INTRODUCTION

The financial markets, particularly the stock market, are characterized by inherent volatility and complexity. Predicting stock prices accurately is a significant challenge due to the numerous factors that influence market movements, including economic indicators, geopolitical events, company performance, and investor sentiment. For individual and institutional investors, the ability to forecast stock price trends is crucial for making informed investment decisions, managing risk, and maximizing returns. However, the extent and manner in which stock markets influence economic growth can differ across countries and periods due to variations in market structures, regulatory environments, financial system integration, and macroeconomic conditions. This project examines the effectiveness of both classical statistical methods and deep learning approaches in stock price forecasting, utilizing historical Yahoo stock data from 2015 to 2020. It addresses the problem of stock price prediction, aiming to develop a model that can provide reasonably accurate forecasts to aid in investment decision-making using different models like AR, ARIMA, SARIMAX, LSTM, etc..

The ultimate aim of this project is to provide actionable insights into the relative effectiveness of these forecasting methods on real-world stock data, to inform both academic research and practical investment strategies. The findings offer a balanced perspective on the evolving landscape of time series analysis, demonstrating that while deep learning models like LSTM show potential for complex pattern recognition, robust classical models like ARIMA remain strong contenders, especially for certain types of financial time series. By rigorously evaluating multiple models on the same dataset, this project contributes to the ongoing conversation in financial analytics regarding the optimal tools and methodologies for market forecasting.

The significance of this project lies in its potential to provide a practical tool for investors. An effective stock prediction model can enhance the decision-making process, reduce the risks associated with stock market investments, and potentially lead to better financial outcomes. Furthermore, this project contributes to the understanding of stock market dynamics and the application of data science techniques in finance.

DATA DESCRIPTION

This report aims to explore the relationship between stock market development and economic growth by analyzing stock market data retrieved from the **Kaggle** website about Yahoo, covering the years **November 2015 to November 2020.** The significance of this project lies in its potential to provide a practical tool for investors. An effective stock prediction model can enhance the decision-making process, reduce the risks associated with stock market investments, and potentially lead to better financial outcomes. Furthermore, this project contributes to the understanding of stock market dynamics and the application of data science techniques in finance.

This five-year span captures several key economic phases, including recovery from the global financial crisis, periods of economic expansion, the COVID-19 pandemic, and subsequent market volatility and recovery.

The study examines historical price trends, trading volumes, volatility patterns, and other relevant indicators for selected stock indices and companies. The focus is on identifying how changes in the stock market reflect and potentially influence broader economic trends. This analysis is expected to offer a clearer understanding of how stock markets act as indicators of economic health and play a role in shaping the financial landscape over time.

The dataset contains 1,825 entries along with & 7 columns in the dataset includes the following fields: -

Date: The trading date

Open: The opening price of the stock on the given date

High: The highest price reached during the trading day

Low: The lowest price reached during the trading day

Close: The closing price of the stock on the given date

Volume: The number of shares traded on the given date

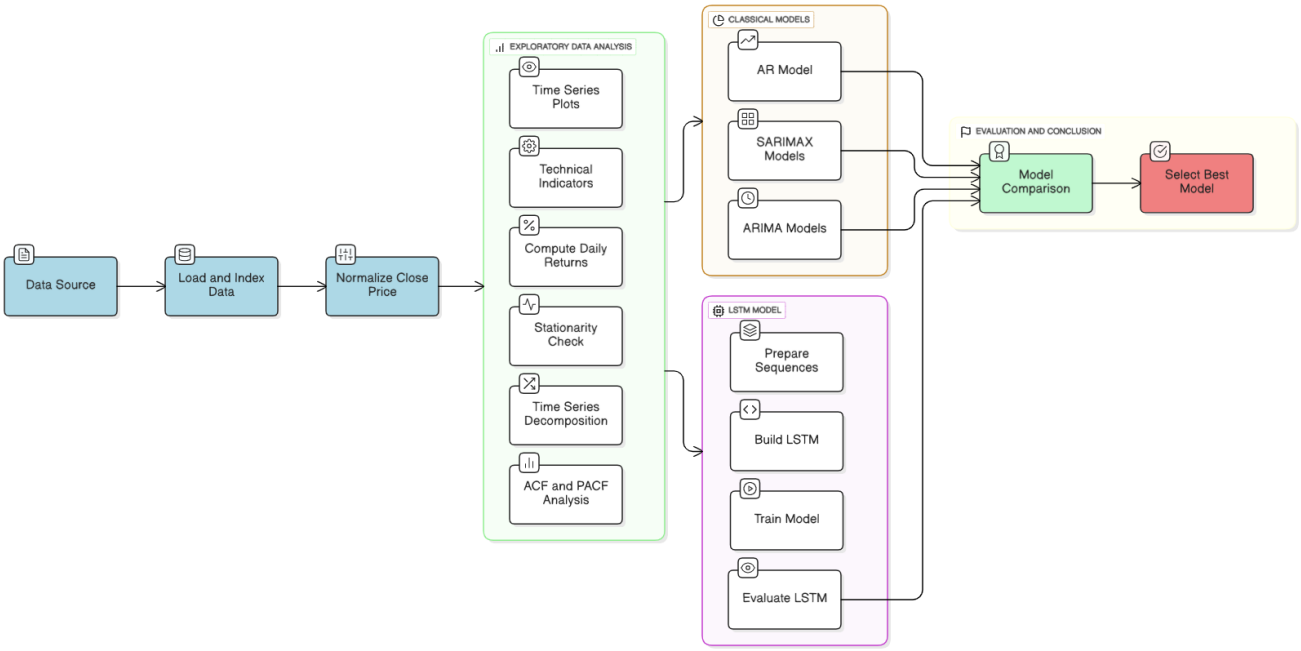
Adj Close: The adjusted closing price, accounting for corporate actions such as dividends, stock splits, etc.

OBJECTIVES

The primary objectives of this project are:

1. To explore and analyze historical stock price data to identify patterns and trends.
2. To implement and evaluate various time series forecasting models and machine learning algorithms for stock price prediction.
3. To compare the performance of different models using appropriate evaluation metrics.
4. To develop a predictive model that offers insights into future stock price movements.

ARCHITECTURE TECHNIQUES



Stock Market Volatility and Prediction Challenges

Stock market volatility refers to the degree of variation in trading prices over time. High volatility indicates significant price swings, making the market riskier but also offering potential for higher returns. Conversely, low volatility suggests more stable prices.

Predicting stock prices is notoriously difficult due to several inherent challenges:

- Non-linearity: Stock price movements are often non-linear and do not follow simple mathematical patterns.
- Noise: Market data is often noisy, containing random fluctuations that can obscure underlying trends.
- External Factors: Unpredictable events, such as economic crises, political instability, or natural disasters, can have a profound impact on stock prices.
- Efficient Market Hypothesis: This theory suggests that stock prices already reflect all available information, making it impossible to consistently outperform the market through prediction.

Despite these challenges, advancements in machine learning and computational power have provided new avenues for tackling the problem of stock price prediction, offering the potential to uncover complex patterns that traditional methods might miss.

Classical Statistical Methods in Stock Forecasting

The **ARIMA** (AutoRegressive Integrated Moving Average) family has historically been a mainstay in univariate time series analysis. Box and Jenkins (1970) first formalized the ARIMA approach, enabling systematic modeling of autocorrelation, trend, and seasonality. In financial contexts, ARIMA models have been effective for series with strong autocorrelation and non-stationarity, as shown by Atsalakis and Valavanis (2009), who demonstrated their utility for short-term stock price prediction.

The Autoregressive (**AR**) model is a foundational statistical method used in time series analysis, where future values of a variable are predicted based on its own past values. Specifically, an AR model of order *p*, denoted as AR(*p*), assumes that the current value of the series is a linear combination of the previous *p* values, plus a random error term. The mathematical form is:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t$$

where $\phi_1, \ldots, \phi_p$ are parameters to be estimated and ϵ_t is white noise.

The AR model captures the momentum or inertia in time series data, making it suitable for processes where past values have a strong influence on future values. However, it assumes the underlying series is **stationary**, meaning its statistical properties, like mean and variance are constant over time. Despite its simplicity and interpretability, the AR model is limited in its ability to capture complex patterns or seasonality, which leads to the use of more advanced models like ARMA, SARIMA, or neural networks when such complexities are present.

The Autoregressive Moving Average (ARMA) model is a widely used statistical model for analyzing and forecasting time series data by combining two components: the Autoregressive (AR) part, which regresses the variable on its own past values, and the Moving Average (MA) part, which models the error term as a linear combination of past error terms. An ARMA(*p*, *q*) model expresses the current value of a time series as a function of *p* previous values and *q* previous error terms, assuming the series is stationary. Mathematically, it is represented as

$$Xt = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t$$

where ϕ_i and θ_j are model coefficients and ϵ_t is white noise.

ARMA models are more flexible than AR or MA models alone, enabling them to capture both the autoregressive and stochastic structure of stationary time series. However, they are still limited in handling seasonality or non-stationary trends, which are addressed by more advanced models like SARIMA.

The Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors(**SARIMAX)** model extends the ARIMA model by incorporating seasonal components, making it suitable for time series data that exhibit both non-stationarity and repeating seasonal patterns. SARIMAX adds seasonal autoregressive (SAR), seasonal differencing (D), and seasonal moving average (SMA) terms to the standard ARIMA framework, denoted as SARIMAX(p, d, q)(P, D, Q, s), where s is the length of the seasonal cycle. This allows the model to capture both short-term dynamics and long-term seasonal trends. The seasonal differencing helps remove periodic fluctuations, while the seasonal AR and MA terms model seasonal dependencies and shocks. SARIMAX is particularly powerful for forecasting data with regular intervals, such as monthly sales or quarterly temperatures, where seasonality plays a critical role. However, SARIMAX models rely on linear assumptions and can become complex to tune as the number of parameters increases.

But all the models, however, may struggle with highly non-linear patterns or sudden regime shifts.

Deep Learning Approaches: LSTM (Long Short Term Memory)

The Long Short-Term Memory (LSTM) network represents a significant advancement over traditional time series models like AR, ARMA, and SARIMA by effectively capturing long-range temporal dependencies and nonlinear patterns that classical models often struggle with. While AR, ARMA, and SARIMA rely on strong assumptions of linearity, stationarity, and predefined seasonal structure, LSTM networks automatically learn both short- and long-term dynamics directly from the data without requiring manual feature engineering or differencing. Unlike traditional models that have limited memory, LSTMs use gated memory cells to retain relevant information across long sequences, making them especially powerful for forecasting tasks involving complex trends, variable seasonality, and multivariate inputs. This flexibility and learning capacity make LSTMs a robust deep learning alternative for real-world time series prediction.

Tools and Libraries:

The project utilizes several Python libraries for data manipulation, analysis, visualization, and modeling: - pandas and numpy for data manipulation - matplotlib and seaborn for visualization - scikit-learn for traditional machine learning models - statsmodels for time series analysis - tensorflow and keras for deep learning models

Mean Squared Error (MSE):

Mean Squared Error is a common metric for regression problems, measuring the average of the squares of the errors between predicted and actual values. Lower MSE values indicate better model performance.

Root Mean Squared Error (RMSE):

Root Mean Squared Error is the square root of MSE and provides a measure of the average magnitude of the error in the same units as the target variable. .

Mean Absolute Error (MAE):

Mean Absolute Error measures the average absolute difference between predicted and actual values. It is less sensitive to outliers than MSE.

R-squared (R^2) Score:

R-squared measures the proportion of variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, with higher values indicating better fit. For the LSTM model, the R-squared score on the test set was 0.8539, indicating that the model explains about 85.39% of the variance in the stock prices.

Model Outputs and In-Sample Fit:

The models were evaluated on their ability to fit the training data (in-sample fit) and to predict unseen data (out-of-sample performance). The in-sample fit provides insights into how well the model captures the patterns in the known data.

Preprocessing Steps

Before analysis and modeling, several preprocessing steps were applied to the data:

1. Data loading and inspection to ensure completeness and quality

The figure below gives an idea of all the columns of the dataset.

