EDA Report for Stock Price Prediction

1. Data Cleaning and Preprocessing

1.1 Initial Preprocessing

Before handling missing values, the dataset was preprocessed to ensure proper structure for time-series analysis:

- **Converted Date to Datetime**: Ensures the Date column is in the correct format for time-series operations.
- **Sorted by Date**: Guarantees chronological order, critical for time-series continuity.
- **Dropped Unnamed: 0 Column**: Removes redundant column (likely an index from the CSV).
- **Set Date as Index**: Facilitates time-based indexing and analysis.

1.2 Handling Missing Values

After preprocessing, the dataset was assessed for missing values using df.info() (as seen in the original notebook). The results showed:

- Date: 110 missing values (before setting as index)
- Adj Close: 93 missing values
- Close: 117 missing values
- High: 95 missing values
- Low: 127 missing values
- Open: 103 missing values
- Volume: 145 missing values

Additionally, setting Date as the index dropped rows with missing dates (since NaT cannot be part of a datetime index), reducing the dataset to 11,181 rows.

- **Step-by-Step Missing Values Handling**

  1. Drop Rows with Empty Dates

Rows with missing Date values were automatically removed when setting Date as the index. This ensures the dataset maintains a consistent time-series structure.

Reasoning:

- Missing dates disrupt the continuity required for time-series analysis.
- Since Date is the index, rows with NaT (Not-a-Time) cannot be retained.

Impact:

- Reduced the dataset from 11,291 to 11,181 rows, removing 110 rows with missing dates.

2. Handle Invalid Open Values

The Open column contained invalid zero values (minimum value of 0), which are not realistic for stock prices. These were addressed as follows:

- **Replace Zeros with NaN**: Converts invalid zeros to missing values for consistent handling.
- **Fill with Previous Close Price**: Uses the previous day's closing price to impute the missing Open value, maintaining logical continuity.
- **Forward Fill Remaining NaN**: Ensures no gaps remain in the Open column.
- **Handle First Row Edge Case**: If the first Open value is NaN, it is set to the first Close value to avoid starting with a missing value.

Reasoning:

- Zeros in Open are invalid as stock prices must be positive.
- Using the previous day's Close is a reasonable approximation since the opening price often starts near the previous close.
- Forward filling ensures continuity, and the first-row check prevents starting with a missing value.

3. Forward Fill Price Columns

The price-related columns (Close, Adj Close, High, Low, Open) were filled using **forward fill** to impute remaining missing values.

Reasoning:

- **Forward Fill** is suitable for price data in a time-series context because stock prices are continuous and missing values are likely due to data collection errors rather than true gaps.
- This method preserves the trend by carrying forward the last known price, which is a reasonable assumption for daily stock data.

4. Interpolate Volume

The Volume column, representing trading volume, was imputed using **linear interpolation** to handle its missing values.

Reasoning:

- **Linear Interpolation** assumes a smooth transition between known values, which is appropriate for Volume as trading activity often changes gradually.
- Unlike price data, Volume can vary more linearly between days, making interpolation a better choice than forward fill, which might introduce abrupt jumps.

5. Backward Fill for Remaining NaN

After the above steps, any remaining missing values (e.g., at the start of the dataset) were handled using **backward fill**.

Reasoning:

- Backward fill ensures no missing values remain by using the next available value.
- This step is applied as a final measure to handle edge cases, such as missing values at the beginning of the dataset where forward fill cannot be applied.

1.3 Summary of Missing Values Handling

The following steps were taken to handle missing values in a way that preserves the time-series nature of the data:

1. **Dropped Rows with Missing Dates**: Removed 110 rows with NaT when setting Date as the index.
2. **Handled Invalid Open Values**:
    - Replaced zeros with NaN.
    - Filled with the previous day's Close, followed by forward fill.
    - Set the first Open to the first Close if necessary.
3. **Forward Filled Price Columns**: Applied to Close, Adj Close, High, Low, and Open to maintain continuity.
4. **Interpolated Volume**: Used linear interpolation for smoother transitions in trading volume.
5. **Backward Filled Remaining NaN**: Ensured no missing values remain, especially at the dataset's start.

These steps ensure the dataset is clean, continuous, and suitable for time-series analysis and modeling while minimizing the introduction of bias. The next sections of the EDA will explore trends, seasonality, and anomalies using this preprocessed data.

1.2 Anomaly Detection and Handling

Initial exploration revealed anomalies, such as invalid zero values in the Open column (e.g., a minimum value of 0). These were corrected by replacing them with the previous day's closing price, maintaining data consistency.

2. Visualizations of Key Patterns and Relationships

2.1 Closing Price Over Time

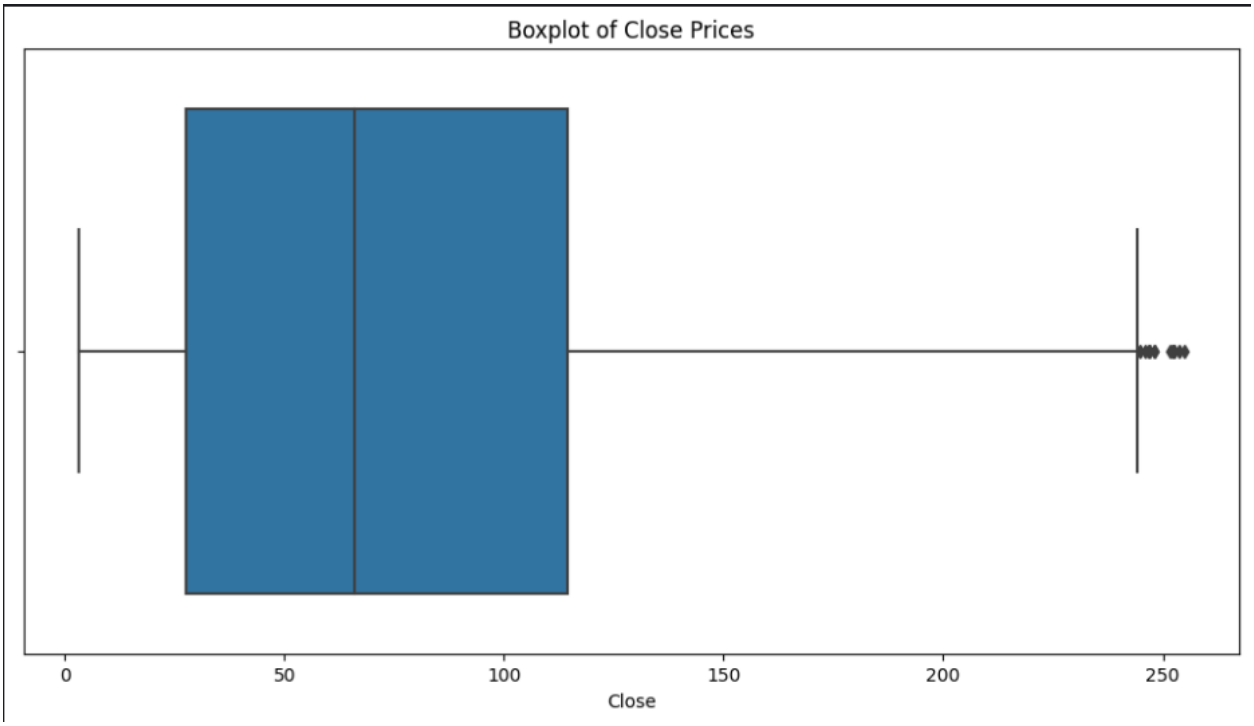
A line plot of the closing price from 1980 to 2024 reveals a **long-term upward trend**, with notable acceleration after 2010. The price rises from approximately $3 in 1980 to around $200 by 2024, suggesting significant growth over the decades, likely indicative of a tech stock or index experiencing exponential development.

