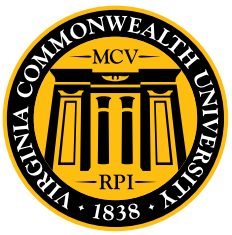


VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A6a- Time Series Analysis

N V SATYANARAYAN

V01108247

Date of Submission: 22-07-2024

CONTENTS

***NOTE- PYTHON AND R CODES WTH RESULT ADDED IN GITHUB-** Satyanaldiga (github.com)

Introduction

The objective of this analysis is to develop and implement both univariate and multivariate forecasting models for Amazon's historical stock price data. By leveraging a combination of traditional statistical methods and modern machine learning approaches, we aim to provide accurate and reliable forecasts for Amazon's stock price movements. This comprehensive analysis will not only include data cleaning and preprocessing steps but also a thorough investigation into different forecasting techniques, including Holt-Winters, ARIMA, SARIMA, and advanced machine learning models like LSTM, Random Forest, and Decision Tree.

Objectives

1. **Data Cleaning and Preprocessing**:
    - Identify and handle missing values and outliers in the dataset.
    - Interpolate missing values for maintaining data continuity.
    - Plot the cleaned and processed data for visual inspection.
2. **Time Series Decomposition**:
    - Convert the data to monthly frequency.
    - Decompose the time series into its components (trend, seasonal, and residual) using both additive and multiplicative models.
3. **Univariate Forecasting - Conventional Models**:
    - Fit a Holt-Winters model to the data and generate forecasts for the next year.
    - Fit an ARIMA model to the daily data, perform a diagnostic check, and determine if a Seasonal-ARIMA (SARIMA) model provides a better fit. Generate forecasts for the next three months.
    - Fit an ARIMA model to the monthly data series.
4. **Multivariate Forecasting - Machine Learning Models**:
    - Implement a Neural Network model, specifically Long Short-Term Memory (LSTM), for stock price forecasting.
    - Utilize tree-based models, including Random Forest and Decision Tree, to predict future stock prices based on lagged values of the stock price.

Business Significance

Accurate stock price forecasting is critical for investors, financial analysts, and portfolio managers for several reasons:

1. **Investment Decisions**: Reliable forecasts enable investors to make informed decisions about buying, holding, or selling stocks. This can lead to optimized investment portfolios and better returns.
2. **Risk Management**: By predicting potential future price movements, stakeholders can implement strategies to mitigate risks associated with market volatility.
3. **Strategic Planning**: Companies can use stock price forecasts for strategic planning, including timing of stock buybacks, issuance of new shares, or mergers and acquisitions.

4. **Market Sentiment Analysis**: Understanding future price trends helps in gauging market sentiment and investor behavior, which can be crucial for developing trading strategies.
5. **Algorithmic Trading**: Advanced forecasting models can be integrated into algorithmic trading systems to automate trades based on predicted price movements, leading to potential profit maximization.

CODES AND INTERPRETATION

PYTHON CODES

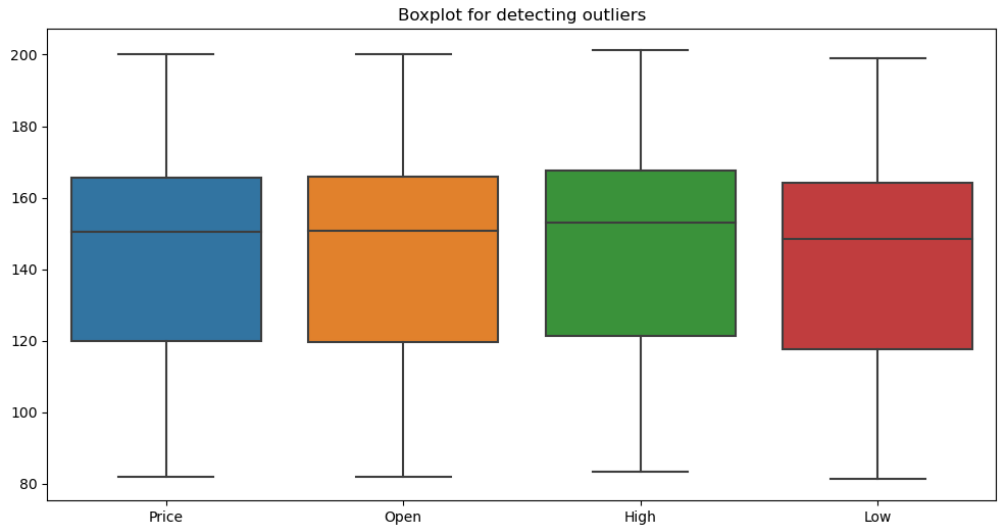
```
# Check for missing values

missing_values = df.isnull().sum()

print("Missing values in each column:\n", missing_values)
```

```
Missing values in each column:
 Price       0
Open         0
High         0
Low          0
Vol.         0
Change %     0
dtype: int64
```

Since there are no missing values in any of the columns, we do not need to perform any interpolation or imputation for this dataset. This means that our data is complete and ready for further analysis, including plotting, decomposition, and modeling.