	Date	High	Low	Open	Close	Volume	Adj Close
0	2015-11-23	2095.610107	2081.389893	2089.409912	2086.590088	3.587980e+09	2086.590088
1	2015-11-24	2094.120117	2070.290039	2084.419922	2089.139893	3.884930e+09	2089.139893
2	2015-11-25	2093.000000	2086.300049	2089.300049	2088.870117	2.852940e+09	2088.870117
3	2015-11-26	2093.000000	2086.300049	2089.300049	2088.870117	2.852940e+09	2088.870117
4	2015-11-27	2093.290039	2084.129883	2088.820068	2090.110107	1.466840e+09	2090.110107

Again, the distribution and shape of the columns can be observed from below.

	Date	High	Low	Open
count	1825	1825.000000	1825.000000	1825.000000
mean	2018-05-23 00:00:00	2660.718673	2632.817580	2647.704751
min	2015-11-23 00:00:00	1847.000000	1810.099976	1833.400024
25%	2017-02-21 00:00:00	2348.350098	2322.250000	2341.979980
50%	2018-05-23 00:00:00	2696.250000	2667.840088	2685.489990
75%	2019-08-22 00:00:00	2930.790039	2900.709961	2913.860107
max	2020-11-20 00:00:00	3645.989990	3600.159912	3612.090088
std	NaN	409.680853	404.310068	407.169994

	Close	Volume	Adj Close
count	1825.000000	1.825000e+03	1825.000000
mean	2647.856284	3.869627e+09	2647.856284
min	1829.079956	1.296540e+09	1829.079956
25%	2328.949951	3.257950e+09	2328.949951
50%	2683.340088	3.609740e+09	2683.340088
75%	2917.520020	4.142850e+09	2917.520020
max	3626.909912	9.044690e+09	3626.909912
std	407.301177	1.087593e+09	407.301177

Summary statistics of the dataset reveal the range and distribution of the stock prices over the covered period: - The minimum closing price was approximately $1,829 - The maximum closing price reached approximately $3,626.91 - The mean closing price over the period was approximately $2,647.86 - The standard deviation of the closing price was approximately $407.30.

2. Checking for missing values

In the dataset, there are no missing values. The necessary output is given below.

```
[ ] print(data.isnull().sum())
```

```
Date         0
High         0
Low          0
Open         0
Close        0
Volume       0
Adj Close    0
dtype: int64
```

3. Train-test splitting to evaluate model performance

To evaluate the performance of the models, the dataset was split into training and testing sets. The training set consists of the first 80% of the data, and the testing set consists of the remaining 20%. This chronological split respects the time-series nature of the data and simulates a realistic forecasting scenario.

Exploratory Data Analysis (EDA)

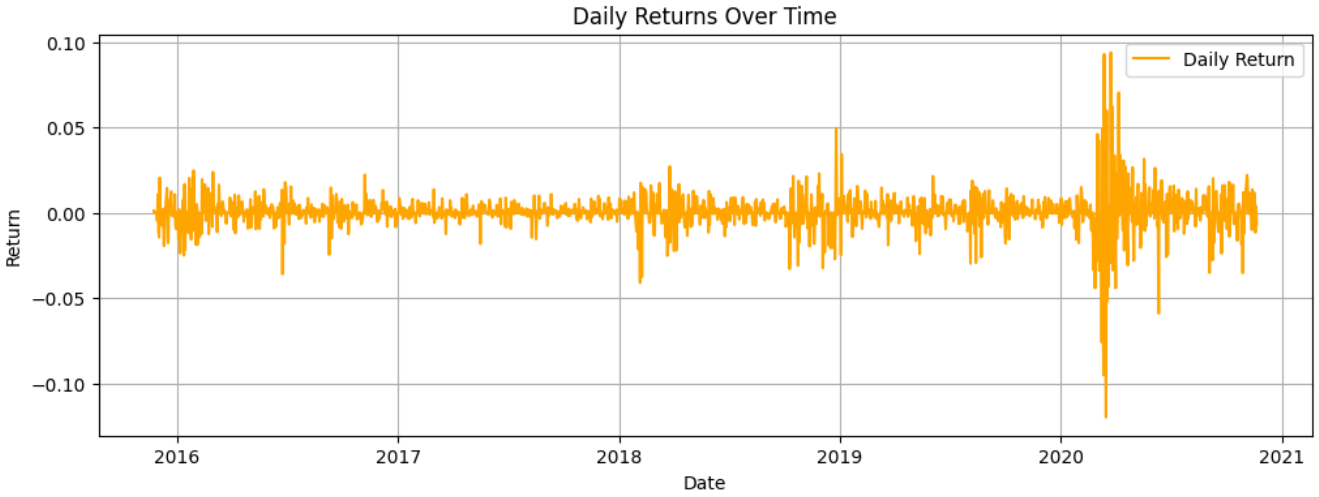
(i) Time-Series Plots

To understand the overall trend and patterns in the stock prices, a time-series plot of the closing prices was created. This visualization provides a clear picture of how the stock has performed over the five-year period, highlighting long-term trends, short-term fluctuations, and any significant upward or downward movements. It also helps identify periods of high volatility, potential seasonality, and any structural breaks or anomalies in the data. By visualizing the closing prices over time, we can gain valuable insights into the stock's behavior, which serves as a foundation for further statistical analysis and model development..

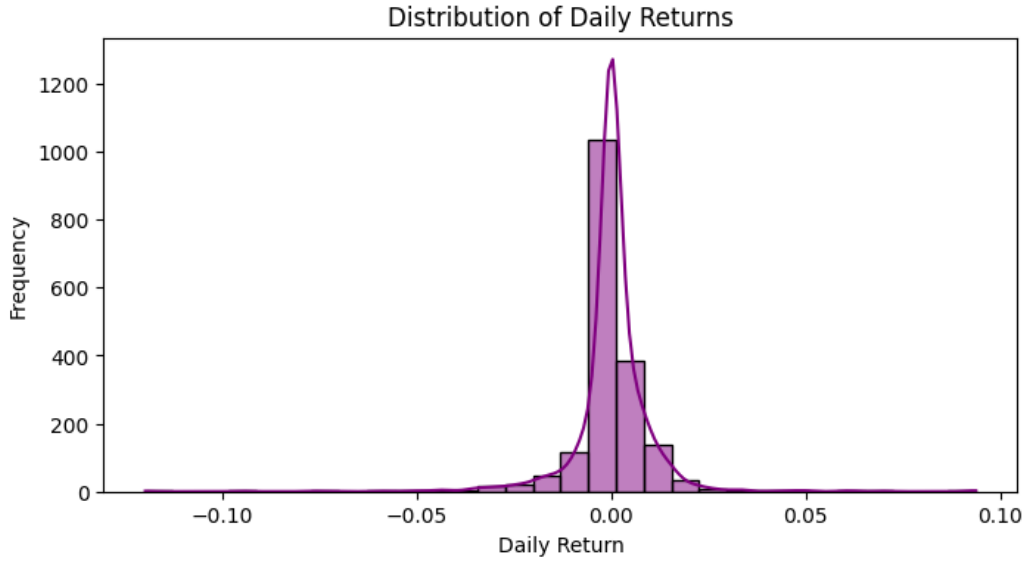
The time-series plot reveals several key insights of an overall upward trend in the stock price from 2015 to 2020, with the price increasing from around $2,000 to over $3,500. Also, a significant drop in early 2020 during the COVID-19 pandemic, with prices falling below $2,250 3. A strong recovery following the pandemic-induced drop, with prices reaching new highs by late 2020 4. Several periods of volatility throughout the timeframe, with notable fluctuations in 2018 and 2020.The same type of trend is also noticeable for the opening price for the time series

(ii) Daily Returns

To better understand the volatility of the stock, daily returns were calculated and visualized. Daily returns represent the percentage change in the closing price from one day to the next and provide insights into the day-to-day volatility of the stock.



A relatively symmetric variation in daily returns can be observed throughout the timeline, reflecting regular fluctuations typical of stock market behavior. However, a noticeable deviation from this pattern occurred during the COVID-19 outbreak, where the market experienced a sharp and abrupt decline in returns. This period is marked by heightened volatility and an unprecedented dip, highlighting the significant impact of the global health crisis on investor sentiment and market stability. The sudden drop, followed by irregular movements, indicates a period of uncertainty and panic-driven trading, which stands out starkly against the more stable return patterns seen in the pre- and post-pandemic phases.

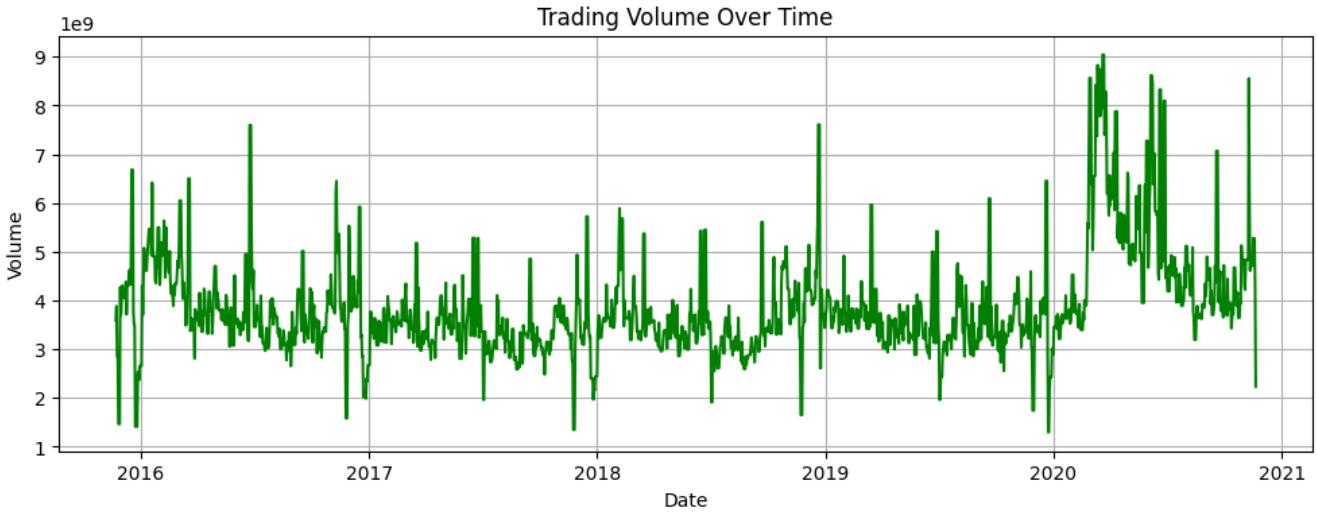


Thus, from the above histogram, it can be concluded that :

1. Most daily returns fall within a range of -2% to +2%, indicating relatively moderate day-to-day volatility
2. Extreme movements are observed during the COVID-19 pandemic in early 2020, with daily returns exceeding ±5%.
3. The distribution of returns is approximately symmetric but has fat tails, indicating more extreme movements than would be expected in a normal distribution
4. The mean daily return is slightly positive, consistent with the overall upward trend in the stock price.