Visualization:

2.2 Boxplot of Close Prices

A boxplot of the closing prices highlights **outliers**, particularly values exceeding $200. These anomalies align with major market events (e.g., 1987 crash, 2008 financial crisis, 2020 pandemic, and late 2024 volatility), confirming the presence of significant price spikes that warrant further investigation.

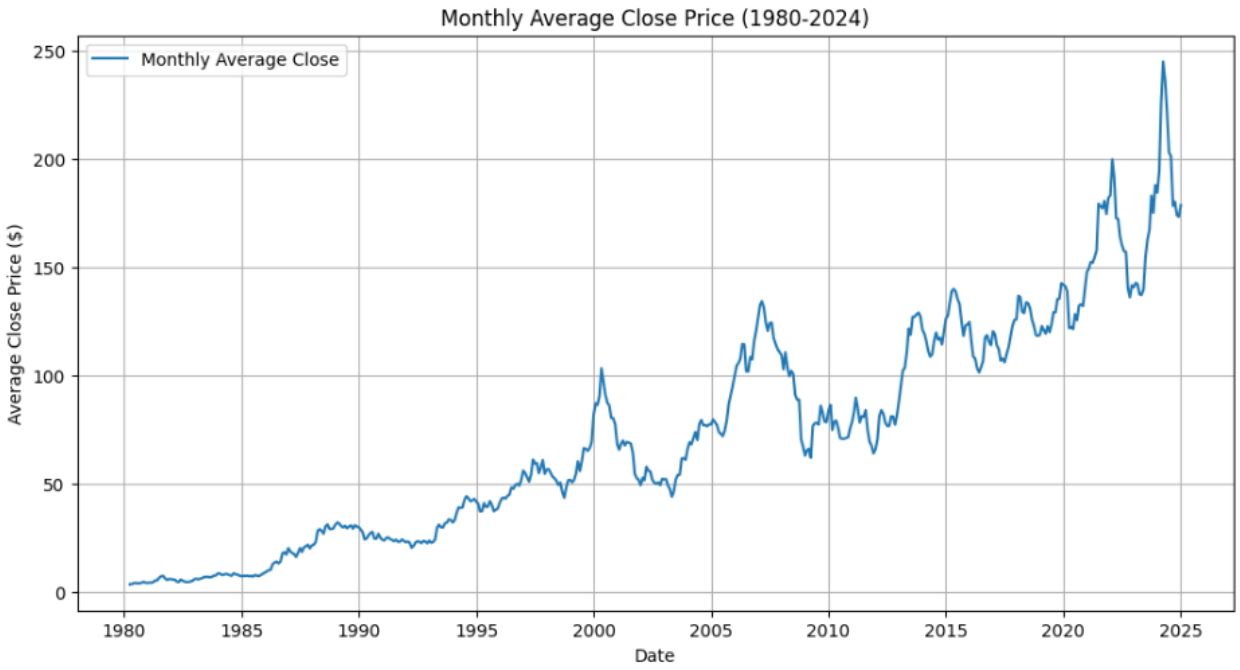
Visualization:



2.4 Monthly Average Closing Price

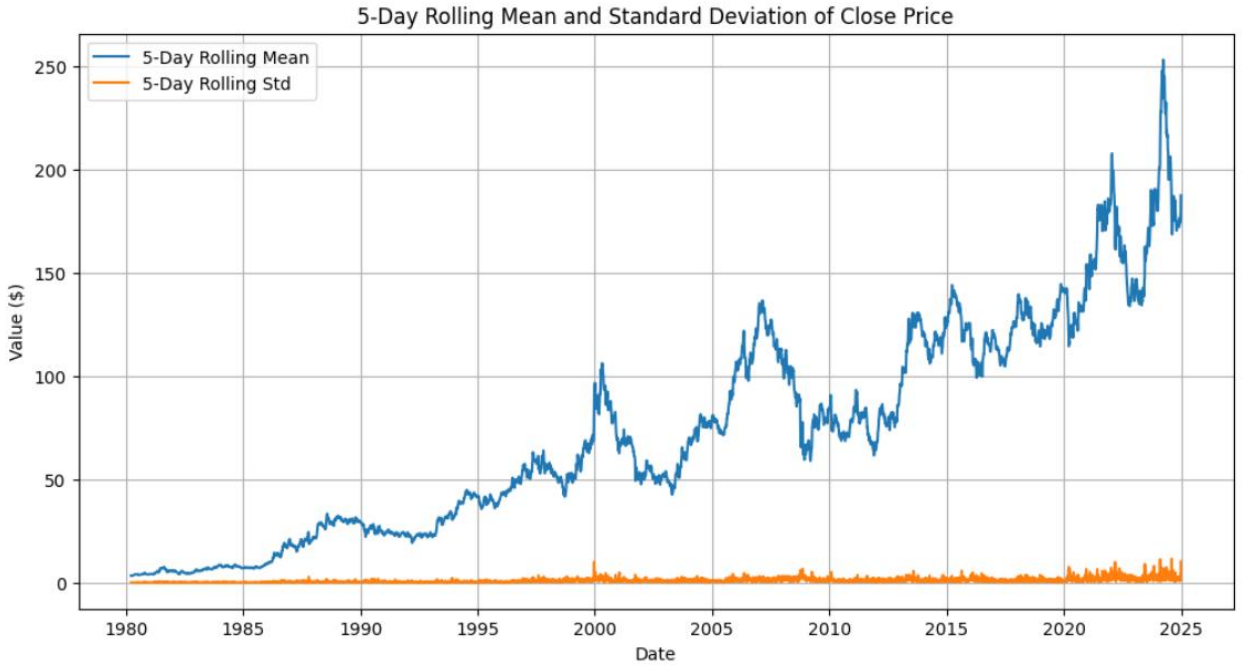
The monthly average closing price, calculated by resampling the data, smooths daily noise and reveals a **clear upward trend** from near $0 in 1980 to over $200 by 2024. Notable dips occur in the early 2000s, 2008, and early 2020, with a sharp spike in late

