TIME SERIES FORECATING

BY

B G HUSSAIN

Contents

Problem Definition

In the volatile and complex environment of financial markets, forecasting the future prices of precious metals—particularly gold—is a challenging yet essential task for investors, traders, and financial institutions. Gold prices are influenced by a multitude of factors such as geopolitical events, currency fluctuations, inflation rates, and market speculation, making them inherently difficult to predict.

Given the critical role that gold plays as both an investment asset and a hedge against market uncertainty, there is a pressing need for a reliable and accurate forecasting model. The ability to anticipate future price trends can provide stakeholders with a strategic advantage, enabling better investment planning and risk management.

This project aims to address this challenge by developing a robust time series forecasting model that utilizes historical data of gold and silver prices. By leveraging statistical and machine learning models—including ARIMA, Holt-Winters (Triple Exponential Smoothing), and others—the goal is to generate accurate predictions of future gold prices. The resulting model is expected to serve as a powerful decision-support tool, offering data-driven insights into market trends and helping stakeholders navigate the uncertainties of the gold market with greater confidence.

Data Overview

- The dataset contains **2 columns**:
    - Date (datetime format)
    - Price (float format, representing gold prices)
- Total number of records: **2,539**

Initial Data Preview

First 5 Records (df.head()):

Date	Price
2023-08-17	1915.2
2023-08-16	1928.3
2023-08-15	1935.2
2023-08-14	1944.0
2023-08-11	NaN

Last 5 Records (df.tail()):

Date	Price
2013-08-23	NaN
2013-08-22	1370.8
2013-08-21	1370.1
2013-08-20	1372.6
2013-08-19	1365.7

Dataset Summary

Shape of the dataset: (2539, 2) — meaning 2,539 rows and 2 columns.

Data types:

Date: datetime64

Price: float64

Statistical Summary of Price Column

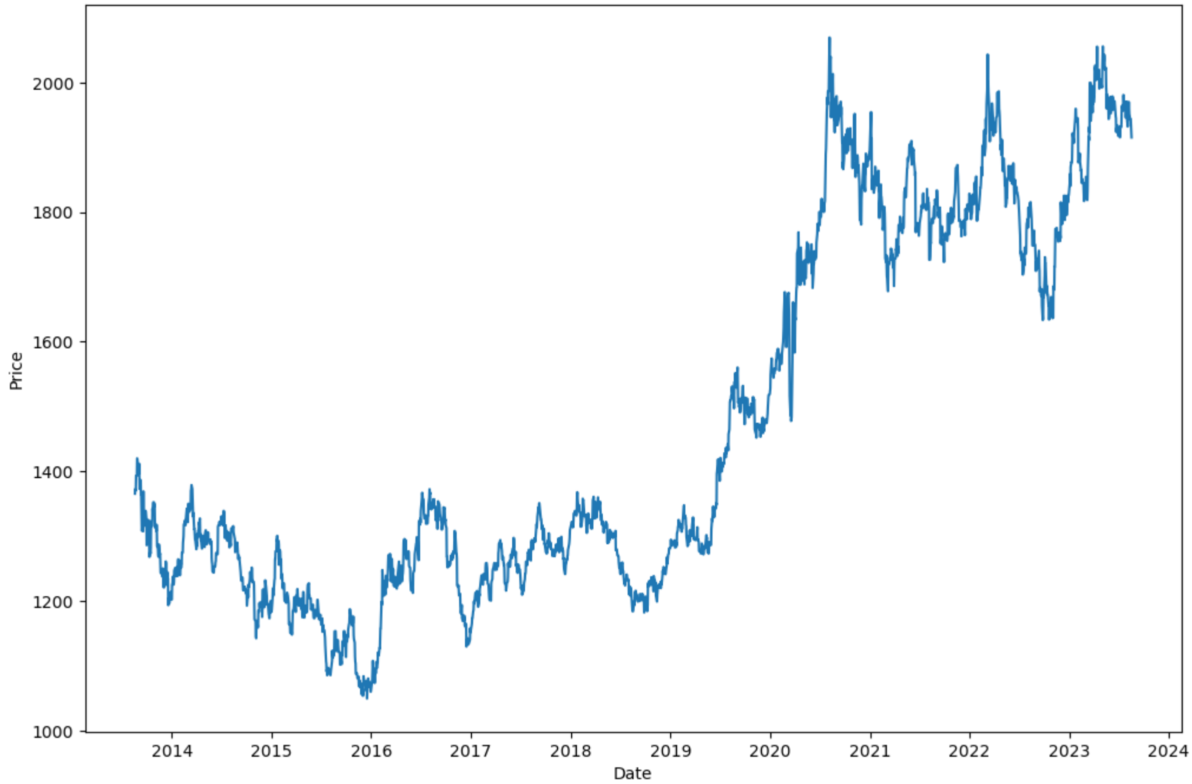
Statistic	Value
Count	2,460
Mean	1468.78
Std Dev	282.98
Min	1049.60
25%	1244.38
Median	1322.15
75%	1775.03
Max	2069.40