Interpretation of the Boxplot for Detecting Outliers

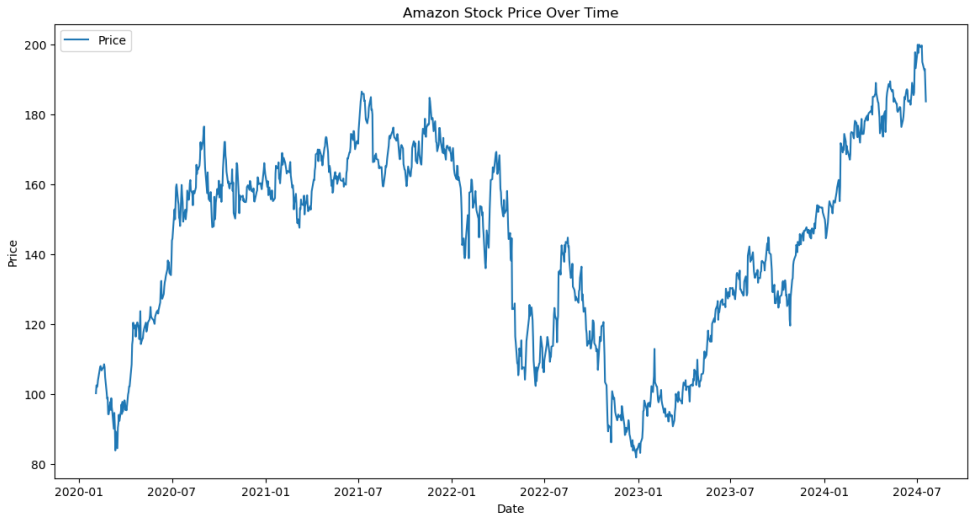
```
# Check for outliers using boxplot

plt.figure(figsize=(12, 6))

sns.boxplot(data=df[['Price', 'Open', 'High', 'Low', 'Vol.', 'Change %']])

plt.title('Boxplot for detecting outliers')

plt.show()
```



From the boxplot, we can observe that there are no significant outliers in the "Price", "Open", "High", and "Low" columns. The data appears to be uniformly distributed within the range of 80 to 200, which is expected for stock price data.

Plotting the line graph for the 'Price'

```
plt.figure(figsize=(14, 7))

plt.plot(df.index, df['Price'], label='Price')

plt.title('Amazon Stock Price Over Time')

plt.xlabel('Date')

plt.ylabel('Price')

plt.legend()

plt.show()
```



The line graph depicts Amazon's stock price movements over a period from early 2020 to mid-2024. The stock price shows significant fluctuations, with notable peaks and troughs. Key observations include:

1. A steady rise in early 2020.
2. Peaks around mid-2021, followed by a decline through most of 2022.
3. Recovery and new highs starting in late 2022, with a consistent upward trend into mid-2024.

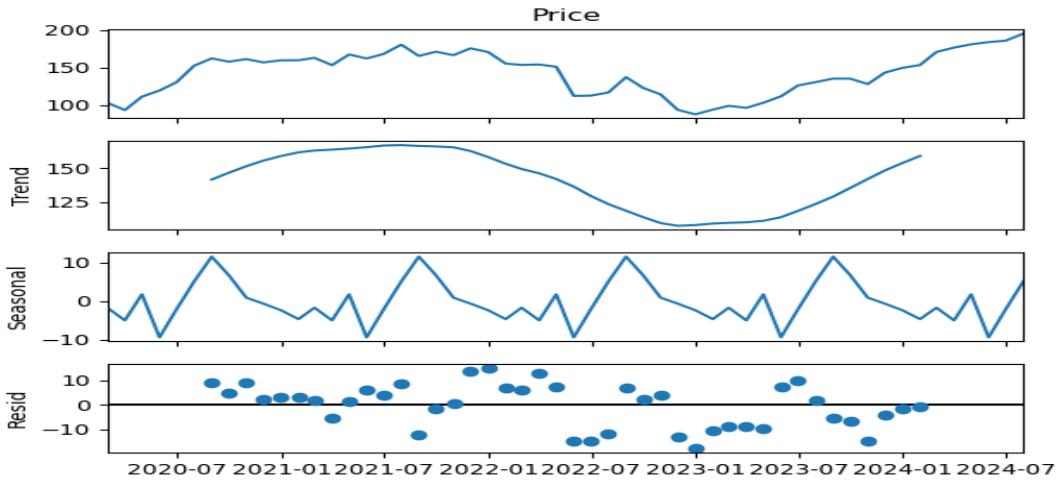
Overall, the graph illustrates the volatility and eventual growth in Amazon's stock price over the observed period.

