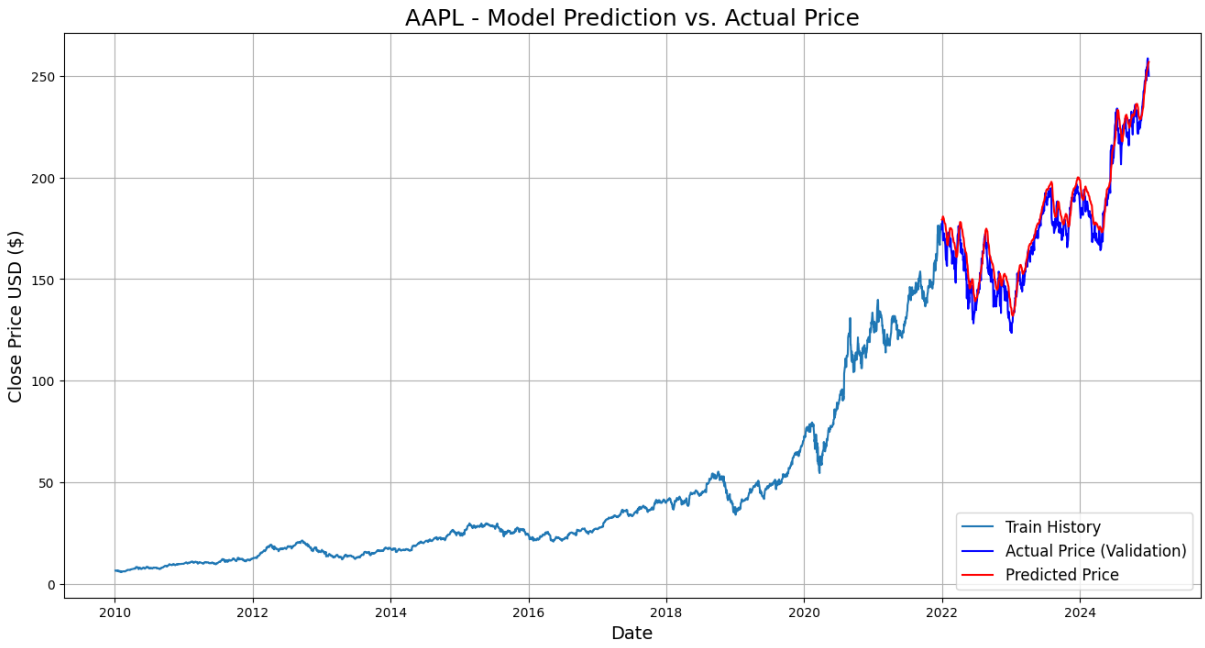

Figure 31: AAPL Model Prediction vs. Actual Price.

5.3.3 Microsoft (MSFT) Forecast

Figure 32: MSFT Training Fit and Loss Curve.