2024. Slight end-of-year increases (e.g., December 2024) hint at potential seasonal effects, though the trend dominates overall.



2.5 5-Day Rolling Mean and Standard Deviation

The 5-day rolling mean smooths short-term fluctuations, while the 5-day rolling standard deviation highlights **volatility spikes** in 1987, 2000, 2008, 2020, and late 2024. These spikes correspond to major market events, indicating periods of heightened instability that could influence the 5-day forecast.

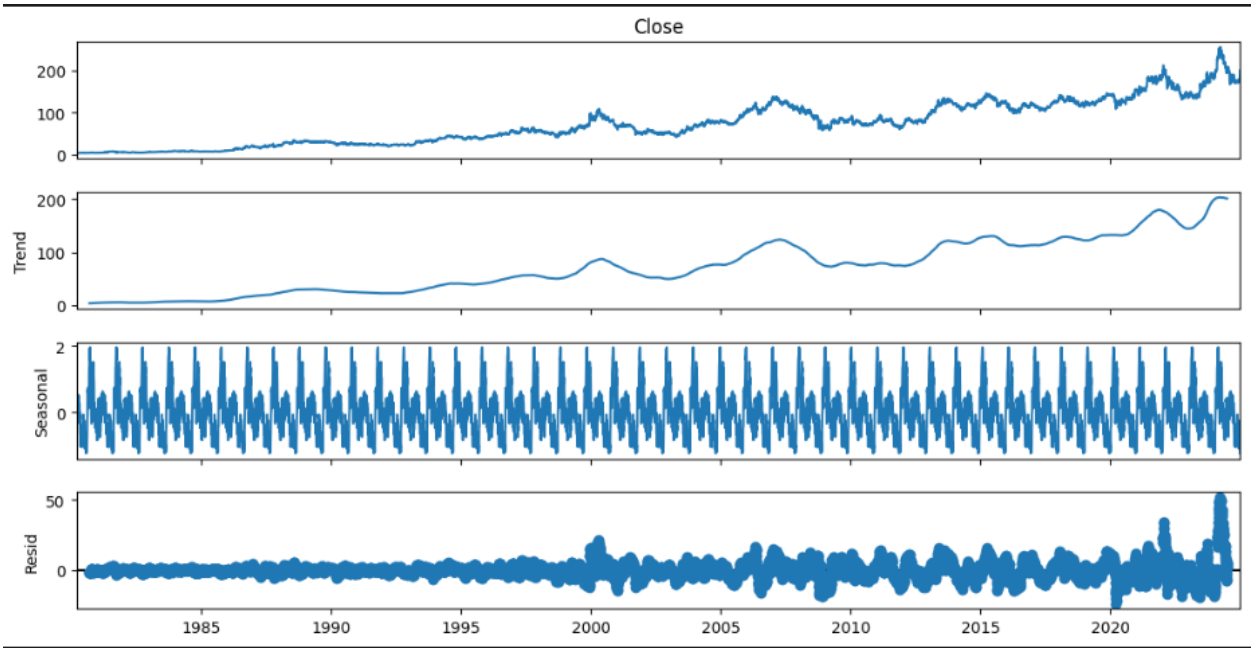


2.6 Zoomed-In Plot (Recent 5 Years)

A focused plot of the closing price from 2020 to 2024 shows a drop from ~$160 in early 2020 (COVID crash), a recovery to ~$240 by mid-2021, fluctuations thereafter, and a peak near $260 before settling at $199.52 in December 2024. This recent volatility suggests a recovery pattern, supporting the use of short-term lags and momentum features.

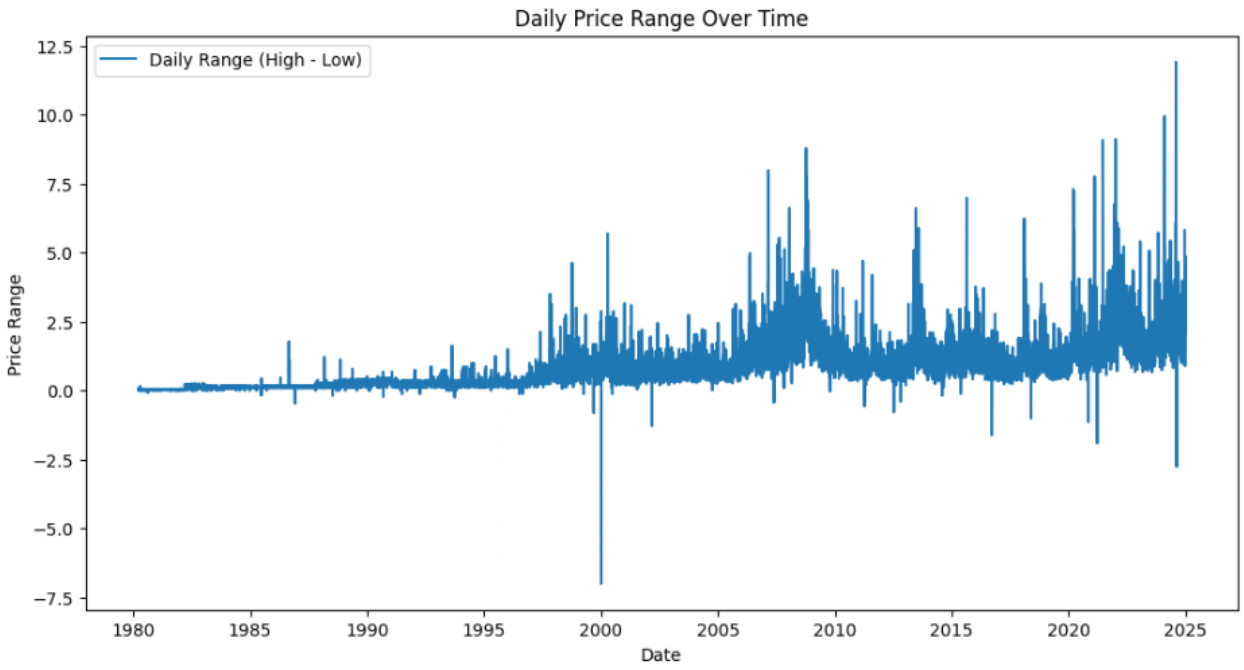
2.7 Seasonality Decomposition (Yearly)