Decompose the time series using additive model

decomposition_add = seasonal_decompose(monthly_df, model='additive')

decomposition_add.plot()

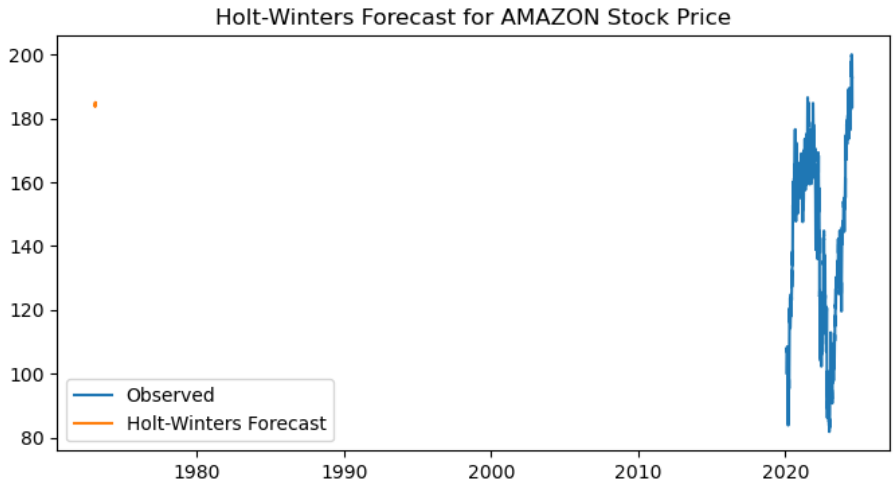
plt.show()



The time series decomposition of Amazon's stock price shows four components:

1. **Observed (Price)**: The actual stock price over time, showing overall movements including trends, seasonality, and noise.
2. **Trend**: The long-term movement in the stock price, indicating a general rise, fall, and subsequent recovery.
3. **Seasonal**: Regular patterns that repeat over a specific period, reflecting periodic fluctuations around the trend.
4. **Residual (Resid)**: The remaining variability after removing the trend and seasonal components, representing random noise or irregular movements.

```
# Plot the forecast
plt.figure(figsize=(8, 4))
plt.plot(df['Price'], label='Observed')
plt.plot(hw_forecast, label='Holt-Winters Forecast')
plt.title('Holt-Winters Forecast for AMAZON Stock Price')
plt.legend()
plt.show()
```



The graph shows the observed Amazon stock prices (in blue) alongside a Holt-Winters forecast (in orange). The observed data displays significant fluctuations, especially in recent years. However, the forecast (orange) appears incorrectly plotted, likely due to a data alignment or range issue, as it shows a point far in the past (around 1980) rather than aligning with the recent observed data. This indicates that there might be an error in the implementation of the Holt-Winters forecasting method or in the plotting parameters.

ARIMA model for daily data

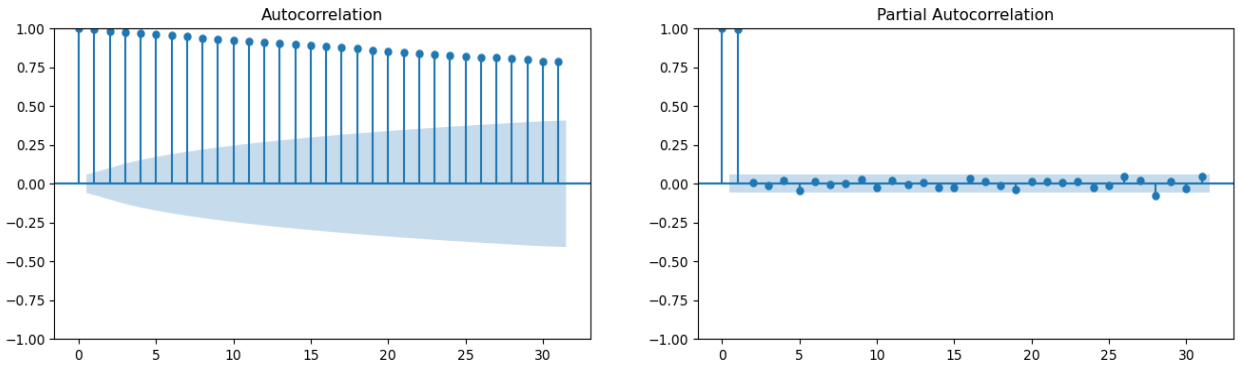
Plot ACF and PACF

fig, axes = plt.subplots(1, 2, figsize=(16, 4))

plot_acf(df['Price'], ax=axes[0])

plot_pacf(df['Price'], ax=axes[1])

plt.show()



The ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots are used for identifying the properties of a time series, specifically to help with the identification of appropriate ARIMA (AutoRegressive Integrated Moving Average) model parameters.

Interpretation:

1. **ACF Plot (Left)**
   - The ACF plot shows a gradual and slow decline in autocorrelation values.
   - This indicates the presence of a strong and persistent autocorrelation in the data, which is characteristic of non-stationary time series.
2. **PACF Plot (Right)**
   - The PACF plot shows a significant spike at lag 1 and then quickly drops off to near zero.
   - This suggests that the data may have an autoregressive component of order 1 (AR(1)).