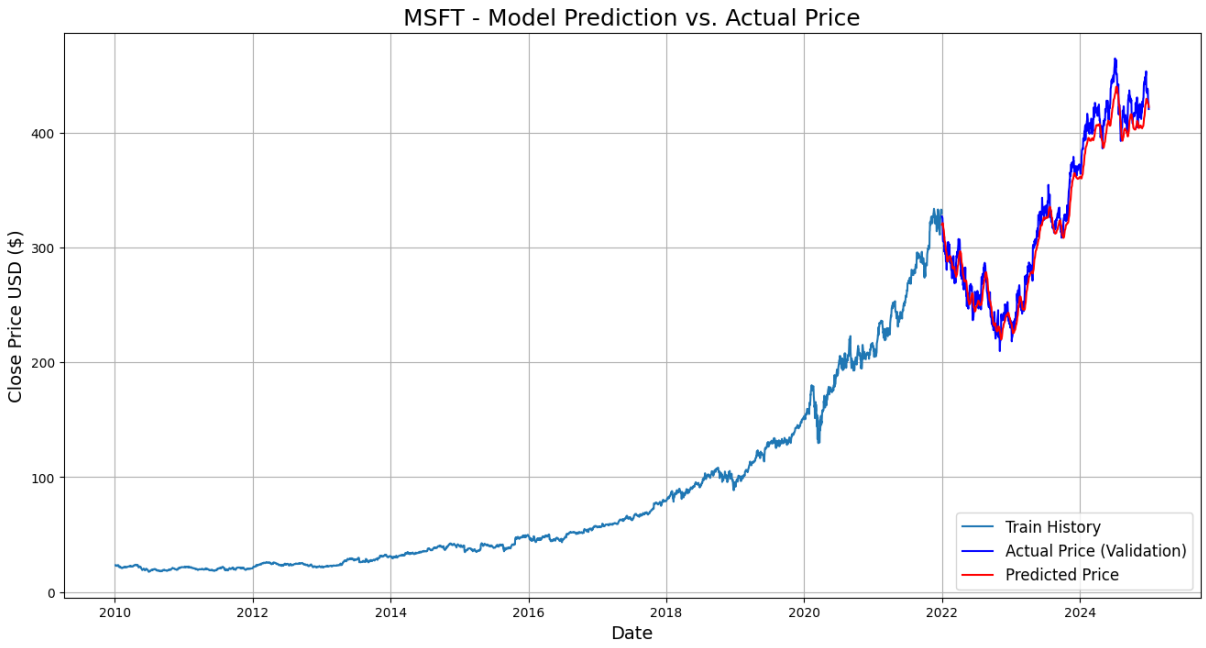

Figure 33: MSFT Model Prediction vs. Actual Price.

5.3.4 Google (GOOGL) Forecast

Figure 34: GOOGL Training Fit and Loss Curve.

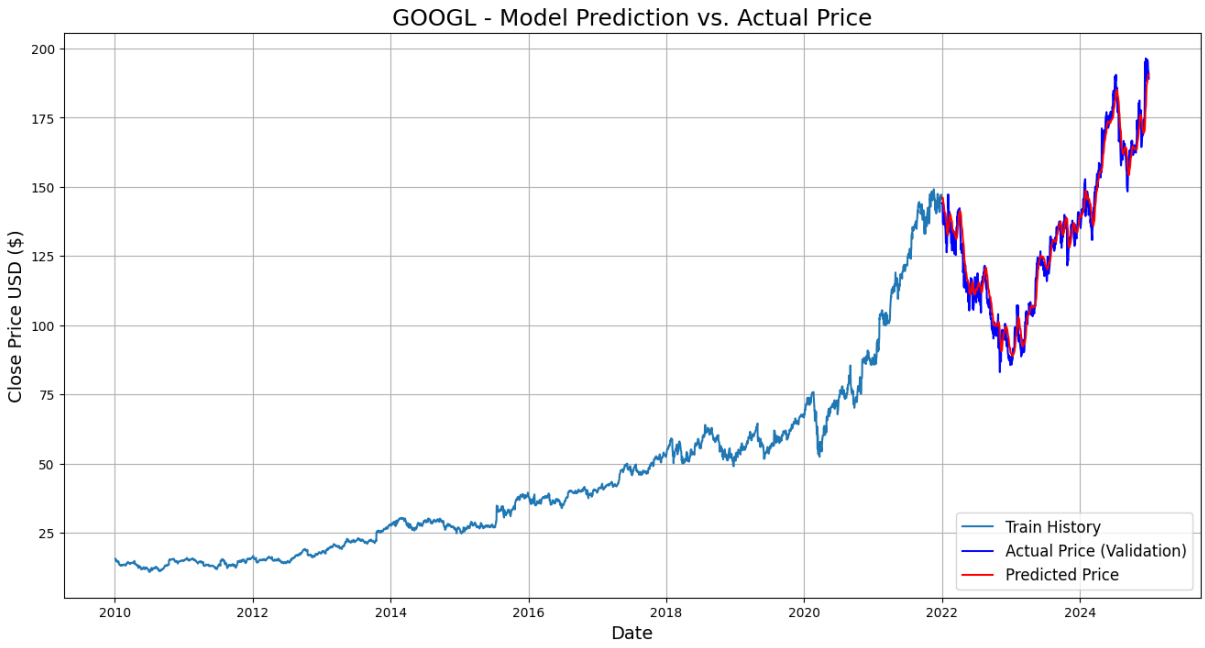

Figure 35: GOOGL Model Prediction vs. Actual Price.

5.3.5 Amazon (AMZN) Forecast

Figure 36: AMZN Training Fit and Loss Curve.