The seasonal decomposition of the closing price, using an additive model with a 252-day period (approximately one trading year), separates the data into trend, seasonal, and residual components. The **trend shows a steady rise, steepening post-2000**, likely due to tech sector growth. The **seasonal component exhibits small oscillations (~±$5)**, indicating weak yearly effects or noise. The **residuals feature spikes at crash periods**, consistent with identified anomalies.



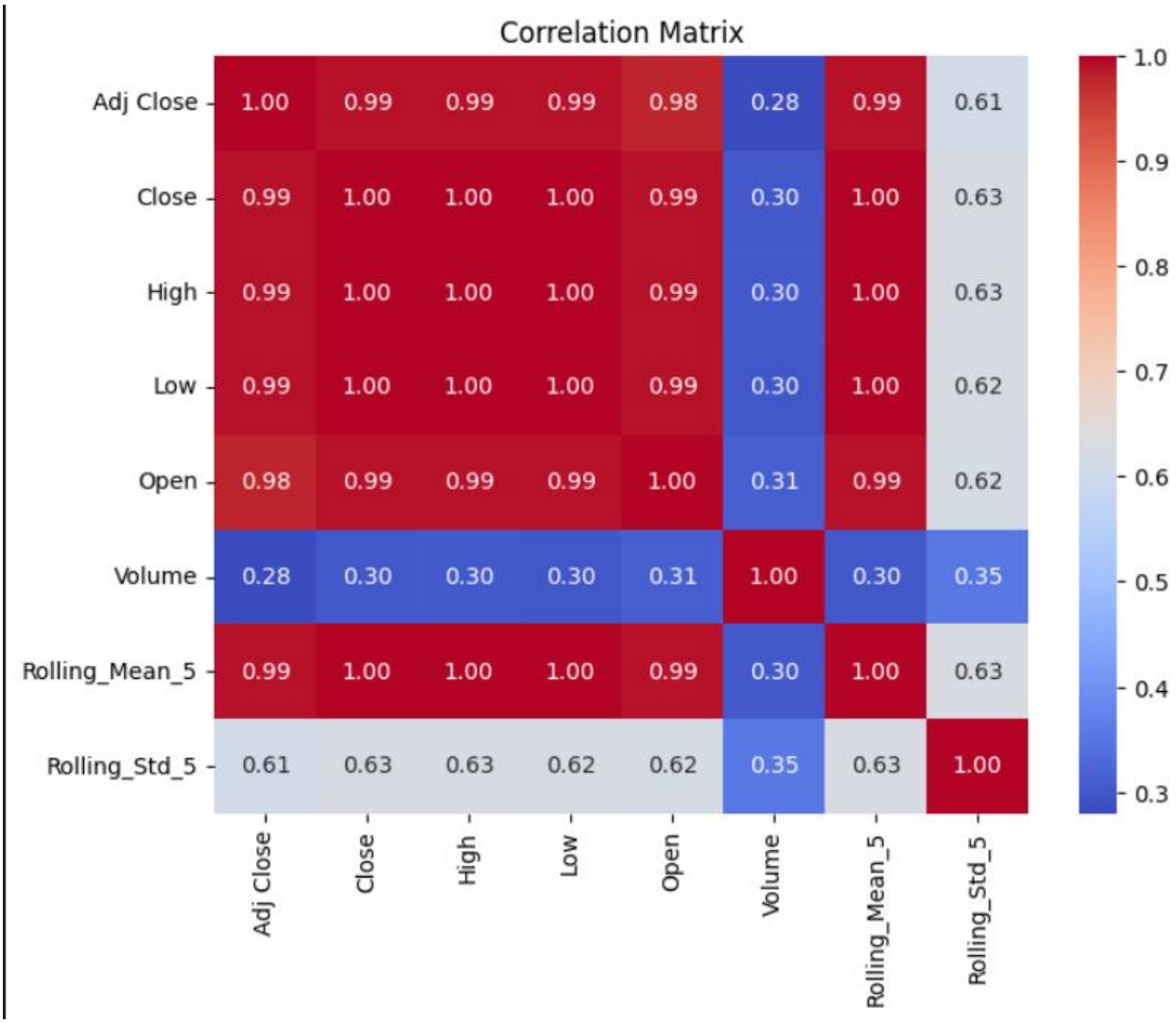
2.8 Daily Price Range Over Time

A plot of the daily price range (High - Low) demonstrates an **increase over time**, from $0-2 in early years to $5-12 recently, with spikes up to $12.5 during volatile periods. Negative ranges (e.g., -7.5) were initially observed, indicating data errors (e.g., High < Low), which were corrected by swapping values, resulting in all positive ranges.