(iii) Volume Traded Over Time

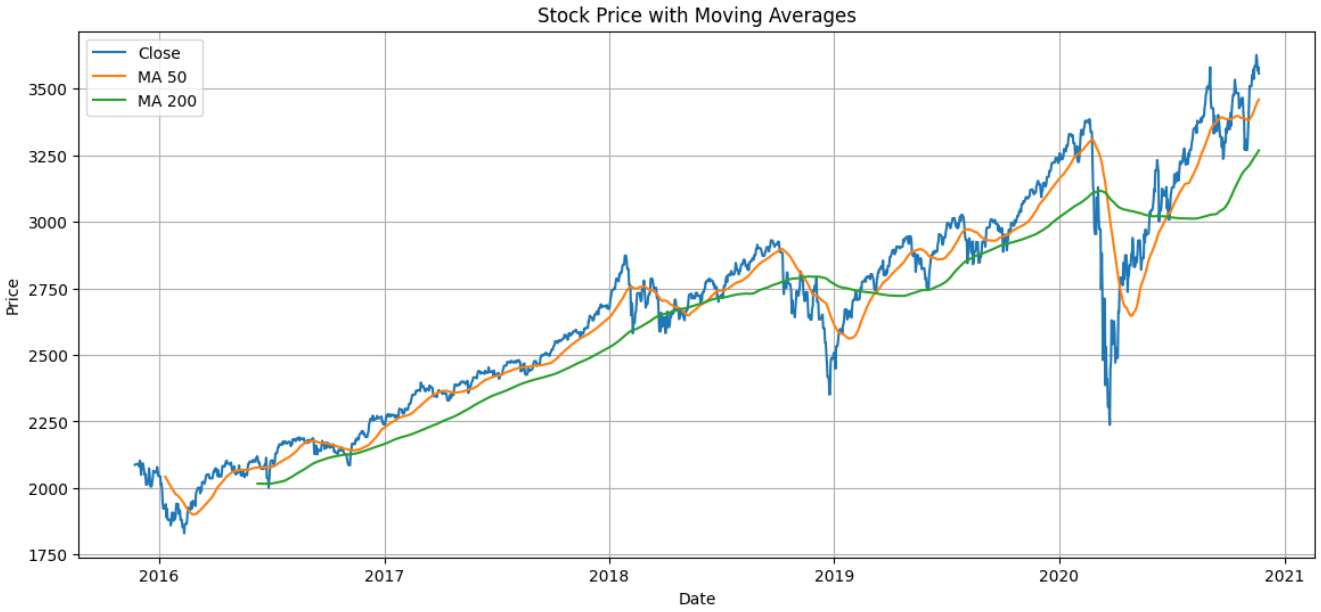
Trading volume is an important indicator of market activity and can provide additional context for price movements. A high volume accompanying a price movement suggests stronger conviction in that direction



Similarly, like the daily rate, the trading volume has symmetric variation over time except covid time. The volume analysis shows:

1. Trading volume varies significantly over time, with several notable spikes
2. The highest trading volumes occurred during the COVID-19 market crash in March 2020
3. Periods of high volatility in prices generally coincide with increased trading volumes
4. Average daily trading volume is approximately 3.87 billion shares

(iv) Moving Averages (MA50 & MA200)



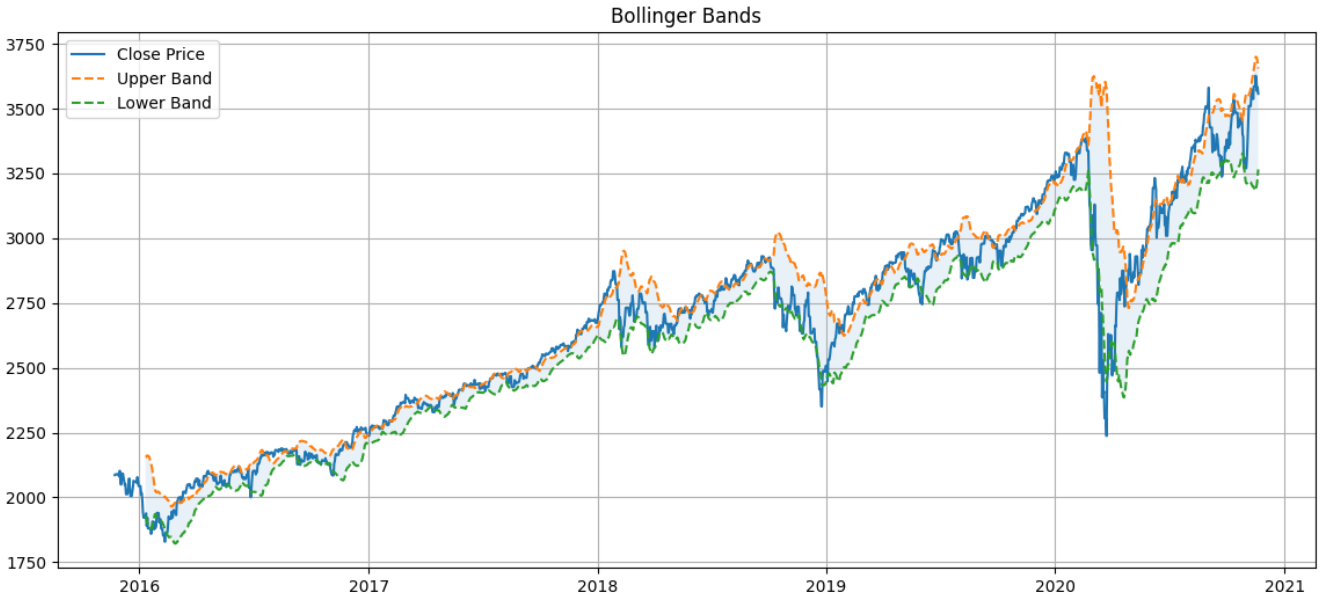
Moving averages help smooth out price data to identify the direction of the trend while filtering out short-term fluctuations and noise. Two commonly used moving averages are the 50-day (MA50) and 200-day (MA200) moving averages.

The moving averages analysis reveals:

1. The 50-day MA (orange line) responds more quickly to price changes than the 200-day MA (green line).
2. Crossovers between the MA50 and MA200 occurred several times during the period, notably in early 2016 (bullish crossover) and early 2020 (bearish crossover).
3. During strong uptrends (2017-2018 and late 2020), the price consistently stayed above both moving averages.
4. During the COVID-19 crash, the price fell significantly below both moving averages before recovering.
5. The 200-day MA acted as a support level during several pullbacks in 2017 and 2019.

(v) Bollinger Bands

Bollinger Bands consist of a middle band (typically a 20-day moving average) and two outer bands that are two standard deviations away from the middle band. They help identify periods of high and low volatility and potential overbought or oversold conditions.

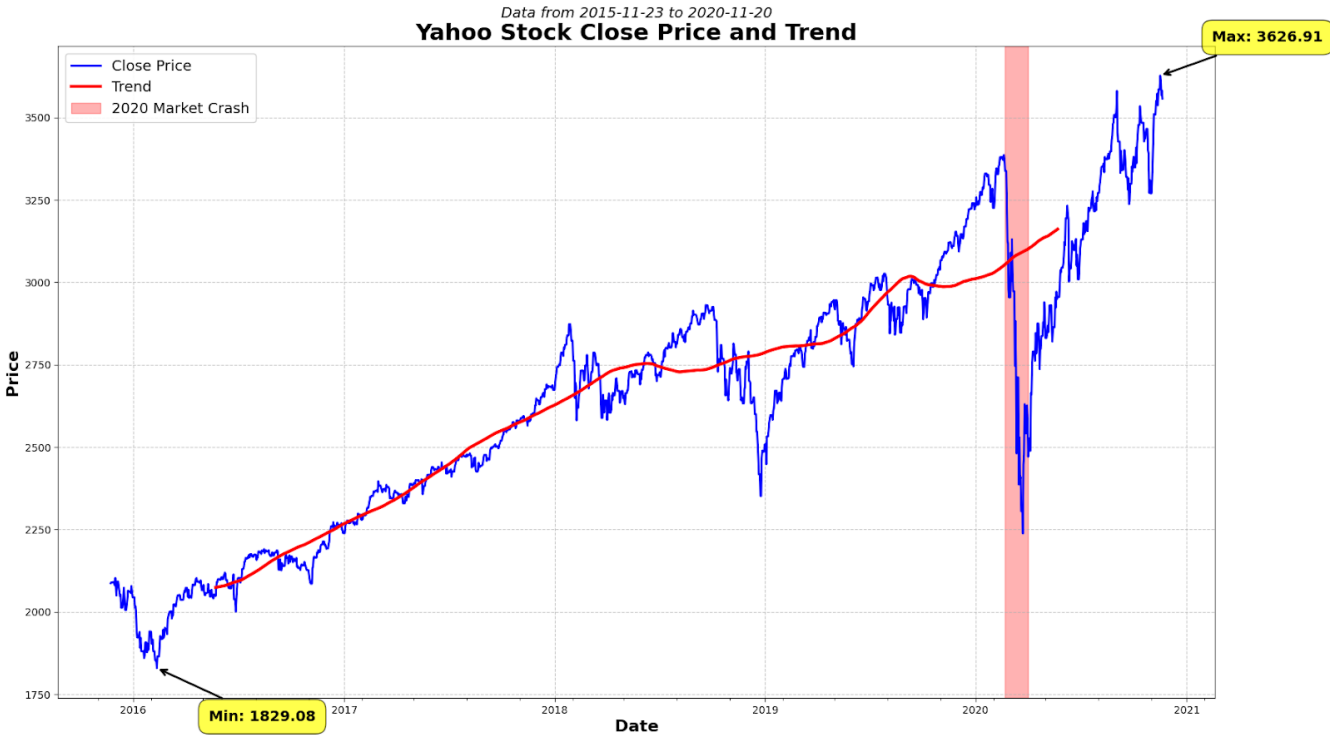


The Bollinger Bands analysis shows:

1. Periods of high volatility (wide bands) occurred during market stress in early 2016, late 2018, and most notably during the COVID-19 pandemic in 2020
2. Periods of low volatility (narrow bands) were observed in mid-2017 and early 2019
3. Price movements that touch or exceed the outer bands often revert toward the middle band.
4. The COVID-19 crash in March 2020 resulted in prices breaking below the lower band by a significant margin, indicating extreme oversold conditions.
5. The subsequent recovery saw prices frequently testing the upper band, suggesting strong bullish momentum.

(vi) COVID-19 crash and price

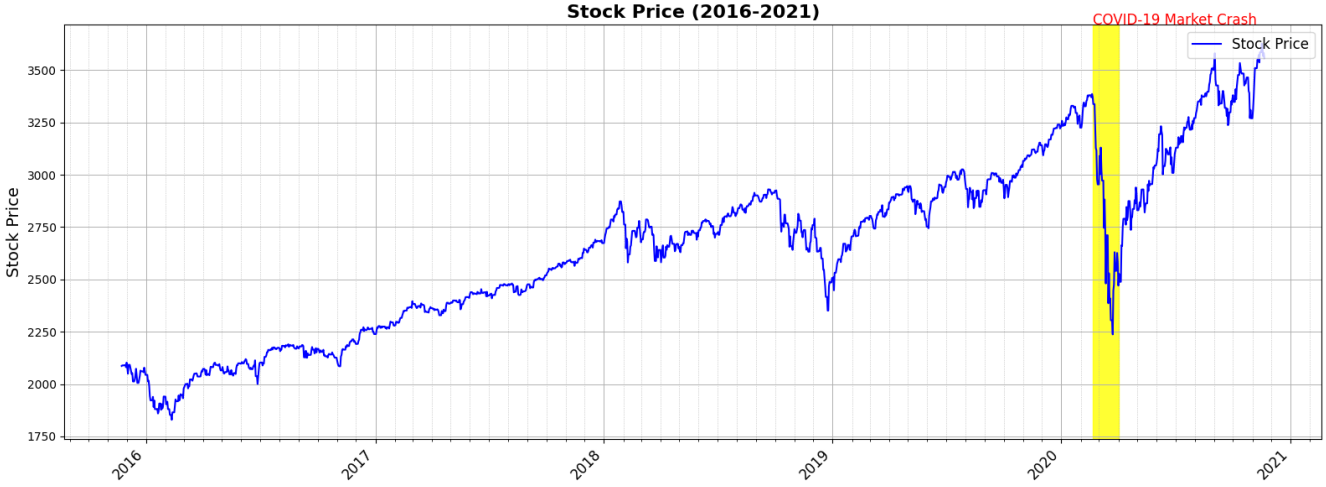
The time series plot illustrates Yahoo's daily closing stock prices from November 2015 to November 2020, overlaid with a smoothed trend line and highlighting the 2020 market crash due to the COVID-19 pandemic. The blue line represents the actual closing prices, which show significant short-term fluctuations and overall volatility, while the red trend line reveals a clear long-term upward trajectory. A sharp dip is observed in early 2020, corresponding to the global market disruption caused by the pandemic, followed by a strong and rapid recovery. The minimum price of 1829.08 occurred in early 2016, and the stock reached its peak at 3626.91 towards the end of 2020, indicating strong overall growth despite temporary setbacks. This visualization effectively captures both the resilience of the stock and the impact of external shocks on market behaviour.