Missing Values Analysis

Missing values in Price: 79

Missing values in Date: 0

Exploratory data analysis

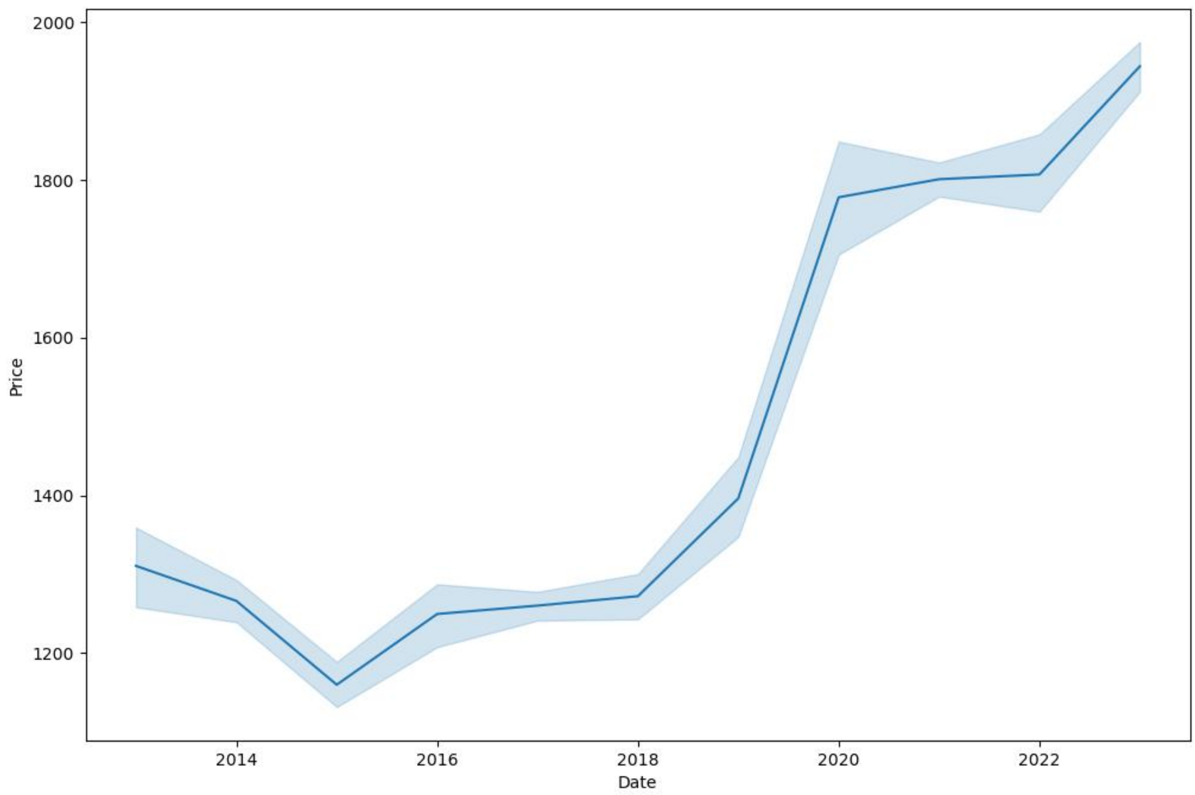


Price Trend Visualization

The provided image displays a line plot illustrating the trend of the 'Price' over time. The x-axis represents the 'Date', spanning from approximately early 2014 to late 2023. The y-axis represents the 'Price', ranging from around 1000 to over 2000.

Observations:

- Overall Upward Trend: The most prominent feature of the plot is a general upward trend in the price over the entire period. While there are fluctuations and periods of sideways movement, the price level in late 2023 is significantly higher than in early 2014.
- Volatility: The price exhibits considerable volatility throughout the timeframe. There are noticeable periods of sharp increases and decreases, indicating market fluctuations.
- Significant Dip in 2015-2016: Between roughly 2015 and early 2016, there is a significant downward dip in the price, followed by a recovery.
- Strong Growth Post-2019: Starting around 2019, the price shows a period of strong and relatively consistent growth, with some pullbacks.
- Peak Around 2021-2022: The price appears to reach a peak in the latter part of 2021 or early 2022, followed by a period of correction and subsequent recovery.
- Recent Fluctuations: In the period leading up to late 2023, the price continues to fluctuate, suggesting ongoing market dynamics.