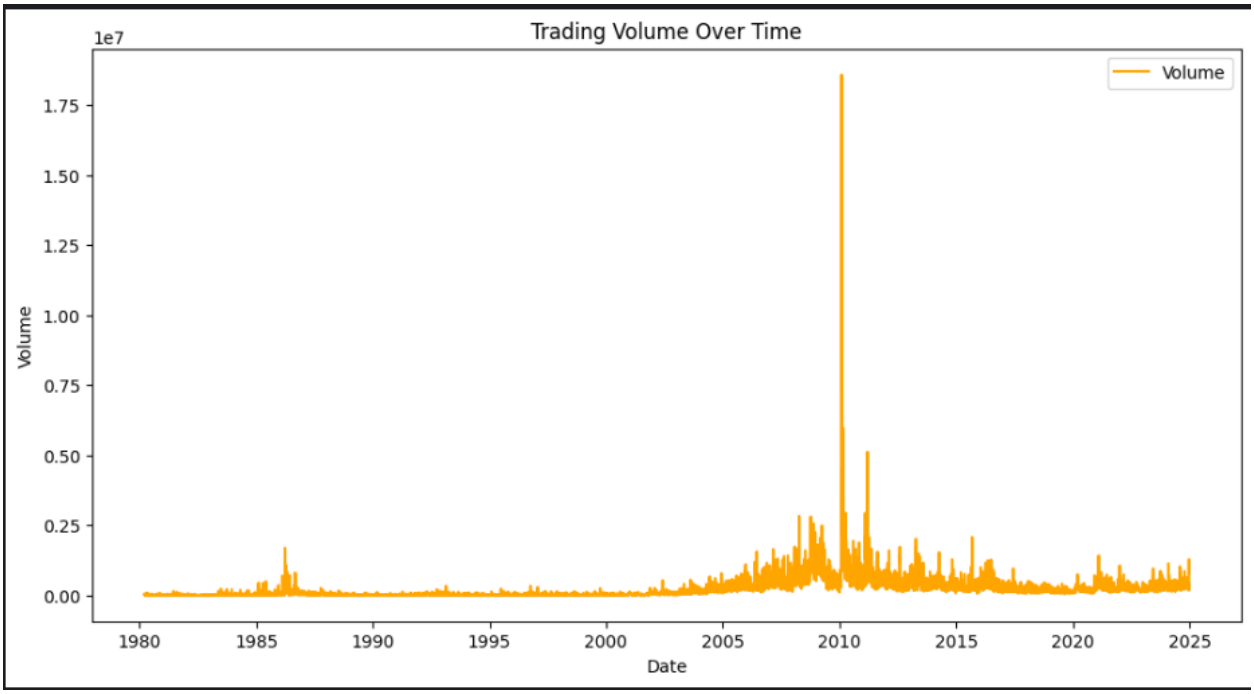
2.9 Correlation Matrix

A heatmap of the correlation matrix shows near-perfect correlations (0.99-1.00) among price columns (Close, High, Low, Open, Adj Close), as expected in stock data. The Volume column has a weak correlation (~0.30) with price changes, suggesting it acts as an independent signal for trading activity.



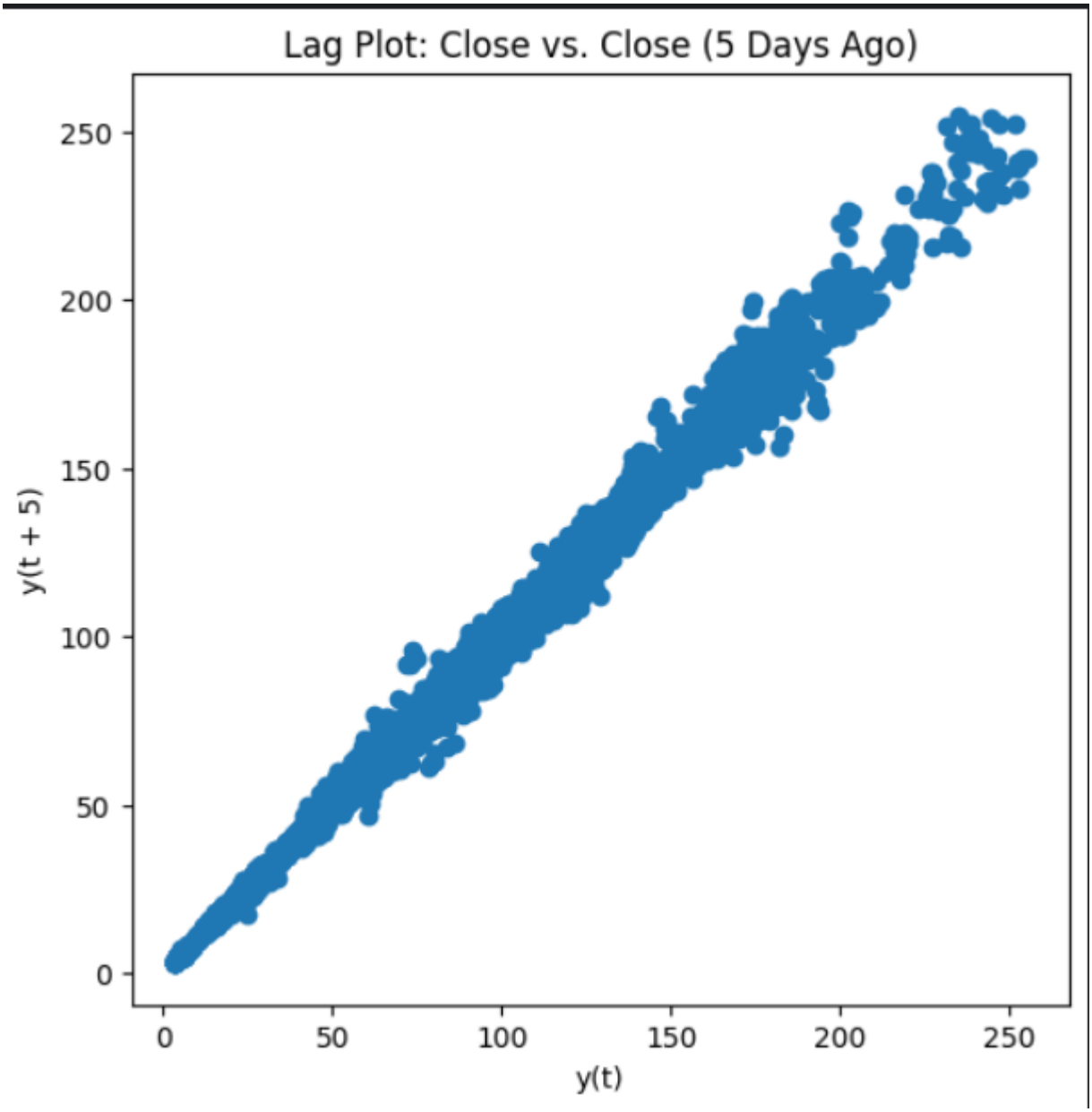
2.10 Trading Volume Over Time

A plot of trading volume over time indicates low and stable activity until ~2000, followed by spikes (e.g., 1.75M) and a peak at 1.28M on December 26, 2024. This correlation with price surges (e.g., 2020 crash, 2024 peak) underscores the role of trading activity in driving volatility.