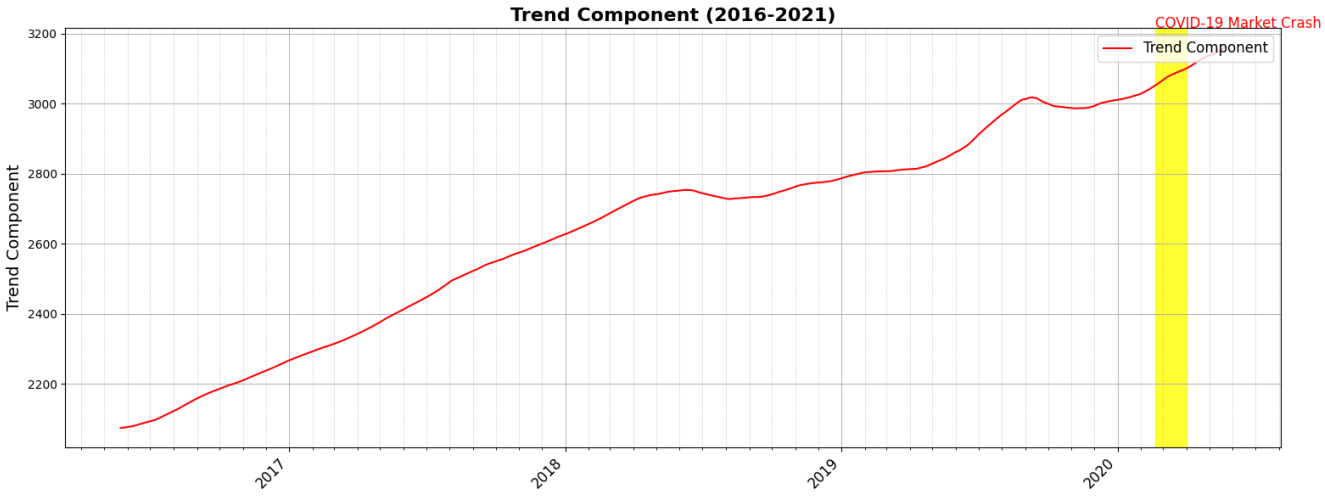


METHODOLOGY

Time Series Decomposition

Time series decomposition breaks down a time series into its constituent components: trend, seasonality, and residual (or random) components. This helps in understanding the underlying patterns in the data.





To gain deeper insights into the underlying structure of the stock price time series, a decomposition analysis was performed. This technique breaks the time series into three key components—trend, seasonality, and residual—allowing for a more detailed understanding of the factors driving price

movements. By isolating these components, we can better interpret long-term behaviour, identify any repetitive patterns, and observe irregular fluctuations that may be due to external shocks or market events. The findings from this decomposition are summarized below:

1. **Trend Component:**
    - Shows a clear long-term upward movement in the stock price.
    - A significant dip is observed during the COVID-19 pandemic, followed by a strong recovery.
    - Indicates the influence of macroeconomic and structural factors over time.
2. **Seasonal Component:**
    - Minimal or negligible seasonality is detected.
    - This is typical for daily financial time series where regular seasonal patterns are weak or absent.
3. **Residual Component:**
    - Captures random fluctuations and irregularities in the data.
    - Exhibits increased volatility during periods of market stress, such as the COVID-19 outbreak.
    - Reflects short-term shocks and unpredictable investor behavior.
4. **Overall Interpretation:**
    - The trend component explains the majority of the stock price movement.
    - Suggests that long-term factors have a more substantial impact than seasonal or purely random variations.

Stationarity Check (ADFuller Test)

Stationarity is an important property for many time series modeling techniques. A stationary time series has constant mean, variance, and autocovariance over time. The Augmented Dickey-Fuller (ADF) test is commonly used to check for stationarity.

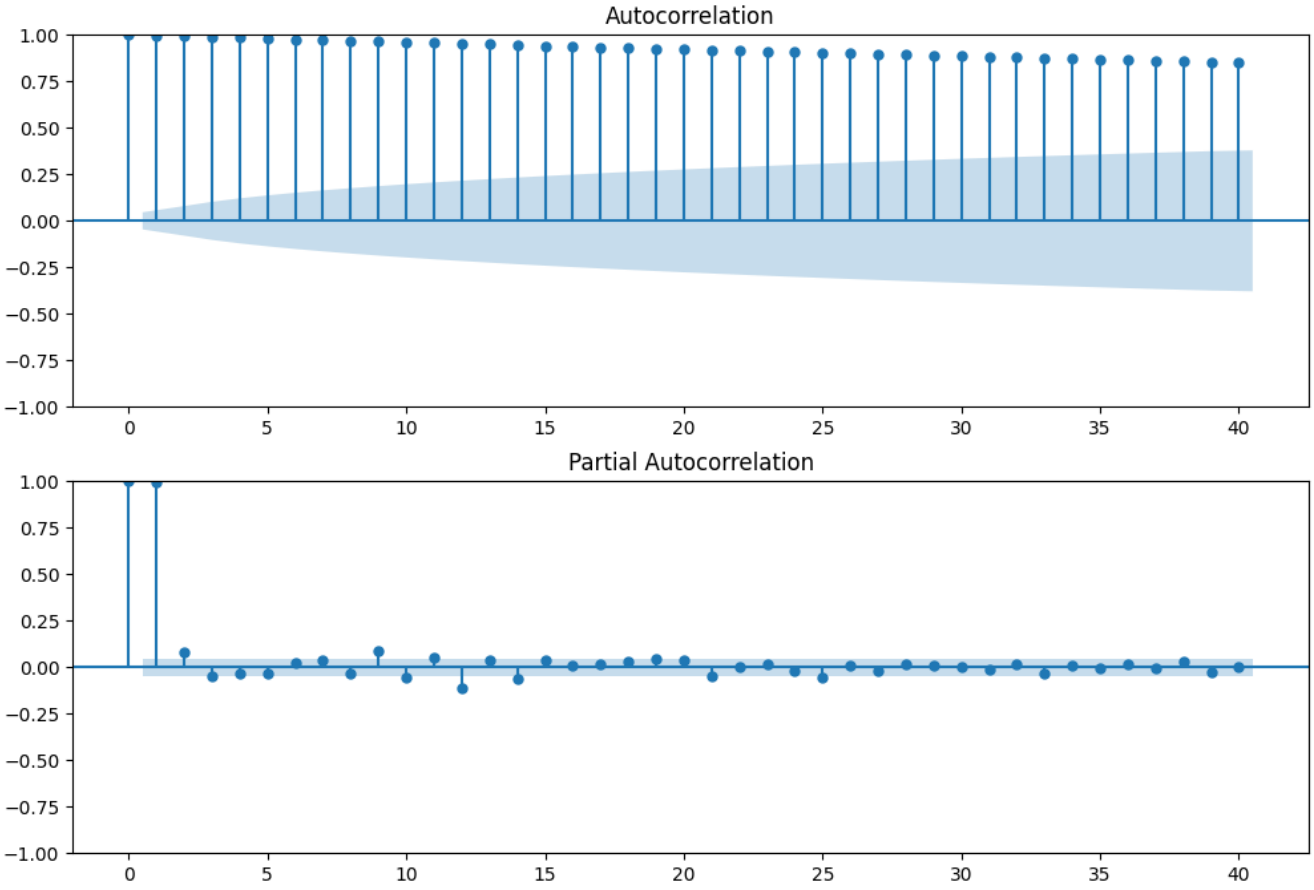
The ADF test was performed on the closing price series and the daily returns series. The results showed:

1. The closing price series is non-stationary (p-value > 0.05), which is expected for stock prices
2. The daily returns series is stationary (p-value < 0.05), making it suitable for certain time series models

This confirms that differencing (calculating returns) is an effective way to transform the non-stationary price series into a stationary series for modeling purposes.

Autocorrelation and Partial Autocorrelation (ACF and PACF) Plots

Autocorrelation and partial autocorrelation plots help identify patterns and dependencies in time series data. They are particularly useful for determining the appropriate parameters for ARIMA models.



The ACF and PACF analysis show:

1. The ACF plot shows minimal significant autocorrelation at most lags, with a few lags showing small but significant correlations
2. The PACF plot similarly shows limited significant partial autocorrelations
3. The lack of strong autocorrelation patterns suggests that daily returns exhibit limited predictability based solely on their past values
4. The pattern suggests that an AR(2) model might be appropriate for modeling the returns series

(i) Autoregressive (AR) Model

An autoregressive model predicts future values based on past values. The order of the AR model (p) indicates how many lagged values are used in the prediction.

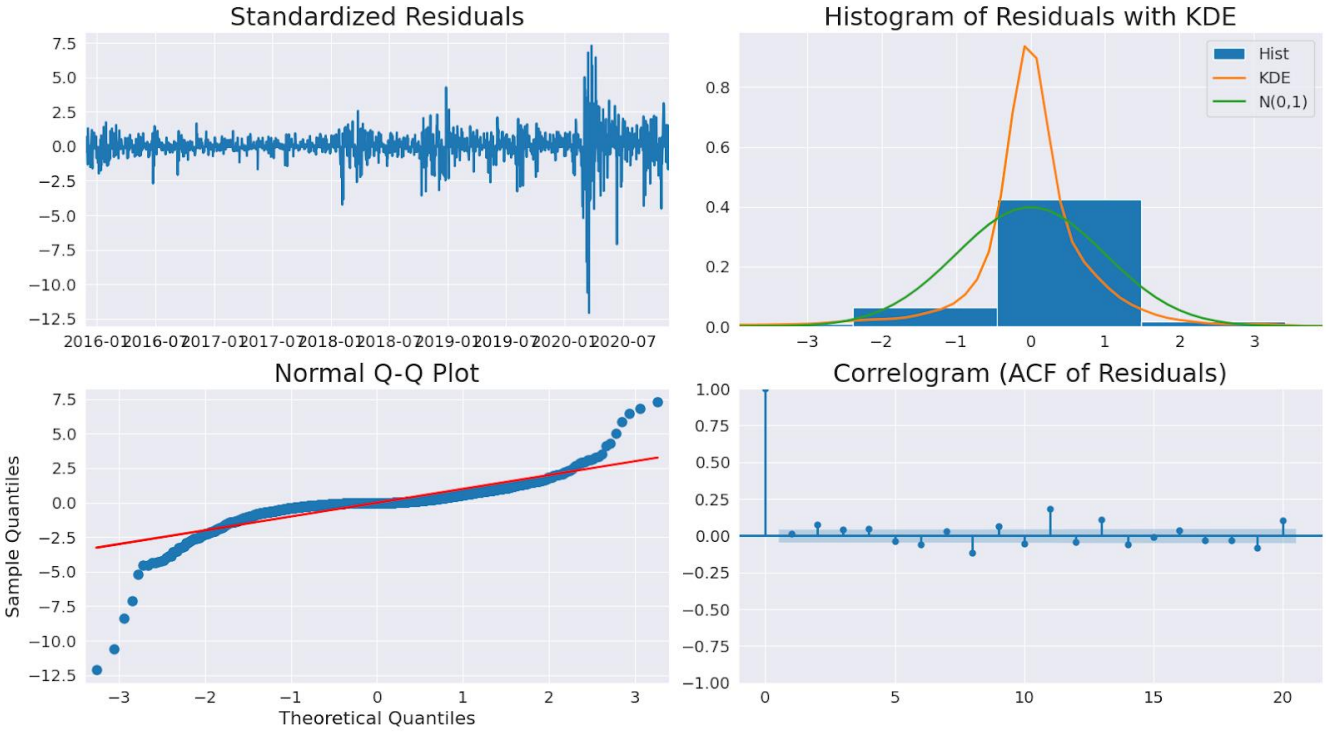
Based on the PACF plot, an AR(2) model was implemented, which uses the two most recent observations to predict the next value. The model was trained on the stationary daily returns series.

