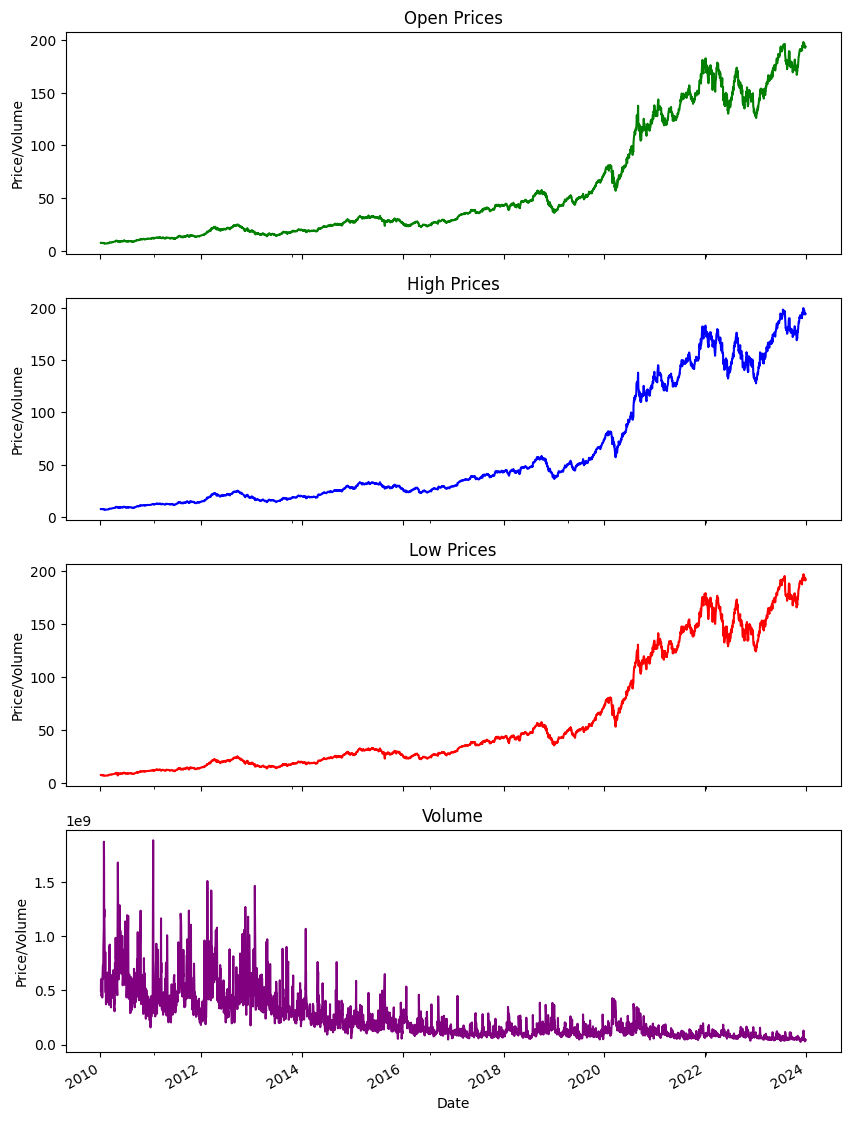
**Apple Stock Price Time Series Analysis and Forecasting**

Files- single\_stock\_time\_series.ipynb - python notebook file, Documentation

1. Necessary Libraries imported:

* pandas (imported as pd): Used for data manipulation and analysis.
* matplotlib.pyplot (imported as plt): Used for data visualization and plotting.
* yfinance (imported as yf): Used to download historical price data for stocks from Yahoo Finance.
* seasonal\_decompose from statsmodels.tsa.seasonal: Used to decompose time series data into its components.
* adfuller and kpss from statsmodels.tsa.stattools: Used for stationarity testing.
* auto\_arima from pmdarima: Used for automatically selecting the best ARIMA model parameters.
* ARIMA from statsmodels.tsa.arima.model: Used for building the ARIMA model.
* mean\_squared\_error from sklearn.metrics: Used to calculate the mean squared error for model evaluation.
* numpy (imported as np): Used for numerical computing.
* warnings: Used to handle warnings during the execution of the code.
* seaborn (imported as sns): Used for statistical data visualization.