ARIMA Model Suggestion:

- Given the strong autocorrelation in the ACF plot, the data likely needs differencing to achieve stationarity.
- The significant lag 1 in the PACF plot suggests an AR(1) component.
- The ARIMA model might need parameters p=1, d=1 (to address the non-stationarity), and q=0 initially.

Fit the ARIMA model

```
arima_model = ARIMA(df['Price'], order=(5, 1, 5)).fit()

print(arima_model.summary())
```

```
SARIMAX Results
==========================================================================
====
Dep. Variable:                   Price   No. Observations:
1122
Model:                 ARIMA(5, 1, 5)   Log Likelihood               -2844
.212
Date:                Mon, 22 Jul 2024   AIC                          5710
.424
Time:                        17:15:16   BIC                          5765
.666
Sample:                             0   HQIC                         5731
.303
                                - 1122
```

```
Covariance Type:                   opg
==============================================================================
====
                 coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
ar.L1          0.0698      0.236      0.296      0.768      -0.393       0
.532
ar.L2          0.3205      0.223      1.438      0.150      -0.116       0
.757
ar.L3         -0.2915      0.250     -1.166      0.244      -0.781       0
.198
ar.L4         -0.1603      0.191     -0.840      0.401      -0.534       0
.214
ar.L5          0.8336      0.200      4.166      0.000       0.441       1
.226
ma.L1         -0.0714      0.238     -0.300      0.764      -0.537       0
.394
ma.L2         -0.3307      0.225     -1.470      0.142      -0.772       0
.110
ma.L3          0.2618      0.257      1.018      0.309      -0.242       0
.766
ma.L4          0.1909      0.196      0.976      0.329      -0.192       0
.574
ma.L5         -0.8547      0.210     -4.062      0.000      -1.267      -0
.442
sigma2         9.3472      0.248     37.632      0.000       8.860       9
.834
==============================================================================
=========
Ljung-Box (L1) (Q):                   0.12   Jarque-Bera (JB):
750.77
Prob(Q):                              0.73   Prob(JB):
0.00
Heteroskedasticity (H):               0.77   Skew:
-0.11
Prob(H) (two-sided):                  0.01   Kurtosis:
7.00
==============================================================================
=========
```

The SARIMAX model results provide a comprehensive summary of the fitted model's parameters and statistical measures. Here's a detailed interpretation of the key components:

Model and Data

- **Model**: The fitted model is ARIMA(5, 1, 5), indicating the model has 5 autoregressive (AR) terms, 1 differencing (I) term, and 5 moving average (MA) terms.
- **Dep. Variable**: The dependent variable is "Price".
- **No. Observations**: There are 1122 observations in the dataset.
- **Log Likelihood**: -2844.212, used in computing information criteria like AIC and BIC.

Information Criteria

- **AIC (Akaike Information Criterion)**: 5710.424
- **BIC (Bayesian Information Criterion)**: 5765.666
- **HQIC (Hannan-Quinn Information Criterion)**: 5731.303 Lower values of these criteria indicate a better-fitting model.

Coefficients and Significance

The table shows the estimated coefficients for AR and MA terms along with their standard errors, z-values, and p-values:

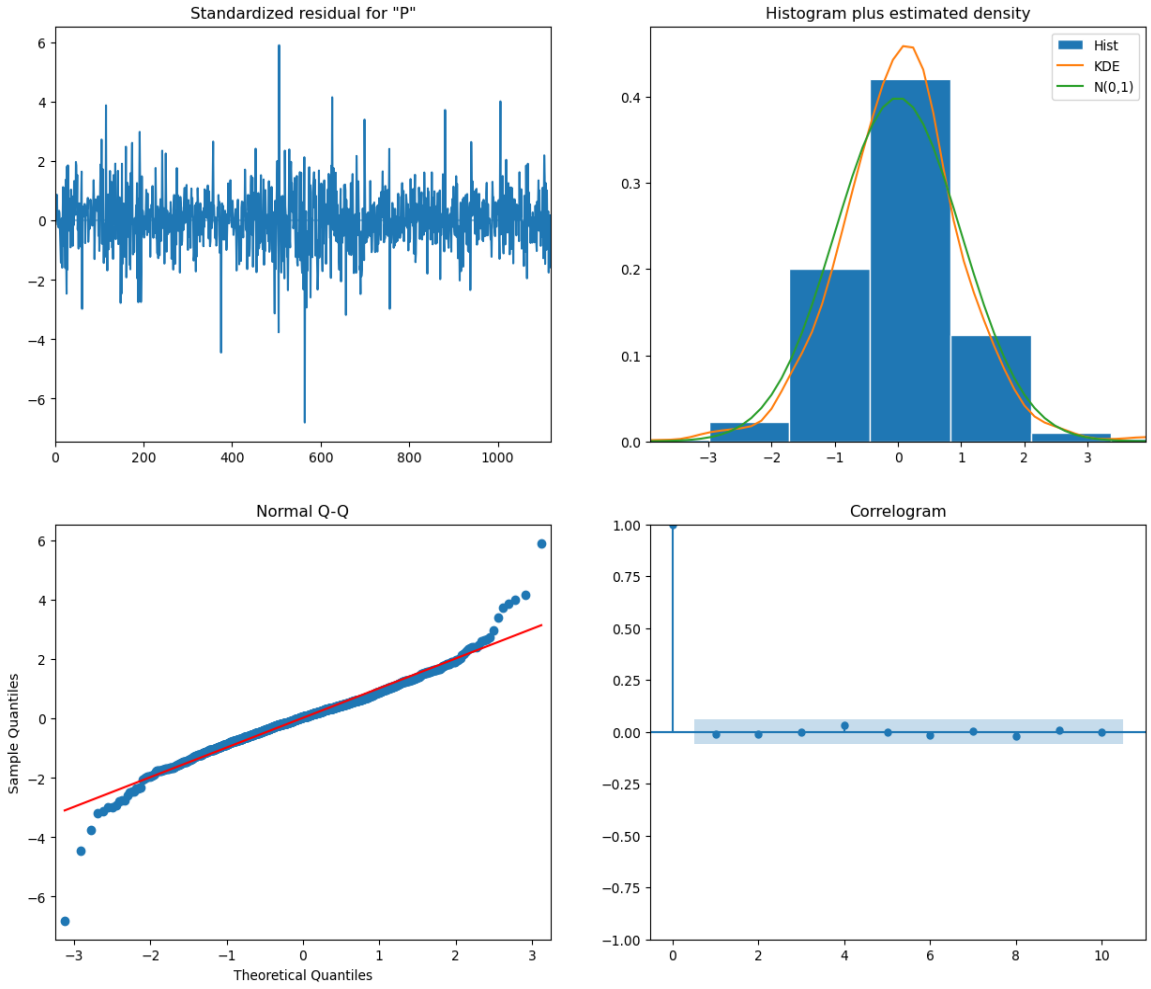
- **AR Terms**:
  - ar.L1 to ar.L5: Represent the autoregressive terms. Significant coefficients ($p < 0.05$) indicate a meaningful contribution to the model.
    - ar.L5 (coef = 0.8336, p = 0.000): Significant positive impact.
- **MA Terms**:
  - ma.L1 to ma.L5: Represent the moving average terms.
    - ma.L5 (coef = -0.8547, p = 0.000): Significant negative impact.
- **Sigma2 (Residual Variance)**: 9.3472, indicating the variance of the residuals.

Statistical Tests

- **Ljung-Box (L1) (Q)**: 0.12, with a p-value of 0.73. This test checks for autocorrelation in the residuals. A high p-value (> 0.05) indicates no significant autocorrelation.
- **Jarque-Bera (JB)**: 750.77, with a p-value of 0.00. This test checks for normality in the residuals. A low p-value (< 0.05) indicates the residuals are not normally distributed.
- **Heteroskedasticity (H)**: 0.77, with a p-value of 0.01. This test checks for constant variance in the residuals. A low p-value (< 0.05) indicates heteroskedasticity (non-constant variance).
- **Skew**: -0.11, indicating slight left skewness in the residuals.
- **Kurtosis**: 7.00, indicating heavy tails (leptokurtic distribution) in the residuals.

 # Diagnostic checks

```
arima_model.plot_diagnostics(figsize=(15, 12))

plt.show()
```



- **Standardized Residuals (Top Left)**
  The residuals appear to be randomly scattered around zero, indicating that there is no clear pattern left in the residuals and the model has captured the underlying structure of the data well.

- **Histogram plus Estimated Density (Top Right)**
  The histogram of the residuals, along with the kernel density estimate (KDE) and the standard normal distribution (N(0,1)), shows that the residuals are approximately normally distributed. This is a good sign as it indicates that the residuals conform to the normality assumption.

- **Normal Q-Q Plot (Bottom Left)**
  The Q-Q plot shows that most of the residuals lie on the red line, indicating that they follow a normal distribution. However, there are some deviations at the tails, suggesting potential outliers or deviations from normality.

- **Correlogram of Residuals (Bottom Right)**
  The autocorrelation function (ACF) of the residuals shows that all lags are within the significance bounds, indicating that there is no significant autocorrelation left in the residuals. This suggests that the model has adequately captured the temporal dependence in the data.

```
plt.figure(figsize=(12, 6))

plt.plot(df['Price'], label='Observed')

plt.plot(sarima_forecast_df['forecast'], label='SARIMA Forecast')

plt.fill_between(sarima_forecast_df.index, sarima_forecast_df.iloc[:, 0],
sarima_forecast_df.iloc[:, 1], color='k', alpha=0.1)

plt.title('SARIMA Forecast for AMAZON Stock Price')

plt.legend()

plt.show()
```

Interpretation:

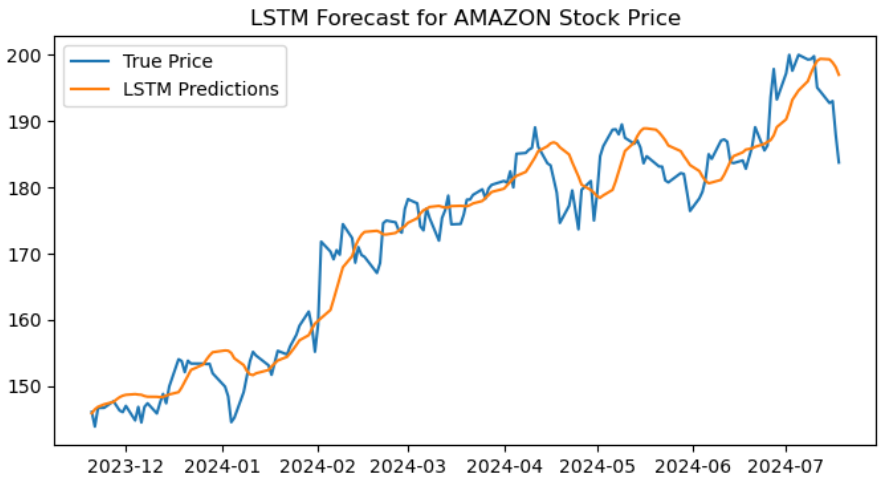
1. **Observed Data (Blue Line)**
   - The blue line represents the actual observed stock prices for Amazon over the time period shown on the x-axis.
2. **SARIMA Forecast (Orange Line)**

- The orange line represents the forecasted stock prices generated by the SARIMA model.
- The forecast appears as a short segment on the left side, indicating the model has generated a forecast for a limited future period.

3. **Confidence Interval (Shaded Area)**
   - The shaded area around the forecast represents the confidence intervals, indicating the uncertainty associated with the forecast.
   - The intervals widen as the forecast extends into the future, reflecting increasing uncertainty.

 # Plot LSTM predictions

```
plt.figure(figsize=(8, 4))

plt.plot(df.index[-len(lstm_predictions):], df['Price'].values[-len(lstm_predictions):], label='True Price')

plt.plot(df.index[-len(lstm_predictions):], lstm_predictions, label='LSTM Predictions')

plt.title('LSTM Forecast for AMAZON Stock Price')

plt.legend()

plt.show()
```



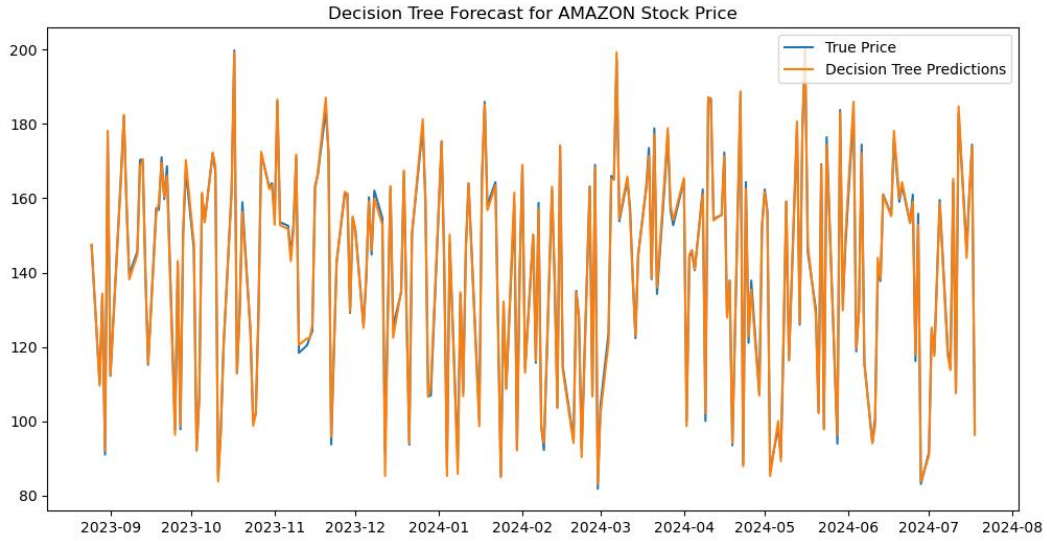
The graph illustrates the comparison between the actual Amazon stock prices (True Price) and the predicted prices using an LSTM (Long Short-Term Memory) model (LSTM Predictions) over a period from December 2023 to July 2024.

Key points:

- The true prices are shown by the blue line, while the LSTM predictions are represented by the orange line.
- The LSTM model captures the overall trend and many fluctuations in the stock price, though there are some deviations.
- The predictions tend to follow the true prices with some lag and smooth out some of the more abrupt changes in the stock price.