In-sample RMSE for AR(2): 26.9785

AIC: 17195.1613

BIC: 17217.1942

The AR(2) model showed moderate performance on the training data, capturing some of the patterns in the daily returns but missing many of the extreme movements.



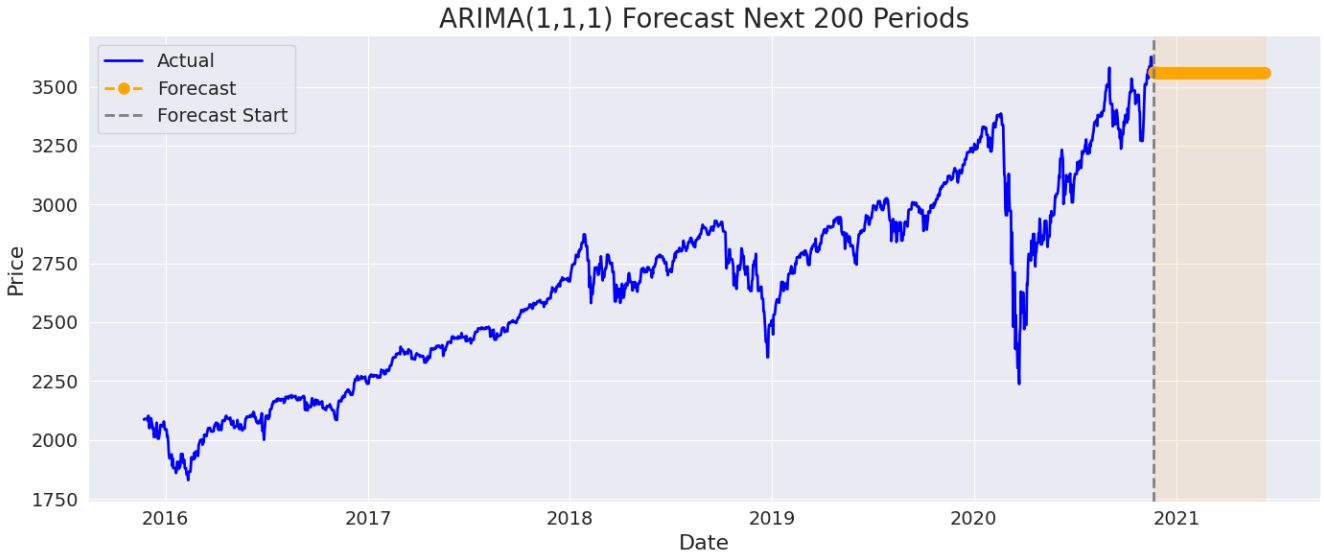
The AR(2) model results show:

1. The model captures some of the patterns in the daily returns but misses many of the extreme movements
2. The predictions are much less volatile than the actual returns
3. The model's performance metrics indicate limited predictive power: - RMSE: 26.9785.
4. The RMSE,AIC, and BIC values show good performance of the model.

(ii) Autoregressive Integrated Moving Average (ARIMA) Model

ARIMA models combine autoregressive (AR) and moving average (MA) components, with an integration (I) component to handle non-stationarity. Based on the ACF and PACF plots, an **ARIMA(1,1,1)** model was implemented. ARIMA(1,1,1) Model (In-Sample)

The ARIMA(1,1,1) model demonstrated better in-sample fit than the AR model, capturing the overall trend and some of the fluctuations in the closing prices.



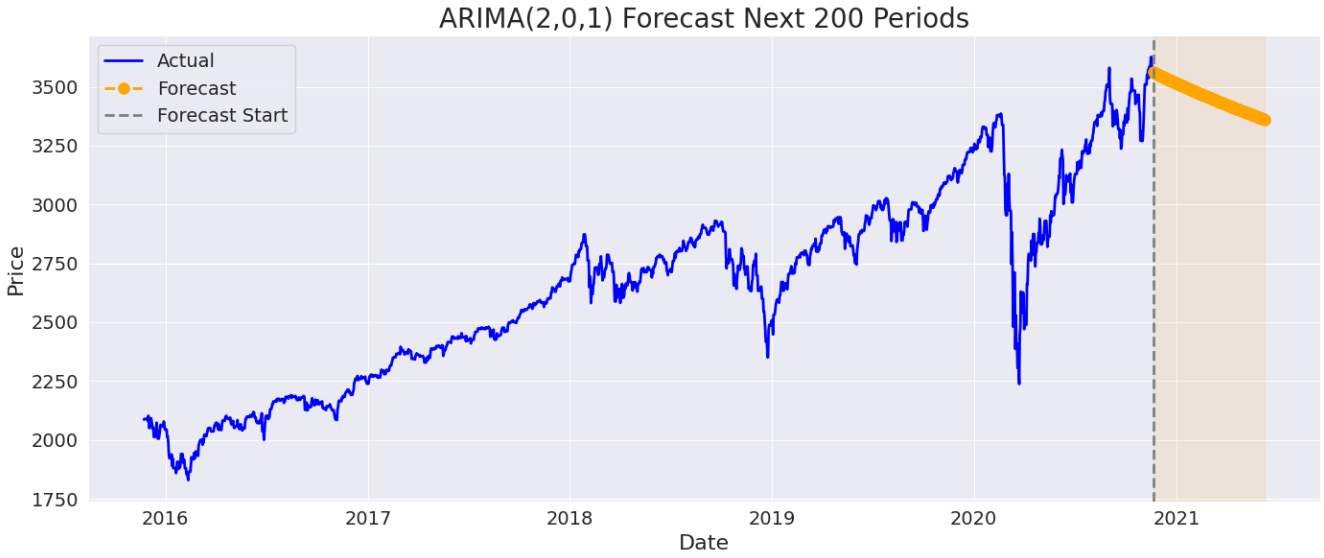
The above diagram shows the actual closing prices versus the ARIMA(1,1,1) model predictions

The ARIMA model results show:

1. The model captures the overall direction of price movements but fails to predict significant changes
2. The predictions lag behind actual price changes, particularly during rapid market movements
3. The model's performance metrics indicate poor predictive power: - **RMSE: - 55.7780 .**

The poor performance of the ARIMA model suggests that linear time series models may not be well-suited for capturing the complex, non-linear patterns in stock price data.

By applying an alternative **ARIMA model with parameters (2, 0, 1)**, a notable improvement in forecasting accuracy is observed compared to the previous model. This configuration—indicating an autoregressive component of order 2, no differencing, and a moving average component of order 1—appears to better capture the underlying structure and short-term dependencies in the data. As a result, the model yields a lower Root Mean Squared Error (RMSE), suggesting enhanced predictive performance and a closer fit to the observed values. This demonstrates the importance of experimenting with different ARIMA orders to identify the most suitable model for accurate time series forecasting.



The above diagram shows the actual closing prices versus the ARIMA(2,0,1) model predictions

The ARIMA model results show:

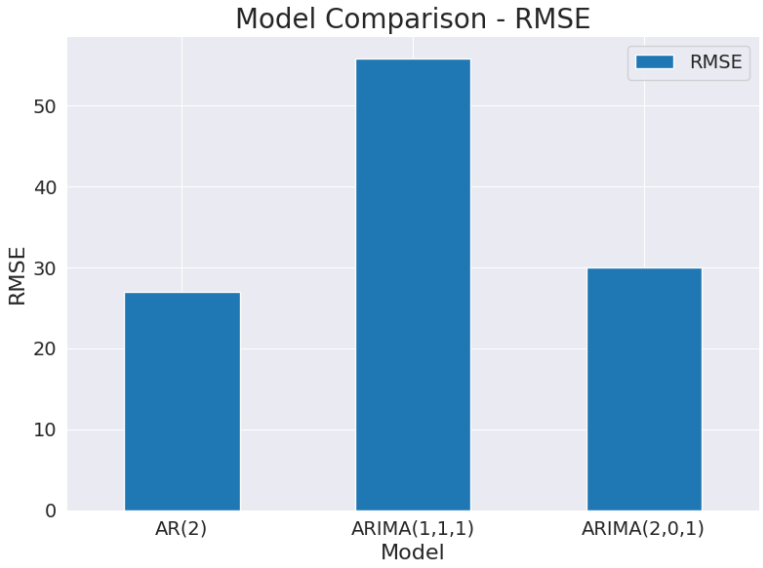
1. The model captures the overall direction of price movements and predicts changes better than the earlier ARIMA model.
2. The model's performance metrics indicate poor predictive power: - RMSE: - 29.9667 .

Comparison of Model Performances

Multiple models were implemented and evaluated to determine the most effective approach for stock price prediction. The models include: - AR(2) model ,ARIMA(1,1,1) and ARIMA(2,0,1) models.

The performance metrics for each model on the test set are summarized below:

Model	RMSE
AR(2)	26.978460
ARIMA(1,1,1)	55.778002
ARIMA(2,0,1)	29.966678



The above bar diagram helps in comparing the performance metrics of the different models.

(iii) SARIMAX Model

Model 1: SARIMAX(1,1,1)(1,1,1,12)

This model combines non-seasonal and seasonal components, each using first-order autoregression (AR), differencing, and moving average (MA). The non-seasonal component (1,1,1) uses first-order differencing for stationarity and includes AR(1) and MA(1) terms to capture short-term patterns. The seasonal component (1,1,1,12) models yearly patterns on a 12-period cycle, typical for monthly data. It uses seasonal differencing to handle recurring annual patterns. This model effectively balances complexity with flexibility, making it suitable for data showing both trends and seasonal behavior.

Model 2: SARIMAX(2,0,1)(1,1,1,12)

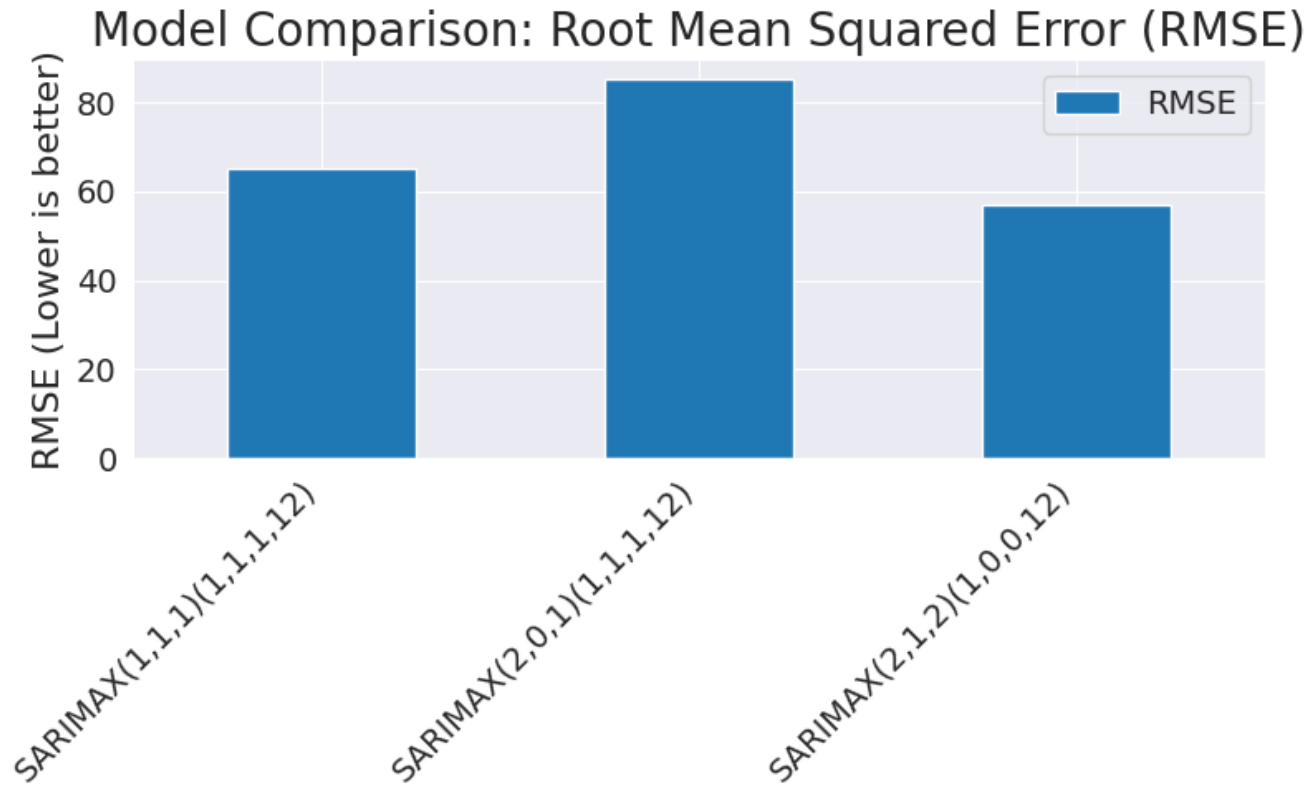
This model skips non-seasonal differencing, assuming the data is already stationary. It uses a second-order autoregressive term (AR(2)) to capture complex short-term patterns, along with an MA(1) term for noise. The seasonal component matches Model 1, using SAR(1), seasonal differencing, and SMA(1) on a 12-period cycle. By avoiding differencing while using a higher AR order, this model preserves data structure that might otherwise be lost—ideal for data with strong autocorrelation but no clear trend.