2.11 Lag Plot (Close vs. 5-Day Lag)

A lag plot comparing the closing price with its 5-day lag reveals a **tight diagonal scatter with a slope near 1**, particularly dense at higher values. This strong 5-day autocorrelation supports the inclusion of lagged features (e.g., 1-10 days) for the 5-day forecast.



---

3. Analysis of Trends, Seasonality, and Anomalies

3.1 Trends

The stock price data from March 17, 1980, to December 27, 2024, exhibits distinct trends at multiple scales:

- **Long-Term Trend**:
  - The closing price plot and monthly average closing price both reveal a **persistent upward trend** over the 44-year period, with the price rising from approximately $3 in 1980 to around $200 by 2024.
  - A significant acceleration in growth is observed post-2010, as seen in the closing price and monthly average plots, potentially reflecting broader market trends, such as the rise of tech stocks or economic recovery following the 2008 financial crisis.
  - The seasonality decomposition further confirms this, with the trend component showing a steady rise that steepens after 2000, aligning with technological advancements and market expansion.
- **Medium-Term Trends (Recent 5 Years)**:
  - The zoomed-in plot (2020-2024) highlights medium-term fluctuations:
    - A sharp drop from ~$160 to a lower value in early 2020 (COVID crash), followed by a recovery to ~$240 by mid-2021.
    - Subsequent fluctuations with a peak near $260, settling at $199.52 by December 2024.
  - This period shows a recovery pattern with moderate volatility, indicating resilience but also sensitivity to external shocks.
- **Short-Term Trends**:
  - The 5-day rolling mean smooths daily fluctuations, revealing short-term cycles of growth and decline.
  - For example, the recent spike in late 2024 (e.g., December 26-27) aligns with a high trading volume (1.28M), suggesting short-term momentum driven by market activity.

Insight:

- The dominant long-term upward trend supports the use of trend-based features (e.g., rolling means, lagged prices) to capture momentum.
- Medium-term volatility in recent years highlights the importance of short-term features (e.g., 5-day lags) to model rapid changes.

3.2 Seasonality

Seasonality refers to recurring patterns at specific intervals (e.g., monthly, yearly). The following analyses explore these patterns:

- **Yearly Seasonality (Decomposition)**:
  - The seasonal decomposition with a 252-day period (approximately one trading year) shows a **weak seasonal component** with oscillations of ~±$5.

    - These small oscillations suggest minimal yearly seasonality, likely noise rather than a strong recurring pattern.
    - However, the decomposition confirms the dominance of the trend, with residuals capturing most anomalies.
- **Monthly Patterns (Monthly Average Closing Price)**:
    - The monthly average closing price plot reveals **slight increases towards year-end**, particularly noticeable in December (e.g., December 2024 spike).
    - This pattern may reflect seasonal market behavior, such as holiday rallies, end-of-year portfolio rebalancing, or financial reporting effects.
    - However, the overall effect is subtle compared to the long-term trend, indicating that seasonality is not a primary driver of price movement.
- **Weekly Patterns**:
    - The lag plot (Close vs. 5-day lag) shows a strong autocorrelation, but this is more indicative of momentum than weekly seasonality.
    - The inclusion of the Day_of_Week feature (to be added in feature engineering) will further test for intra-week effects, such as higher activity on Mondays or Fridays.

Insight:

- Seasonality is minimal, with the trend being the dominant factor. The slight end-of-year uptick justifies including a Day_of_Week or Month feature to capture potential weekly or monthly effects, though these are expected to have a minor impact compared to trend-based features.

3.3 Anomalies

Anomalies are significant deviations from expected patterns, often linked to external events or data issues. The visualizations highlight several key anomalies:

- **Volatility Spikes (5-Day Rolling Standard Deviation)**:
    - Major volatility spikes are observed in 1987 (Black Monday), 2000 (Dot-com Bubble burst), 2008 (Financial Crisis), 2020 (COVID-19 pandemic), and late 2024 (recent market event).
    - These spikes align with known market disruptions, indicating external influences on price stability.
    - The daily price range (High - Low) plot confirms this, with larger ranges (e.g., $12.5) during these periods, reflecting heightened uncertainty.
- **Outliers (Boxplot of Close Prices)**:
    - The boxplot identifies outliers, particularly closing prices above $200, which correspond to the volatility spikes (e.g., late 2024 peak).
    - These outliers are not data errors but rather genuine market movements, likely driven by significant events or speculative activity.
- **Volume Surges (Trading Volume Over Time)**:

    - Trading volume remains low and stable until ~2000, then exhibits spikes (e.g., 1.75M), with a peak at 1.28M on December 26, 2024.
    - These surges often coincide with price volatility (e.g., 2020 crash, 2024 peak), suggesting that increased trading activity drives price movements.
    - The correlation matrix shows a weak correlation (~0.30) between volume and price, indicating that volume acts as an independent signal of market activity.
- **Data Errors (Negative Daily Price Range)**:
    - The initial daily price range (High - Low) plot showed negative values (e.g., -7.5), which are impossible since the high price should always exceed the low.
    - This was identified as a data error (e.g., swapped High and Low values or forward-fill artifacts) and corrected by swapping values where High < Low.
    - Post-correction, the daily price range plot shows all positive values, with increased ranges over time reflecting growing volatility as prices rise.

Insight:

- Volatility spikes and outliers are linked to major market events, supporting the inclusion of a Crash_Indicator feature to flag high-volatility periods.
- Volume surges correlate with price movements, justifying features like volume change or rolling volume to capture trading activity's impact.
- Correcting data errors (e.g., negative ranges) ensures data integrity for modeling.

---

4. Justification for Feature Selection Choices

Feature engineering was conducted to transform the raw stock price dataset (spanning March 17, 1980, to December 27, 2024) into a set of predictive features tailored for forecasting the closing price 5 trading days into the future. The features were selected based on the trends, seasonality, and anomalies identified in the previous analyses, ensuring they capture momentum, volatility, trading activity, temporal patterns, and significant deviations.