1. Data Preparation and Exploration:

* Overview of Data: The historical price data for Apple Inc. (AAPL) from January 1, 2010, to January 1, 2024, was downloaded using the Yahoo Finance API (yfinance). The dataset consists of six columns: Open, High, Low, Close, Adj Close, and Volume, totaling 3522 entries. There are no missing values in the dataset, ensuring data completeness.
* Exploratory Detailed Analysis: Exploratory data analysis involved visualizing the Open, High, Low, and Volume data over time using line plots. First, inspection of the data was done by checking the first 5 rows of the dataset. From the summary statistics, historical price data was derived such as the mean opening price is approximately 59.88, the mean closing price is approximately 59.93, the mean volume traded is approximately 2.42 billion etc. Similarly, the standard deviation measures the dispersion or spread of the data points around the mean. For example: the standard deviation of the closing price is approximately 55.44, indicating relatively high volatility in the closing prices, the standard deviation of the volume traded is approximately 220.46 million, indicating significant variability in trading volume. Then comes the minimum and maximum closing price like minimum closing price observed is 6.86, while the maximum is approximately 1.88 billion. Quartiles provide information about the distribution of the data. The 25th percentile (first quartile) of the closing price is approximately 19.61, indicating that 25% of the closing prices fall below this value. The median (50th percentile) of the closing price is approximately 32.34, indicating that 50% of the closing prices fall below this value. The 75th percentile (third quartile) of the closing price is approximately 91.21, indicating that 75% of the closing prices fall below this value. Visualization of other datas like High, Low, Open, Volume over year -  
    
  

No missing values were found in the dataset. Additionally, a correlation matrix was generated to examine the relationships between different variables. Due to version issues, the coorelation matrix image is not showing perfectly in the code.

Correlation Matrix:

Open High Low Close Adj Close Volume

Open 1.000000 0.999907 0.999887 0.999767 0.999716 -0.529344

High 0.999907 1.000000 0.999861 0.999888 0.999832 -0.528420

Low 0.999887 0.999861 1.000000 0.999893 0.999851 -0.530831

Close 0.999767 0.999888 0.999893 1.000000 0.999950 -0.529698

Adj Close 0.999716 0.999832 0.999851 0.999950 1.000000 -0.528539

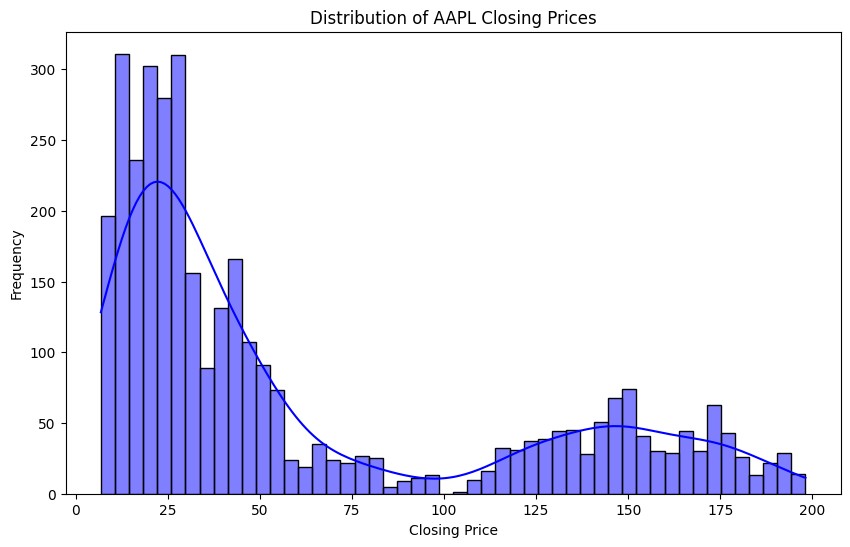
Volume -0.529344 -0.528420 -0.530831 -0.529698 -0.528539 1.000000

**Strength of Correlation**: The values in the correlation matrix range from -1 to 1. A value closer to 1 indicates a strong positive correlation, while a value closer to -1 indicates a strong negative correlation. Values close to 0 suggest a weak or no linear correlation.