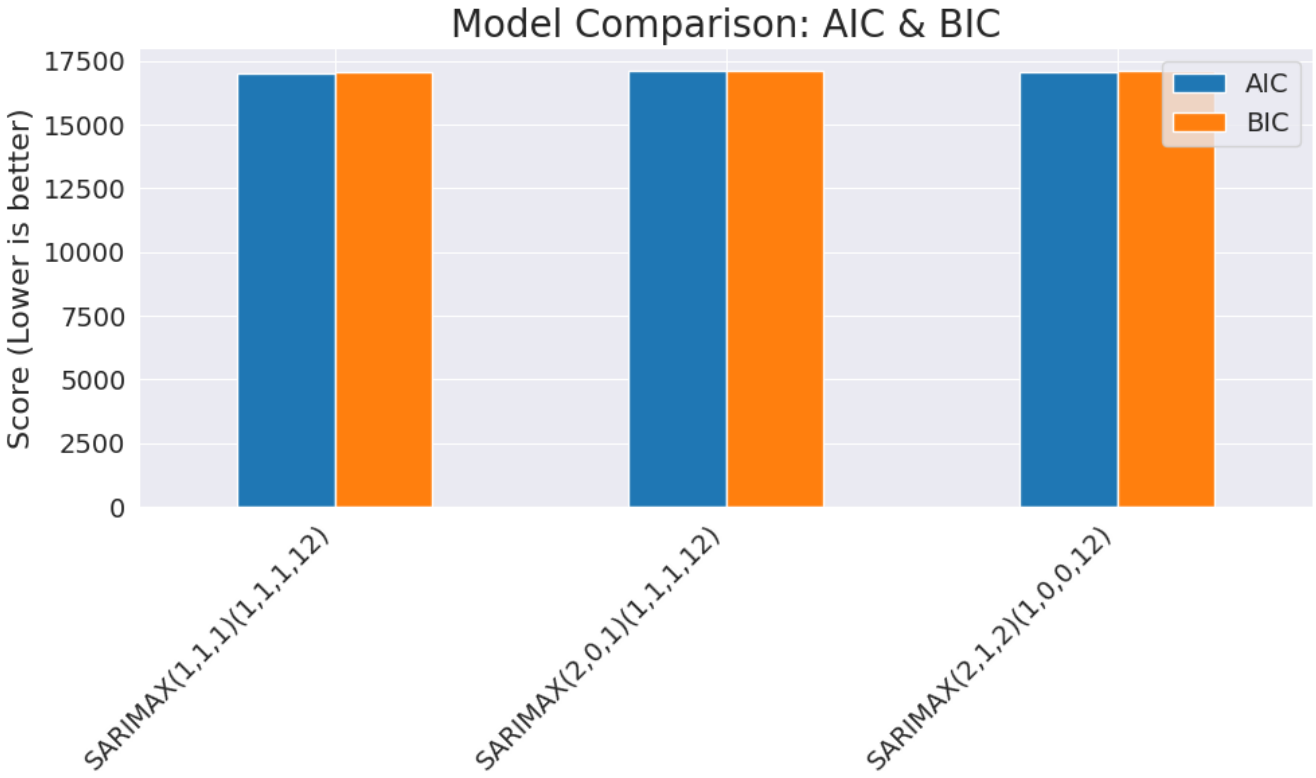
Model 3: SARIMAX(2,1,2)(1,0,0,12)

This model features a sophisticated non-seasonal structure with AR(2) and MA(2) terms, plus first-order differencing for stationarity. It uses a simpler seasonal component, with just a seasonal AR(1) term on a 12-period cycle. This design excels at capturing yearly patterns in data with mild seasonality but complex short-term fluctuations.

The following table summarizes the performance metrics for the three models:

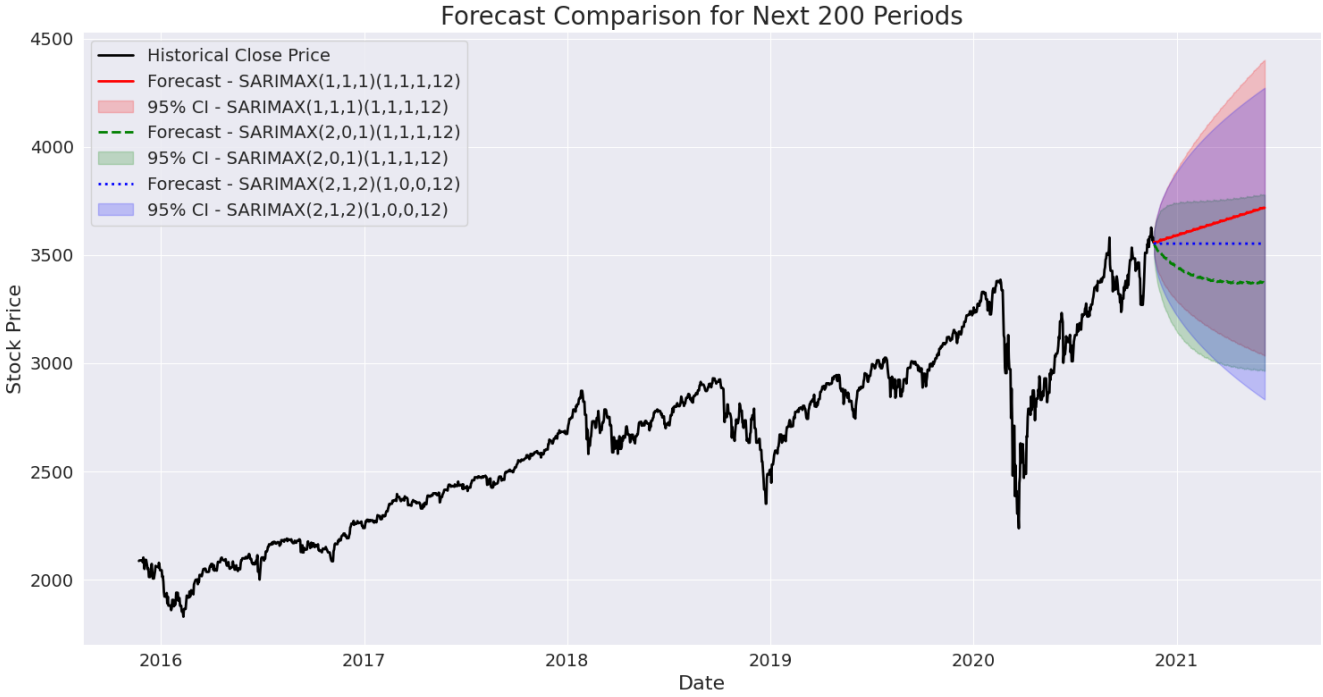
Model	AIC	BIC	RMSE
SARIMAX(1,1,1)(1,1,1,12)	17014.0066	17041.4788	65.1399
SARIMAX(2,0,1)(1,1,1,12)	17105.6529	17138.6228	85.3284
SARIMAX(2,1,2)(1,0,0,12)	17067.2269	17100.2333	56.8642


Model Comparison: Root Mean Squared Error (RMSE)
RMSE (Lower is better)
0
20
40
60
80
RMSE
SARIMAX(1,1,1)(1,1,1,12)
SARIMAX(2,0,1)(1,1,1,12)
SARIMAX(2,1,2)(1,0,0,12)


Model Comparison: AIC & BIC
Score (Lower is better)
0
2500
5000
7500
10000
12500
15000
17500
AIC
BIC
SARIMAX(1,1,1)(1,1,1,12)
SARIMAX(2,0,1)(1,1,1,12)
SARIMAX(2,1,2)(1,0,0,12)

The **SARIMAX(2,1,2)(1,0,0,12)** model achieved the **lowest Root Mean Squared Error (RMSE) of 56.8642**, indicating that it provided the best in-sample fit among the models evaluated. A lower RMSE suggests that the model's predictions are, on average, closest to the actual observed values, making it the most accurate in terms of minimizing forecasting errors within the training dataset. This model's more complex non-seasonal structure—featuring both higher-order autoregressive (AR) and moving average (MA) components—appears to capture the short-term dependencies in the data more effectively.

On the other hand, the **SARIMAX(1,1,1)(1,1,1,12)** model yielded the **lowest AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion)** scores. These criteria penalize model complexity, so lower values indicate a better trade-off between goodness of fit and model simplicity. This suggests that while SARIMAX(2,1,2)(1,0,0,12) may offer a better in-sample fit, the SARIMAX(1,1,1)(1,1,1,12) model is **more parsimonious**, potentially generalizing better to unseen data by avoiding overfitting. Thus, the choice between the two models depends on the priority: forecasting accuracy versus model interpretability and generalizability.



The figure presents a comparative visualization of stock price forecasts generated by three different SARIMAX models—**SARIMAX(1,1,1)(1,1,1,12)**, **SARIMAX(2,0,1)(1,1,1,12)**, and **SARIMAX(2,1,2)(1,0,0,12)**—plotted over the most recent segment of the time series and extending into a 200-period forecast horizon. The **black solid line** represents the historical closing prices of the stock, showing a strong upward trend from 2016 to 2020, along with periods of sharp dips and recoveries, particularly around the 2020 market crash.