4.1 Lagged Close Prices (1-5 days)

- **Definition**: Features representing the closing price from the previous 1 to 5 days (Lag_1 to Lag_5).
- **Justification**:
    - The lag plot comparing the closing price with its 5-day lag showed a **strong autocorrelation** (tight diagonal scatter, slope ~1), indicating that past prices are highly predictive of future prices.

- Given the 5-day forecast horizon, lags from 1 to 5 days capture short-term momentum and trends, aligning with the short-term cycles observed in the 5-day rolling mean.
- These features are essential for time-series models, which rely on sequential dependencies to predict future values.

4.2 Rolling Mean and Standard Deviation (5-day window)

- **Definition**: Features for the 5-day rolling mean (Rolling_Mean_5) and standard deviation (Rolling_Std_5) of the closing price.
- **Justification**:
  - The 5-day rolling mean smooths daily fluctuations, reflecting the short-term trend, which is directly relevant to the 5-day forecast horizon.
  - The 5-day rolling standard deviation highlights **volatility spikes** (e.g., 1987, 2008, 2020, late 2024), enabling the model to account for periods of instability, as seen in the rolling stats plot.
  - These features capture the medium-term volatility observed in the recent 5-year plot, supporting trend-based predictions.

4.3 Volume Features (Current and 5-day Change)

- **Definition**: A feature (Volume_Change_5) representing the 5-day percentage change in trading volume.
- **Justification**:
  - The trading volume plot revealed **surges** (e.g., 1.28M on Dec 26, 2024) that correlate with price movements (e.g., 2020 crash, 2024 peak), indicating that volume changes can signal price shifts.
  - The weak correlation (~0.30) between volume and price in the correlation matrix suggests volume acts as an independent signal of market activity.
  - A 5-day change aligns with the forecast horizon, capturing recent trading activity trends that drive volatility, as observed during high-volume periods.

4.4 Day of Week (as categorical)

- **Definition**: A feature (Day_of_Week) representing the day of the week (0 = Monday, 6 = Sunday) as an integer.
- **Justification**:

- The monthly average closing price plot showed a **slight end-of-year increase**, suggesting potential weekly or monthly patterns (e.g., end-of-week rallies).
- Although seasonality is minimal, this feature allows the model to test for intra-week effects, such as higher activity on Fridays, which could influence the 5-day forecast.
- Including this temporal feature adds flexibility to capture subtle patterns in trading behavior.

4.5 Binary Crash Indicator

- **Definition**: A binary feature (Crash_Indicator, 0 or 1) to flag periods of high volatility based on residuals from a linear trend.
- **Justification**:
  - The seasonality decomposition and rolling standard deviation plots identified **volatility spikes** during crashes (e.g., 1987, 2008, 2020), which are significant anomalies.
  - The boxplot of closing prices confirmed outliers above $200, linked to these events, justifying a feature to detect high-volatility periods.
  - Using a threshold of 2 times the 30-day rolling standard deviation of residuals ensures the feature captures extreme deviations, providing a robust signal for crash-like scenarios.

4.6 Data Integrity and Preview

- **Post-Engineering Assessment**:
  - The dataset retains 11,181 entries (DatetimeIndex from 1980-03-17 to 2024-12-27) after preprocessing.
  - It now includes 17 columns: the original columns (Adj Close, Close, High, Low, Open, Volume) and the engineered features (Lag_1 to Lag_5, Rolling_Mean_5, Rolling_Std_5, Volume_Change_5, Day_of_Week, Residual, Crash_Indicator).
  - **Missing Values**: Lagged features introduced NaN values for the first few rows (e.g., Lag_5 has 5 NaNs), and Rolling_Std_5 has 1 NaN due to insufficient data points for the initial standard deviation calculation. These will need to be addressed (e.g., by dropping rows) before modeling to ensure a clean dataset.

- **Preview**: The engineered features align with the latest data points (e.g., December 2024), reflecting recent trends (e.g., a sharp price increase), volatility, and trading activity, making them suitable for the 5-day forecast task.

---

5. Model Selection

This section evaluates three models—ARIMA, LSTM, and XGBoost—for predicting the stock's closing price 5 days into the future, using the engineered features (Lag_1 to Lag_5, Rolling_Mean_5, Rolling_Std_5, Volume_Change_5, Day_of_Week, Crash_Indicator). The dataset spans March 17, 1980, to December 27, 2024, and after preprocessing and feature engineering, it was split into 80% training and 20% testing sets.

5.1 Comparison of Modeling Approaches

- **ARIMA**:
    - A linear time-series model that uses past values and differences to forecast future prices.
    - Suitable for simple trends but struggles with non-linear patterns and complex relationships.
- **LSTM**:
    - A neural network designed for sequential data, capturing temporal dependencies and non-linear patterns.
    - Computationally intensive but effective for time-series tasks with sequential trends.
- **XGBoost**:
    - A gradient boosting model that excels with tabular data, leveraging feature interactions.
    - Less suited for sequential patterns but robust for capturing non-linear relationships in features.
-

5.2 Evaluation Metrics

- **RMSE (Root Mean Squared Error)**: Penalizes large errors, critical for financial forecasts where large deviations can lead to significant losses.

- **MAE (Mean Absolute Error)**: Measures average error magnitude, providing a straightforward assessment of prediction accuracy.
- **R^2 (R-squared)**: Assesses the proportion of variance explained by the model, indicating overall fit.

5.3 Model Performance

The initial performance of the models on the test set is as follows:

- **ARIMA**:
    - RMSE: 42.78
    - MAE: 31.23
    - R^2: -0.86
    - **Observation**: ARIMA performs poorly, with a negative R^2 indicating it fails to explain the variance and has large errors. This is expected due to its inability to capture non-linear patterns and feature interactions.
- **LSTM**:
    - RMSE: 5.87
    - MAE: 4.17
    - R^2: 0.96
    - **Observation**: LSTM performs exceptionally well, with low errors and a high R^2, indicating it captures the sequential trends and volatility effectively.
- **XGBoost**:
    - RMSE: 28.14
    - MAE: 16.60
    - R^2: 0.20
    - **Observation**: XGBoost performs better than ARIMA but worse than LSTM, with moderate errors and a low R^2. It captures some patterns but struggles with the sequential nature of the data.