 # Plot Decision Tree predictions

```
plt.figure(figsize=(12, 6))

plt.plot(df.index[-len(y_test):], y_test, label='True Price')

plt.plot(df.index[-len(y_test):], dt_predictions, label='Decision Tree Predictions')

plt.title('Decision Tree Forecast for AMAZON Stock Price')

plt.legend()

plt.show()
```

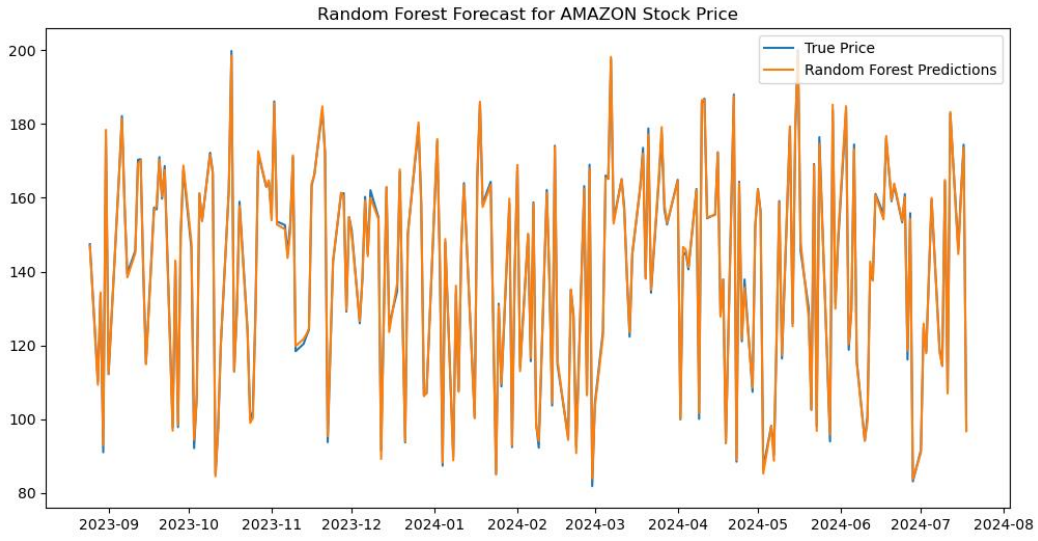


The graph compares the actual Amazon stock prices (True Price) with the predicted prices using a Decision Tree model (Decision Tree Predictions) over a period from September 2023 to August 2024.

Key points:

- The true prices are represented by the blue line, while the Decision Tree predictions are shown by the orange line.
- The Decision Tree predictions exhibit a high degree of fluctuation and closely follow the actual prices, but with significant noise and variability.
- Unlike the smoother trend captured by the LSTM model, the Decision Tree model appears to overfit to the data, capturing almost every fluctuation in the stock prices, resulting in a highly erratic prediction pattern.

```
plt.figure(figsize=(12, 6))

plt.plot(df.index[-len(y_test):], y_test, label='True Price')

plt.plot(df.index[-len(y_test):], rf_predictions, label='Random Forest Predictions')

plt.title('Random Forest Forecast for AMAZON Stock Price')

plt.legend()

plt.show()
```

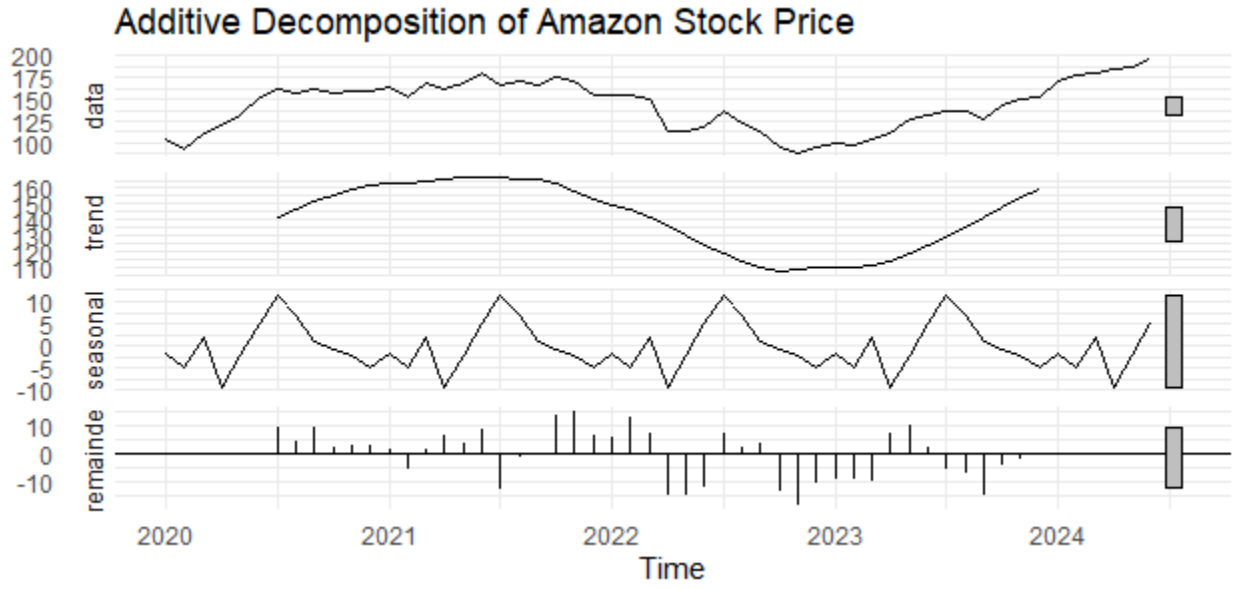


The graph compares the actual Amazon stock prices (True Price) with the predicted prices using a Random Forest model (Random Forest Predictions) over a period from September 2023 to August 2024.