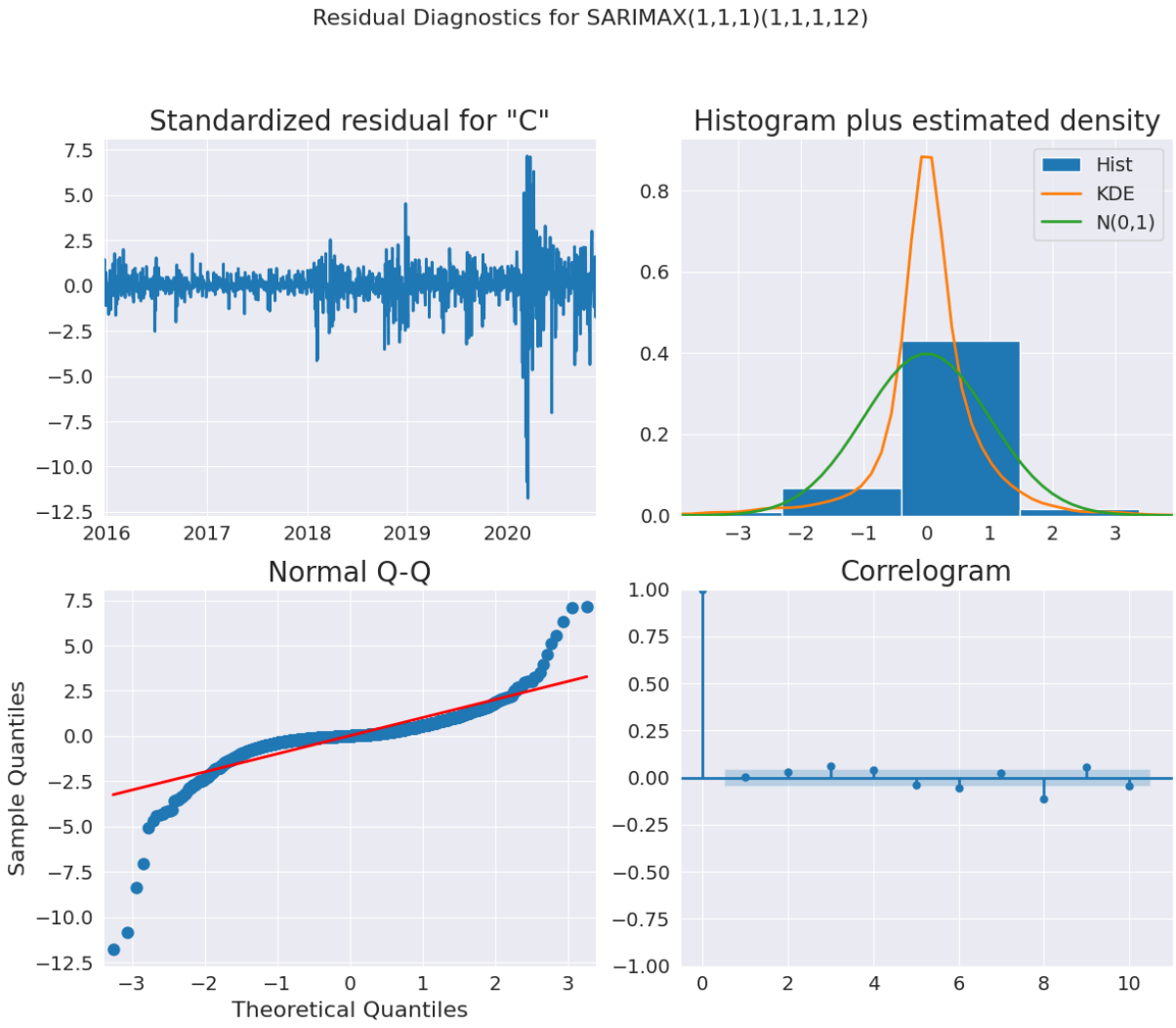
Each forecast is displayed as a distinct line:

- The **red dashed line** represents the forecast from **SARIMAX(1,1,1)(1,1,1,12)**.
- The **green dashed line** shows the forecast from **SARIMAX(2,0,1)(1,1,1,12)**.
- The **blue dashed line** corresponds to **SARIMAX(2,1,2)(1,0,0,12)**.

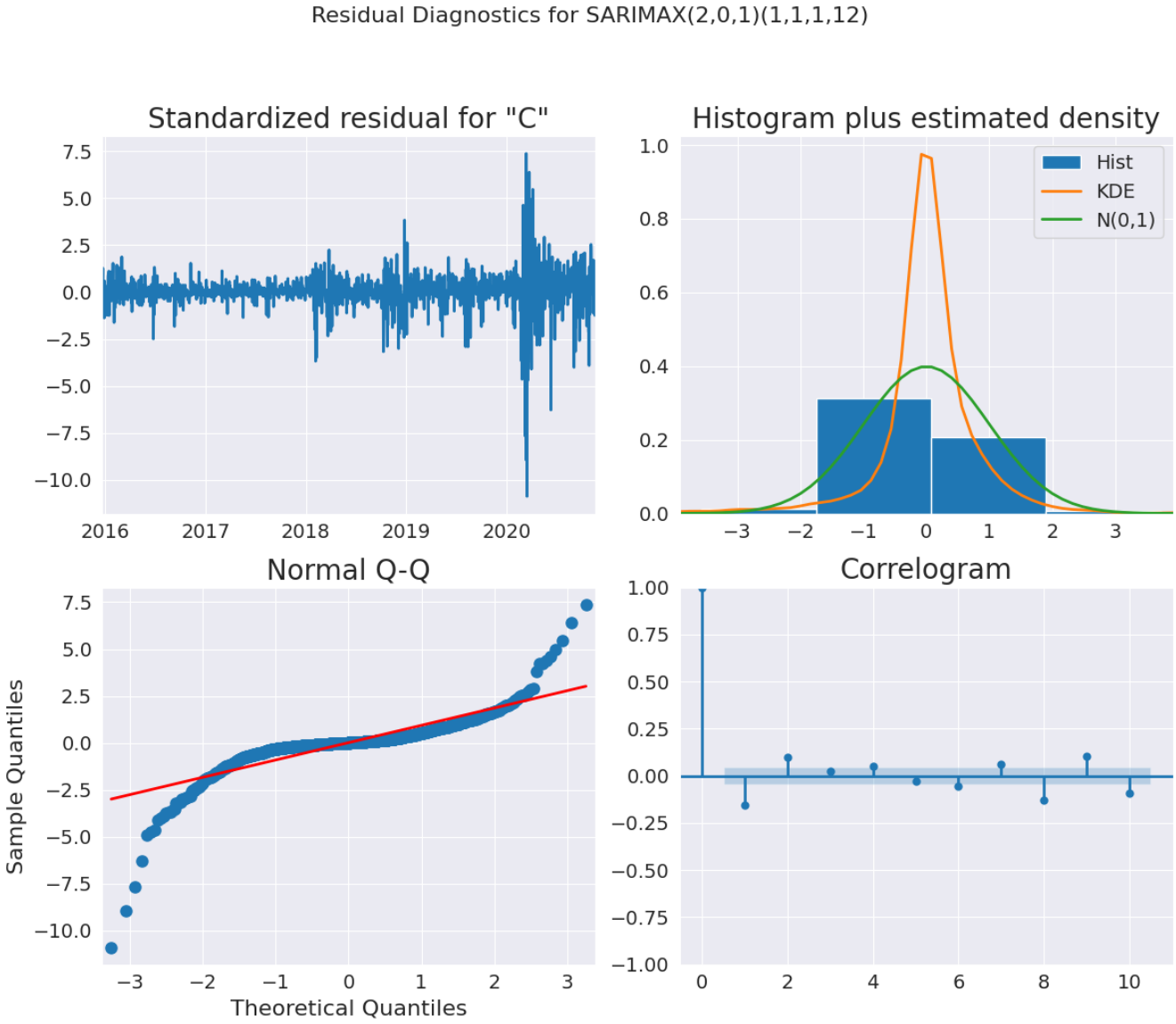
Accompanying each forecast is a **95% confidence interval (CI)** shaded in semi-transparent color bands—pink, green, and blue respectively—illustrating the model's uncertainty around its future predictions. These intervals widen as the forecast extends, reflecting increased uncertainty further into the future.

From the figure, it's evident that while all three models capture the general trend of the stock, they diverge in terms of forecast magnitude and confidence spread. Notably, **SARIMAX(2,1,2)(1,0,0,12)** appears more conservative, with narrower confidence bounds and a more stable trajectory. In contrast, **SARIMAX(1,1,1)(1,1,1,12)** projects a more aggressive upward trend but with a wider confidence interval, indicating higher volatility or uncertainty. This visualization effectively highlights the trade-offs between different model specifications in terms of forecast accuracy, variance, and uncertainty.

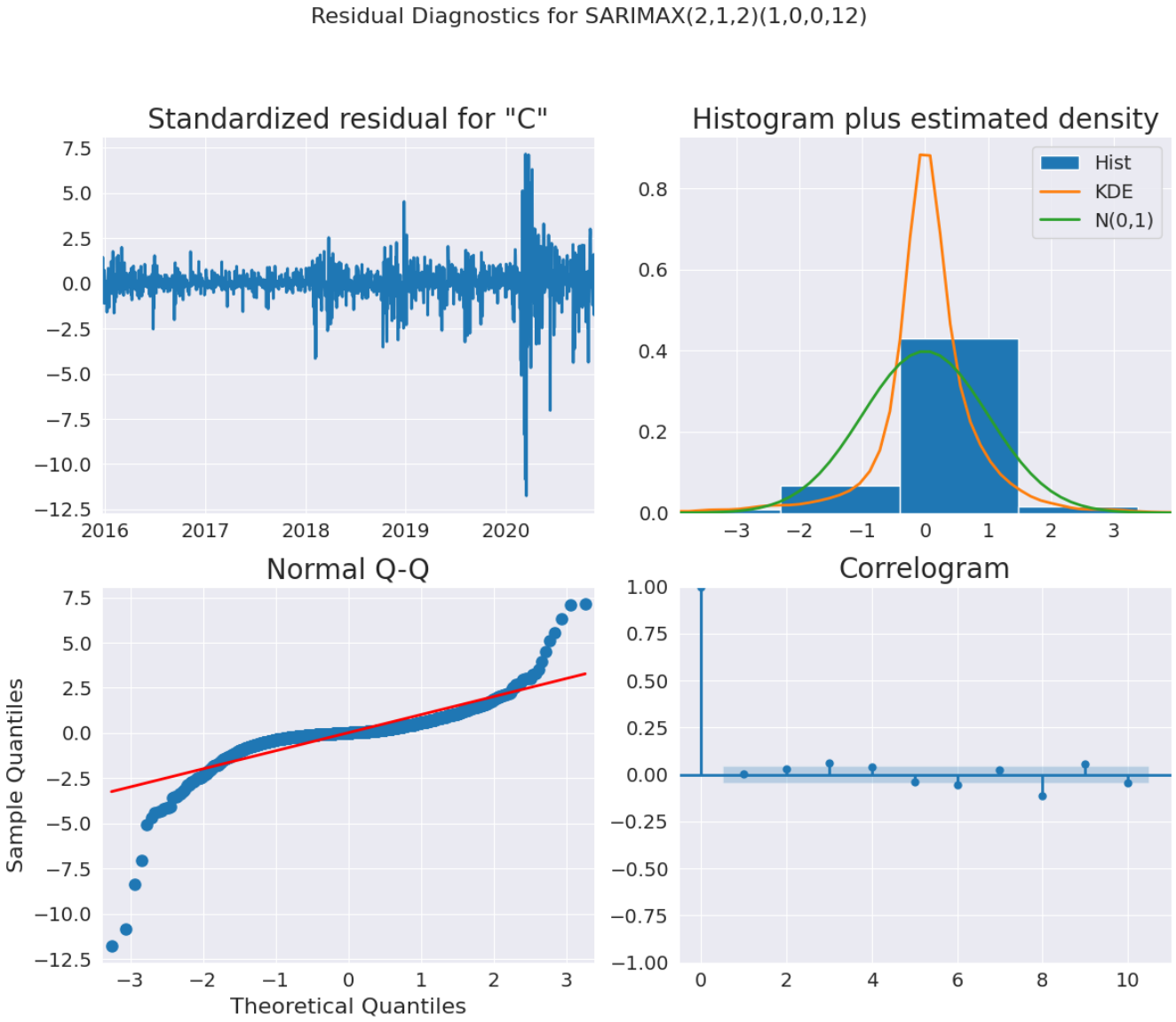
Residual Diagnostic Plots for sarimax(1,1,1)(1,1,1,12)



Residual Diagnostic Plots for sarimax(2,0,1)(1,1,1,12)



Residual Diagnostic Plots for sarimax(2,1,2)(1,0,0,12)