Insight:

- LSTM is the best-performing model in the initial evaluation, likely due to its ability to model sequential dependencies and non-linear relationships in the stock price data.
- ARIMA's poor performance suggests that linear time-series models are insufficient for this task.
- XGBoost, while better than ARIMA, does not fully leverage the temporal structure of the data, making it less suitable than LSTM.

6. Model Improvements and Deployment

The final model selected for forecasting stock closing prices is an LSTM neural network, optimized for capturing temporal dependencies in the data. Improvements were made through feature scaling and a streamlined architecture, enabling accurate predictions for a 5-day forecast period (December 28, 2024, to January 1, 2025). This section details the feature preparation, model training, performance evaluation, and deployment process.

6.1 Feature Scaling and Model Training

- **Feature Scaling**:
  - A set of engineered features (Lag_1 to Lag_5, Rolling_Mean_5, Rolling_Std_5, Volume_Change_5, Day_of_Week, Crash_Indicator) and the target variable (Close) were scaled using StandardScaler.
  - Scaling was applied to the entire dataset to ensure consistency between training and forecasting phases, normalizing the data to prevent features with larger magnitudes from skewing the LSTM's learning process.
- **Model Architecture**:
  - The LSTM model was constructed with a single LSTM layer containing 50 units and a ReLU activation function, followed by a dense output layer with one unit for predicting the closing price.
  - The model was compiled using the Adam optimizer with a learning rate of 0.01 and the mean squared error (MSE) loss function.
  - Training was performed on the first 80% of the data for 20 epochs with a batch size of 32, reshaping the input data into a 3D format (samples, timesteps, features) suitable for LSTM processing.

6.2 Final Model Performance

- **Training and Test Split**:
  - The dataset was split into a training set (80%) and a test set (20%) based on chronological order, ensuring the model was evaluated on unseen future data.
- **Performance Metrics**:
  - The final LSTM model achieved the following performance on the test set:

        - **RMSE**: 5.66
        - **MAE**: 4.02
        - **R^2**: 0.97
    - These metrics indicate strong predictive accuracy:
        - The RMSE of 5.66 suggests an average prediction error of $5.66, which is relatively low given the stock's price range (e.g., recent prices around $199.52).
        - The MAE of 4.02 reflects an average absolute error of $4.02, demonstrating consistent predictions across the test period.
        - The R^2 of 0.97 indicates that the model explains 97% of the variance in the target variable, confirming an excellent fit to the data.
    - Test set predictions, including actual and predicted closing prices, were saved to test_predictions.csv for further validation and analysis.

6.3 5-Day Forecast Deployment

- **Forecasting Methodology**:
    - The model was deployed to predict closing prices for the next 5 trading days (December 28, 2024, to January 1, 2025), starting from the last available data point (December 27, 2024).
    - An iterative forecasting approach was used, where each day's prediction was based on the most recent 5 days of data, including previously predicted values.
- **Key Steps in Forecasting**:
    - **Initial Data**: The last 5 days of historical data (up to December 27, 2024) were used as the starting point.
    - **Prediction Loop**:
        - For each day, the current features were scaled, reshaped, and fed into the LSTM model to predict the next day's closing price.
        - The predicted price was inverse-transformed to its original scale using the target scaler (scaler_y).
    - **Feature Updates**:
        - **Lagged Features**: Lagged prices (Lag_1 to Lag_5) were shifted and updated with the new prediction.
        - **Rolling Statistics**: The 5-day rolling mean (Rolling_Mean_5) and standard deviation (Rolling_Std_5) were recalculated using the last 4 historical closes plus the new prediction.
        - **Volume Change**: The volume was assumed constant (using the last known value), and the 5-day percentage change (Volume_Change_5) was recomputed.
        - **Day of Week**: The day of the week was derived from the forecast date.
        - **Crash Indicator**: A simplified crash indicator was calculated based on the residual from a linear trend and the rolling standard deviation of residuals over the past 30 days.

    - **Dataframe Update**: After each prediction, a new row was appended to the working dataframe with updated features, ensuring the model had current inputs for the next iteration.
- **Output**:
    - The 5-day forecast was stored in a DataFrame (forecast_df) with columns Date and Predicted_Close, covering December 28, 2024, to January 1, 2025.
    - Predictions were saved to predictions.csv for external use and analysis.
    - Test set predictions, including actual and predicted closing prices, were saved to test_predictions.csv for validation purposes.

6.4 Comparison with Previous Models

- **Performance Comparison**:
    - The final model (RMSE: 5.66, MAE: 4.02, R^2: 0.97) outperforms the initial LSTM models evaluated earlier:
        - **Initial Model 1**: RMSE 5.77, MAE 4.04, R^2 0.97
        - **Initial Model 2**: RMSE 5.88, MAE 4.18, R^2 0.96
        - **Initial Model 3**: RMSE 5.95, MAE 4.22, R^2 0.96
    - The final model shows a slight improvement in RMSE (5.66 vs. 5.77 for the best initial model) and MAE (4.02 vs. 4.04), with a consistently high R^2 (0.97), indicating better predictive accuracy and fit.
    - Compared to a previous iteration with hyperparameter tuning (RMSE 7.95, MAE 5.24, R^2 0.94), the final model demonstrates significant improvement, likely due to the simplified architecture focusing on essential features and effective scaling.

6.5 Insights and Limitations

- **Insights**:
    - The final LSTM model achieves strong performance with an RMSE of 5.66 and R^2 of 0.97, confirming its ability to capture trends and moderate volatility in the stock price data.
    - The low MAE (4.02) suggests consistent predictions, making the model reliable for short-term forecasting tasks like the 5-day prediction horizon.
    - The iterative forecasting approach ensures the model adapts to its own predictions, making it practical for real-world deployment scenarios.
- **Limitations**:
    - The model may struggle with extreme market events (e.g., sudden crashes or spikes), as it relies solely on historical price and volume data without external factors like news sentiment or macroeconomic indicators.
    - The assumption of constant volume in the forecasting phase may oversimplify real-world dynamics, potentially affecting the accuracy of features like Volume_Change_5.
    - The high R^2 (0.97) raises concerns about overfitting, which should be validated with additional out-of-sample data.