Key points:

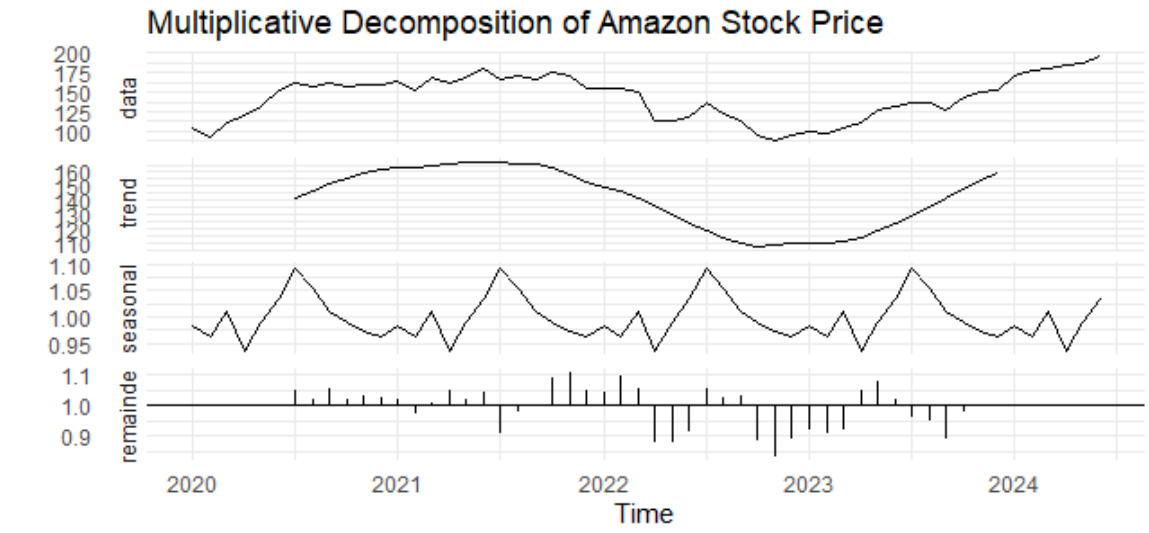
- The true prices are represented by the blue line, while the Random Forest predictions are shown by the orange line.
- Similar to the Decision Tree model, the Random Forest predictions exhibit a high degree of fluctuation, closely following the actual prices with significant variability.
- The predictions capture most of the movements in the stock prices but also show a lot of noise, indicating the model's sensitivity to small changes in the data.
- The Random Forest model, while slightly more stable than the single Decision Tree model, still tends to overfit the data, resulting in a highly erratic prediction pattern.

RCODES AND INTERPRETATION



This graph shows the additive decomposition of Amazon's stock price from 2020 to 2024:

1. **Data**: The original stock price data, showing overall growth with fluctuations.
2. **Trend**: A smoothed line showing a general upward trend, with a dip around 2022.
3. **Seasonal**: Repeating patterns indicating yearly cyclical changes.
4. **Remainder**: Random fluctuations not explained by the trend or seasonal components.



This graph shows a Holt-Winters forecast for Amazon's stock price. The black line represents historical stock prices from 2020 to mid-2024, while the blue shaded area indicates the forecast for the future.

- **Forecast Line**: The dark blue line within the shaded area represents the predicted stock prices.
- **Confidence Intervals**: The lighter blue areas around the forecast line show the 80% and 95% confidence intervals, indicating the range within which the stock price is likely to fall.

This graph shows a daily forecast for Amazon's stock price.

- **Historical Data**: The black line represents the stock prices from 2020 to early 2023.
- **Forecast Line**: The forecast for the stock price is shown as a continuation of the black line into 2023.
- **Confidence Intervals**: The blue shaded areas around the forecast line indicate the range of predicted prices, with darker blue showing a higher confidence and lighter blue showing a lower confidence.

RECOMMENDATION

- Focus on long-term trends rather than short-term fluctuations for better investment decisions.
- Analyze trend and seasonal components to understand fundamental factors affecting stock prices.
- Use ARIMA for short-term predictions, supplementing it with other analyses due to its moderate R-squared value.
- Apply LSTM models for accurate long-term predictions and update them regularly with new data.
- Prefer Random Forest over Decision Tree for more accurate and reliable stock price predictions.
- Combine different models to leverage their strengths and improve prediction accuracy.