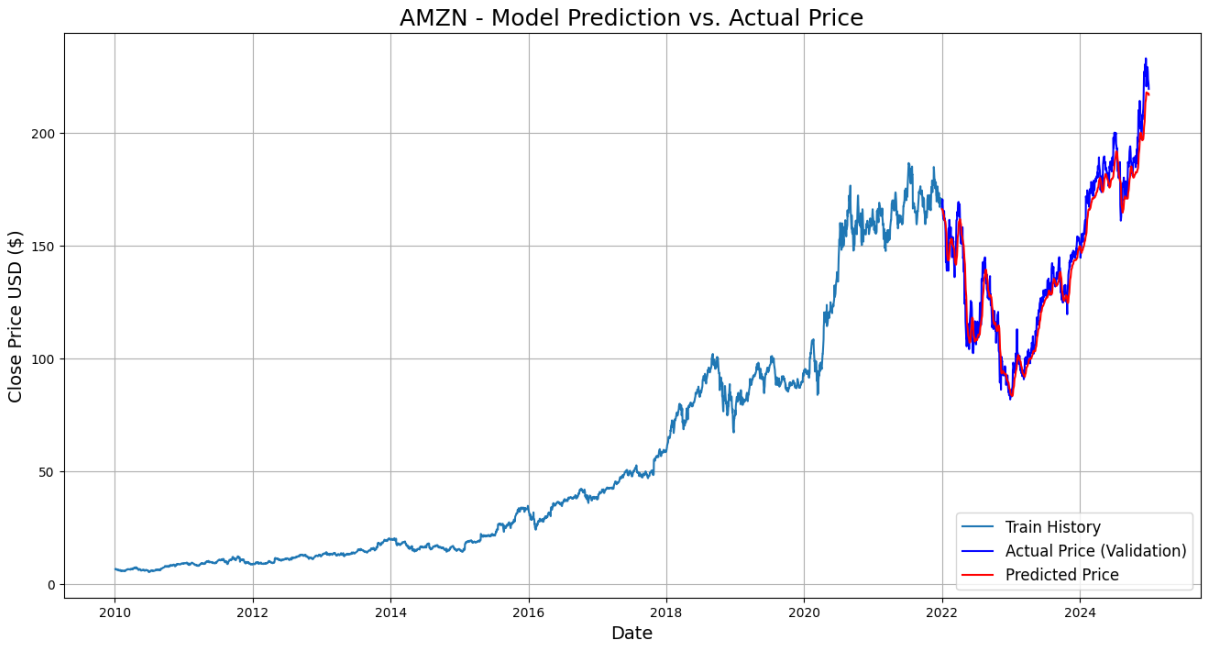

Figure 37: AMZN Model Prediction vs. Actual Price.

6 Conclusion and Future Work

6.1 Summary of Findings

This project successfully implemented a comprehensive workflow for analyzing and forecasting the stock prices of five major technology companies. The **Exploratory Data Analysis** revealed distinct growth trajectories and volatility profiles for each stock. The normalized comparison, for instance, highlighted NVIDIA's exponential growth, particularly in recent years, outperforming its peers in relative terms.

The **Long Short-Term Memory (LSTM) network**, built in PyTorch, proved to be a capable tool for this forecasting task. The model was able to learn and capture the underlying trends from the historical price data, as evidenced by the consistently decreasing loss curve during training and the close alignment of its predictions with the actual price movements in the final visualizations. The quantitative **RMSE scores** provided a concrete measure of the model's prediction error, confirming its effectiveness.

6.2 Model Limitations and Key Takeaways

While the LSTM model performed well, it's crucial to acknowledge its limitations.

- **Reactive, Not Predictive of Events:** The model's predictions are based entirely on historical price patterns. It has no knowledge of external factors and therefore cannot predict sudden price shifts caused by unforeseen events like major news, geopolitical events, or unexpected earnings reports. This is visible in the plots where the prediction line lags during sharp, sudden market changes.
- **Dependence on Past Trends:** The model assumes that future price movements will follow patterns similar to those seen in the past. If a stock's behavior fundamentally changes, the model's accuracy will likely decrease.
- **Not a Financial Advisor:** The output of this model should be treated as a technical analysis tool, not as financial advice. Real-world trading involves a multitude of other factors, including fundamental analysis, risk management, and market sentiment.

6.3 Future Work and Potential Enhancements

This project serves as a strong foundation that can be expanded upon in several ways to create an even more sophisticated forecasting system:

- **Multivariate Analysis:** The current model is **univariate** (it only uses the 'Close' price). A significant improvement would be to make it **multivariate** by including other features like 'Volume', 'Open', 'High', 'Low', and technical indicators (e.g., RSI, MACD, Bollinger Bands).
- **Sentiment Analysis:** Incorporate sentiment analysis by scraping financial news headlines or social media data (like Twitter). The sentiment (positive, negative, neutral) could be quantified and used as an additional input feature, allowing the model to react to market news.
- **Advanced Model Architectures:** Experiment with more complex architectures, such as adding **Attention mechanisms** to the LSTM, which would allow the model to focus on the most important parts of the historical price sequence when making a prediction.

- **Hyperparameter Optimization:** Systematically tune the model's hyperparameters (e.g., `lookback` period, number of layers, hidden dimensions, learning rate) using techniques like Grid Search or Bayesian Optimization to find the optimal configuration.

Ultimately, this project demonstrates the significant potential of deep learning in the financial domain. The LSTM model successfully captured the complex dynamics of stock price movements, providing a powerful analytical tool and a solid framework for future development.