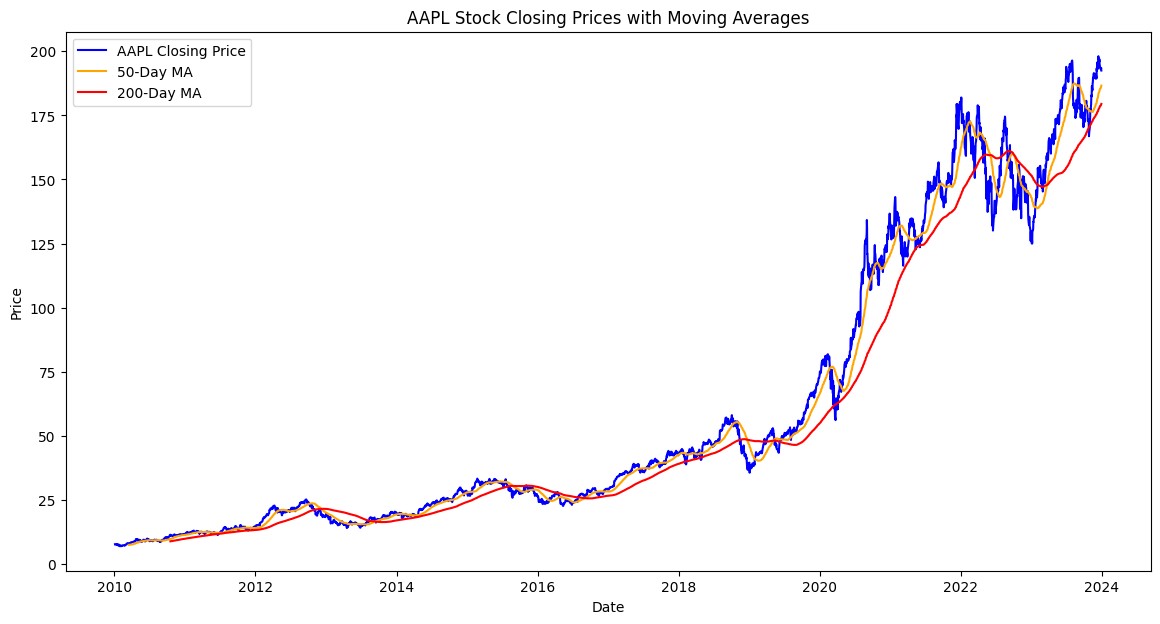
**Correlation between Variables**:

* **Open, High, Low, Close, and Adj Close**: These variables show a very high positive correlation with each other, with correlation coefficients close to 1. This indicates that these variables move almost in tandem, which is expected since they are all related to stock prices.
* **Volume**: The correlation between volume and the other variables is negative, indicating an inverse relationship. This suggests that as stock prices increase, trading volume tends to decrease, and vice versa.

**Interpretation**: The high positive correlation among Open, High, Low, Close, and Adj Close suggests that they move together and provide similar information about the stock's performance. The negative correlation between Volume and the other variables implies that changes in stock prices are not necessarily accompanied by changes in trading volume, or vice versa.

Next, we have the distribution of the AAPL closing price:

Plotting moving averages to smooth out the time series and identify trends-



Now, since we don’t have any missing columns. We apply dropna() function to check if there are still missing values left. Also we convert dataframe into a period index with a daily frequency (‘D’). This step ensures that the index represents dates in a structured format, which is essential for time series analysis and plotting.

1. Time Series Decomposition:

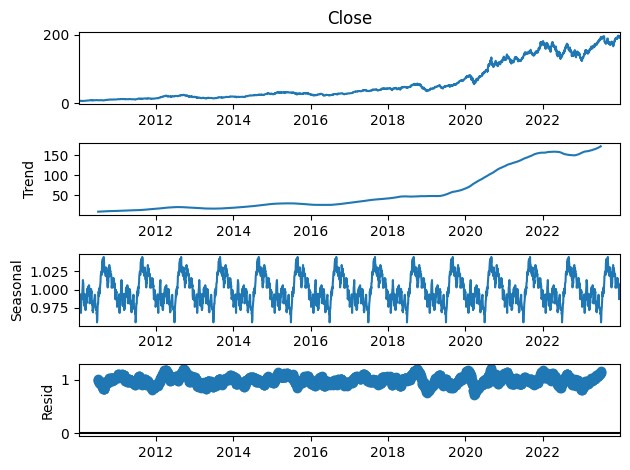
**Observed Component**: The observed component represents the original time series data, which is the actual closing prices of the Apple stock. Each value in this component corresponds to the actual observed closing price on a specific date.

**Trend Component**: The trend component captures the long-term movement or direction of the time series, smoothing out short-term fluctuations. In the provided output, the trend component is shown to be missing (NaN) for the first few dates. This is likely due to the method used for trend estimation.

**Seasonal Component**: The seasonal component represents the repetitive, periodic patterns or fluctuations in the time series data that occur at fixed intervals (e.g., daily, weekly, or yearly). Each value in this component represents the seasonal effect observed on a specific date.

**Residual Component**: The residual component, also known as the irregular component, captures the random fluctuations or noise that cannot be explained by the trend or seasonal components. It represents the difference between the observed values and the values predicted by the trend and seasonal components. In the provided output, the residual component is shown to be missing (NaN) for the first few dates.

Overall, time series decomposition allows us to break down a time series into its underlying components, making it easier to analyze and understand the various patterns and trends present in the data.



1. Stationarity Testing:

**Augmented Dickey-Fuller (ADF) Test**:

* + ADF Statistic: 0.8198709204450109
  + p-value: 0.9919502443702507
  + The ADF test statistic is 0.8198709204450109, and the corresponding p-value is 0.9919502443702507.
  + The null hypothesis of the ADF test is that the time series has a unit root, indicating it is non-stationary.
  + In this case, since the p-value (0.9919502443702507) is greater than 0.05 (common significance level), we fail to reject the null hypothesis. This suggests that the time series is non-stationary.

**Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test**:

* + KPSS Statistic: 7.468582417538178
  + p-value: 0.01
  + The KPSS test statistic is 7.468582417538178, and the corresponding p-value is 0.01.
  + The null hypothesis of the KPSS test is that the time series is trend-stationary.
  + In this case, since the p-value (0.01) is less than 0.05, we reject the null hypothesis. This suggests that the time series is non-stationary.

Based on the results of both tests, we conclude that the original time series (closing prices of Apple stock) is non-stationary. Additionally, the code snippet checks if the time series is non-stationary based on the ADF and KPSS test results. If either the ADF test's p-value is greater than 0.05 or the KPSS test's p-value is less than 0.05, differencing is applied to the time series. The differenced series is then tested for stationarity using the ADF test again.

The results after differencing are as follows:

* ADF Statistic (Differenced): -13.201987933901483
* p-value (Differenced): 1.0897495574254808e-24

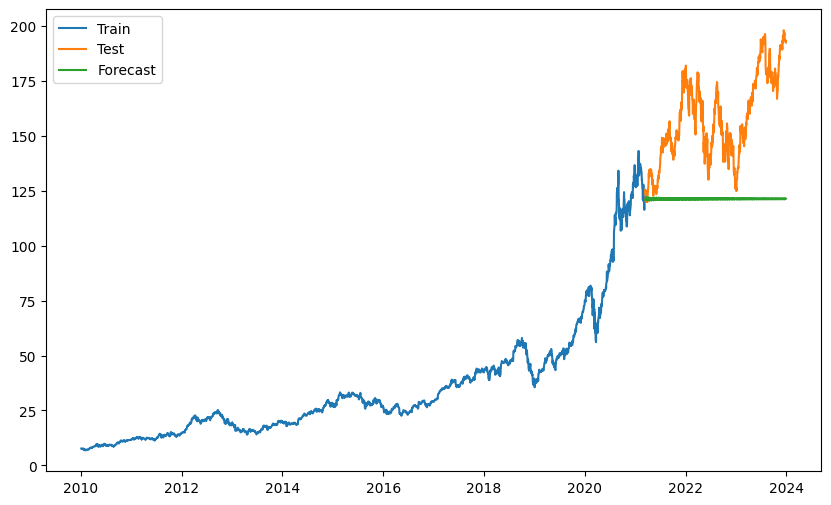
With the significantly low p-value (close to 0), we reject the null hypothesis, indicating that the differenced time series is stationary.

1. Forecasting Model Development:

* Data Splitting:

The time series data (closing price of Apple Stock) is divided into training sets and test sets using an 80 – 20 split. 80% of the data is used for training and 20% is used for testing.

* Grid Search for Best ARIMA model parameters: A nested loop is used to iterate over different combinations of p, d, and q parameters for the ARIMA model. The range of values for p, d, and q is defined as (0 to 5) for both p and q, and (0 to 1) for d. Inside the loop, an attempt is made to fit an ARIMA model to the training data using the current combination of parameters (p, d, q). If the model fitting is successful without any errors, the Akaike Information Criterion (AIC) value of the fitted model is calculated. A lower AIC value indicates a better fit of the model to the data. The best AIC value and corresponding parameters (p, d, q) are updated if the current model has a lower AIC value than the previously best-performing model. If an error occurs during model fitting (e.g., due to invalid parameter values), the error message is printed, and the loop continues to the next parameter combination. Once all combinations have been tested, the loop outputs the best ARIMA order (p, d, q) and the corresponding best AIC value.
* Forecasting: If a suitable ARIMA model is found during the grid search, we proceed to make forecasts using the model. In this case, we got –   
  order – (4,1,5); Best AIC: 7709.21739. Basically, we used autoARIMA, so that it will find the best parameters itself. We then plot the actual closing prices from the test set along with the forecasted values and the closing prices from the training set.



Then calculation of the RMSE is done. In this case, the RMSE was 41.27645910944367. The calculated RMSE value is printed to evaluate the accuracy of the forecasting model. A lower RMSE indicates better performance.

**Data:** The image shows three sets of closing prices:

* **Test Set (green line):** These are the actual closing prices that the ARIMA model was trying to predict.
* **Forecasted Values (likely a red or blue line):** This line represents the values that the ARIMA model predicted for the closing prices in the test set.
* **Training Set (not visible):** This data was used to train the ARIMA model. It includes historical closing prices that the model analyzed to identify patterns and trends.
* A close match between the lines indicates good forecasting performance. If the forecasted line closely aligns with the actual closing prices (green line), it suggests the ARIMA model effectively captured the underlying trends and patterns in the data. This indicates the model can be a reliable tool for short-term predictions within the range of the test set.