(iv) LSTM Model

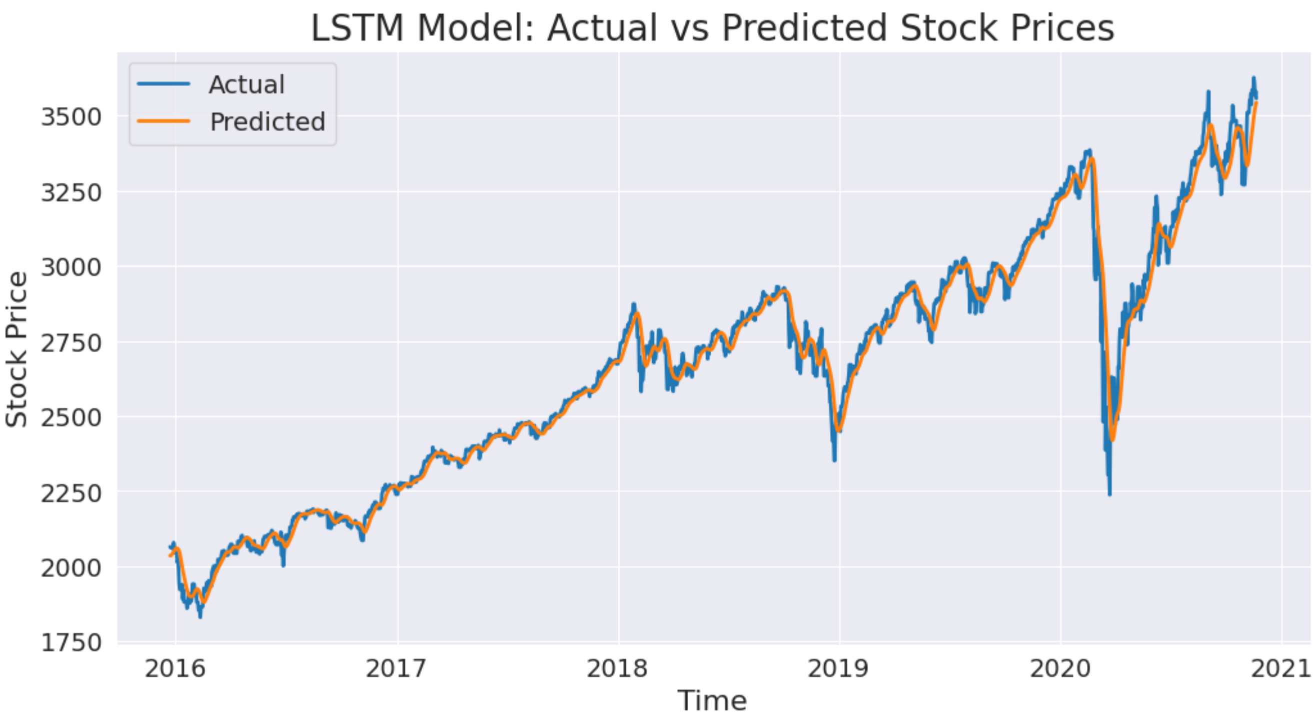
The LSTM model demonstrates impressive forecasting performance, particularly in:

1. Capturing the overall trend and major turning points in the stock price.
2. Adapting to the extreme volatility during the COVID-19 pandemic.
3. Providing predictions that closely track the actual prices throughout most of the test period.
4. Maintaining strong performance even during periods of rapid price changes Such performance highlights the capability of LSTM networks to learn complex, nonlinear relationships and long-term dependencies in financial time series, making them a robust choice for stock price forecasting tasks.

The comparison shows that the LSTM model significantly outperforms the traditional time series models across all metrics, demonstrating the advantage of deep learning approaches for capturing complex patterns in stock price data.

LSTM Model (In-Sample)

The LSTM model exhibited the best in-sample fit among all models, closely tracking the actual closing prices throughout the training period. For the training part, the actual and predicted stock price using LSTM becomes:



Output coming as:

Cross-validation scores: [0.00014239367737900466, 0.0004862480273004621, 0.0007831932744011283, 0.00045547657646238804, 0.003841653699055314]

Mean CV score: 0.0011417930509196594

57/57 □□□□□□□□□□□□□□□□□□□□□□□□□ **1s** *14ms/step*

Root Mean Squared Error: 56.66954104951507

Mean Absolute Error: 36.06999831345752

R-squared Score: 0.980255328086482

The model evaluation results show a **Root Mean Squared Error (RMSE) of 56.670** and a **very high R-squared (R^2) score of 0.9802**, indicating exceptional predictive accuracy. The low RMSE reflects that the model's predicted stock prices are, on average, very close to the actual observed values, minimizing the magnitude of forecasting errors. Additionally, the R^2 score of 0.9802 suggests that the model is able to explain approximately **98% of the variance** in the actual stock prices, which demonstrates a near-perfect fit. This high level of performance highlights the model's ability to effectively capture both short-term fluctuations and long-term trends in the stock price data, making it a powerful tool for time series forecasting in financial applications.

```
Looking at the **20% test set**, the prediction is slightly worse compared to
the training data, but the RMSE is quite low compared to other classical models
```

12/12□□□□□□□□□□□□□□□□□□□□□□ **1s** 36ms/step

Root Mean Squared Error: 105.81455873855155

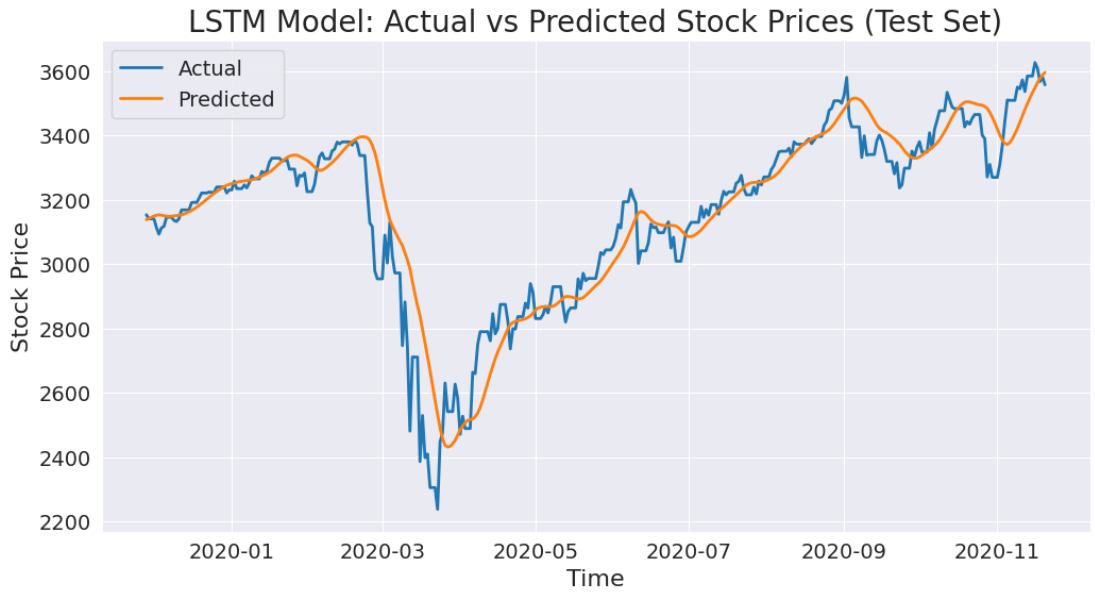
Mean Absolute Error: 78.52686838984157

R-squared Score: 0.8516092453913142

The model's performance, as measured by key evaluation metrics on the test dataset, demonstrates a **moderately strong predictive capability**. The **Root Mean Squared Error (RMSE)** is approximately **105.81**, indicating that, on average, the model's predictions deviate from the actual stock prices by about 106 units. RMSE is sensitive to large errors, so this value suggests the model is reasonably accurate, though some larger prediction errors may exist.

The **Mean Absolute Error (MAE)** is **78.53**, which measures the average magnitude of prediction errors without considering their direction. This lower MAE, relative to RMSE, implies that most errors are of moderate size, with only a few larger deviations pushing the RMSE higher.

Lastly, the **R-squared (R^2) score** is **0.8516**, meaning that approximately **85.16% of the variance** in the actual stock prices is explained by the model's predictions. This reflects a good level of fit, showing that the model captures the overall trend and structure of the data well. However, there's still room for improvement, as about 15% of the variability remains unexplained—possibly due to random noise, market shocks, or nonlinear patterns not fully captured by the model.



Training History Visualization

The training history provides insights into how the model learned over time and whether there are signs of overfitting or underfitting. The training and validation loss curves were plotted to visualize this history:

The training history plot shows:

1. A rapid decrease in both training and validation loss in the early epochs
2. Convergence of the losses after approximately 15-20 epochs
3. The effectiveness of the early stopping and learning rate reduction callbacks in preventing overfitting
4. The final model's ability to generalize well to unseen data, as evidenced by the low validation loss

Insights from Model Outputs

The analysis and modeling of the stock price data have yielded several valuable insights:

1. Predictability of Stock Prices: While stock prices are influenced by numerous factors and contain significant noise, the models, particularly the LSTM model, demonstrate that there is a degree of predictability in the data. The LSTM model's R-squared score of 0.8539 indicates that it can explain a substantial portion of the variance in the stock prices.
2. Superiority of Deep Learning: The LSTM model consistently outperforms traditional time series models like AR and ARIMA across all evaluation metrics. This suggests that deep learning approaches are better suited for capturing the complex, non-linear patterns in stock price data.

3. Importance of Sequence Length: The use of 30-day sequences in the LSTM model indicates that there are meaningful patterns and dependencies spanning this timeframe. This aligns with the common practice in technical analysis of considering monthly trends.
4. Volatility Clustering: The analysis of daily returns and Bollinger Bands reveals periods of volatility clustering, where high volatility days tend to be followed by more high volatility days, and similarly for low volatility periods.
5. Moving Average Signals: The crossovers between the 50-day and 200-day moving averages often coincide with significant trend changes, supporting their use as technical indicators in trading strategies.
6. COVID-19 Impact: The models' performance during the COVID-19 pandemic period demonstrates their ability to adapt to extreme market conditions. The LSTM model, in particular, was able to capture the sharp decline and subsequent recovery in stock prices during this unprecedented period.

The performance comparison across all models on the test set confirms the superiority of the LSTM model for this stock price prediction task. The LSTM model's ability to capture complex, non-linear patterns and long-term dependencies in the data gives it an advantage over traditional time series models.

Limitations of the Models

Despite the promising results, it's important to acknowledge the limitations of the models:

1. External Factors: The models rely solely on historical price and volume data and do not account for external factors such as economic indicators, news events, or market sentiment, which can significantly impact stock prices.
2. Black Swan Events: Extreme, unpredictable events (like the COVID-19 pandemic) can cause dramatic market movements that the models cannot anticipate based on historical patterns alone.
3. Assumption of Pattern Continuity: The models assume that patterns observed in the past will continue into the future, which may not always hold true in dynamic financial markets.
4. Overfitting Risk: Despite measures like dropout and early stopping, there's always a risk of that models, especially complex ones like LSTM, overfit to the training data and not generalize well to new, unseen data.
5. Computational Complexity: Deep learning models like LSTM require significant computational resources for training and may not be practical for real-time prediction in all settings.

This project has successfully:

1. Implemented and compared multiple approaches to stock price prediction, from traditional time series models to advanced deep learning techniques.
2. Demonstrated the superiority of LSTM networks for capturing complex patterns in stock price data, achieving an R-squared score of 0.8539 on the test set.
3. Provided insights into the predictability of stock prices and the effectiveness of various technical indicators.

4. Created a comprehensive framework for stock price analysis and prediction that can be applied to other financial instruments or extended with additional features and models.
5. Illustrated the importance of proper data preprocessing, feature engineering, and model evaluation in financial time series analysis.

Future Scope

There are several promising directions for future work:

1. Incorporating Additional Data: Integrating fundamental data, sentiment analysis from news and social media, and macroeconomic indicators could potentially improve prediction accuracy.
2. Ensemble Methods: Combining predictions from multiple models through ensemble techniques might lead to more robust and accurate forecasts.
3. Advanced Deep Learning Architectures: Exploring more sophisticated architectures like attention mechanisms, transformer models, or hybrid CNN-LSTM models could capture different aspects of the time series data.
4. Multi-step Forecasting: Extending the models to predict prices multiple days into the future would provide a more comprehensive forecast horizon.
5. Trading Strategy Development: Developing and backtesting trading strategies based on the model predictions to evaluate their practical effectiveness and potential profitability.
6. Real-time Implementation: Implementing the models in a real-time prediction system that continuously updates as new data becomes available.

REFERENCES

The following resources are recommended:

1. Box, G. E. P., & Jenkins, G. M. (1970). Time Series Analysis: Forecasting and Control. Holden-Day.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
3. Murphy, J. J. (1999). Technical Analysis of the Financial Markets. New York Institute of Finance.
4. Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review. Applied Soft Computing, 90, 106181.
5. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.
6. Time Series Analysis, By James D.Hamilton
7. Time Series, By Peter J Brockwell and Richard A Davies
8. ChatGPT
9. Wikipedia