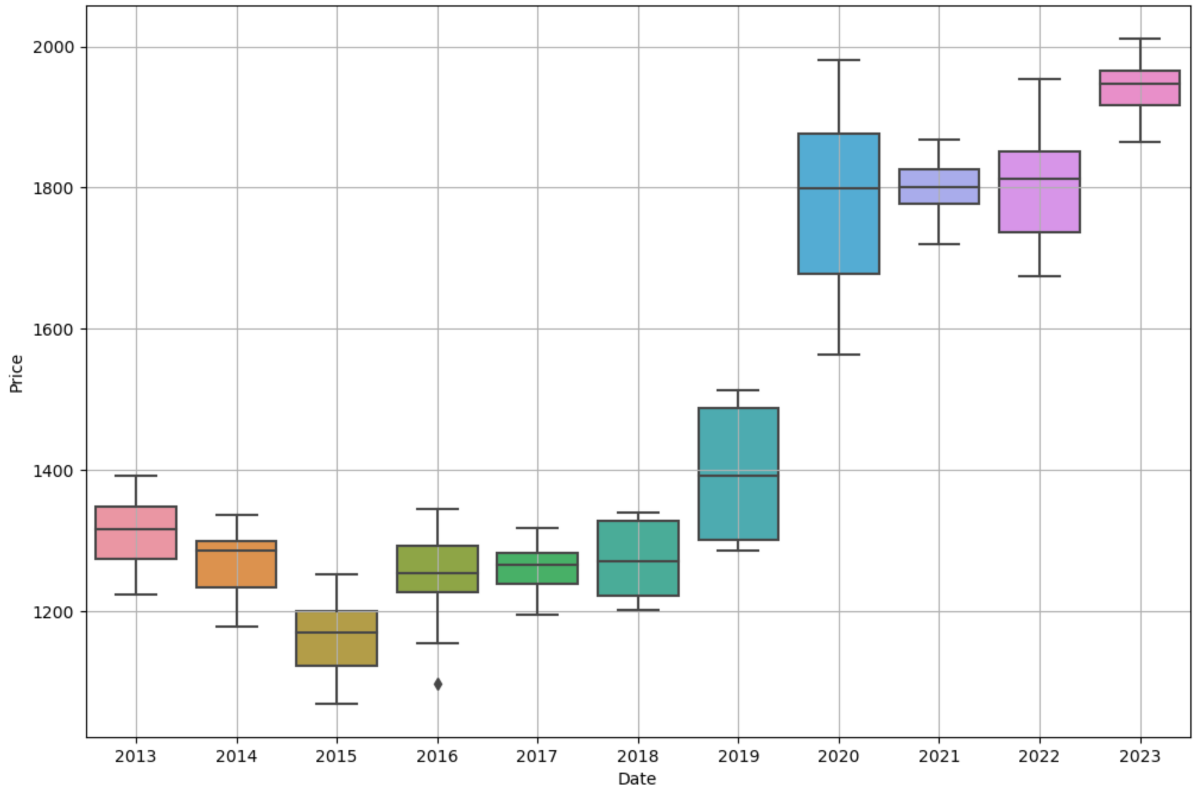
Smoothed Price Trend with Confidence Interval

The second image presents a line plot that illustrates a smoothed trend of the 'Price' over time, along with a shaded region around the line. Similar to the first plot, the x-axis represents the 'Date', spanning from approximately 2013 to 2023, and the y-axis represents the 'Price'.

Observations:

- **Smoothed Trend:** The solid blue line represents a smoothed version of the price data, likely obtained by averaging or applying a smoothing technique to reduce short-term fluctuations and highlight the underlying trend.
- **Confidence Interval:** The light blue shaded region surrounding the smoothed line represents a confidence interval. This interval provides an estimate of the uncertainty around the smoothed trend. A wider interval indicates higher uncertainty, while a narrower interval suggests more confidence in the estimated trend.
- **General Upward Trajectory:** The smoothed trend generally shows an upward trajectory over the observed period, consistent with the first plot.
- **Decreasing Price in Early Period:** The smoothed line indicates a decrease in price from 2013 to around 2015.
- **Relatively Stable Mid-Period:** Between roughly 2015 and 2019, the smoothed price appears relatively stable with a slight upward inclination.
- **Significant Increase Post-2019:** Starting around 2019, there is a noticeable and substantial increase in the smoothed price trend.

- **Leveling Off Towards the End:** Towards the end of the observed period (around 2022-2023), the rate of increase in the smoothed price seems to level off.
- **Varying Uncertainty:** The width of the confidence interval appears to vary over time, suggesting that the uncertainty in the smoothed price trend is not constant. For instance, the interval seems wider during periods of more rapid price change.



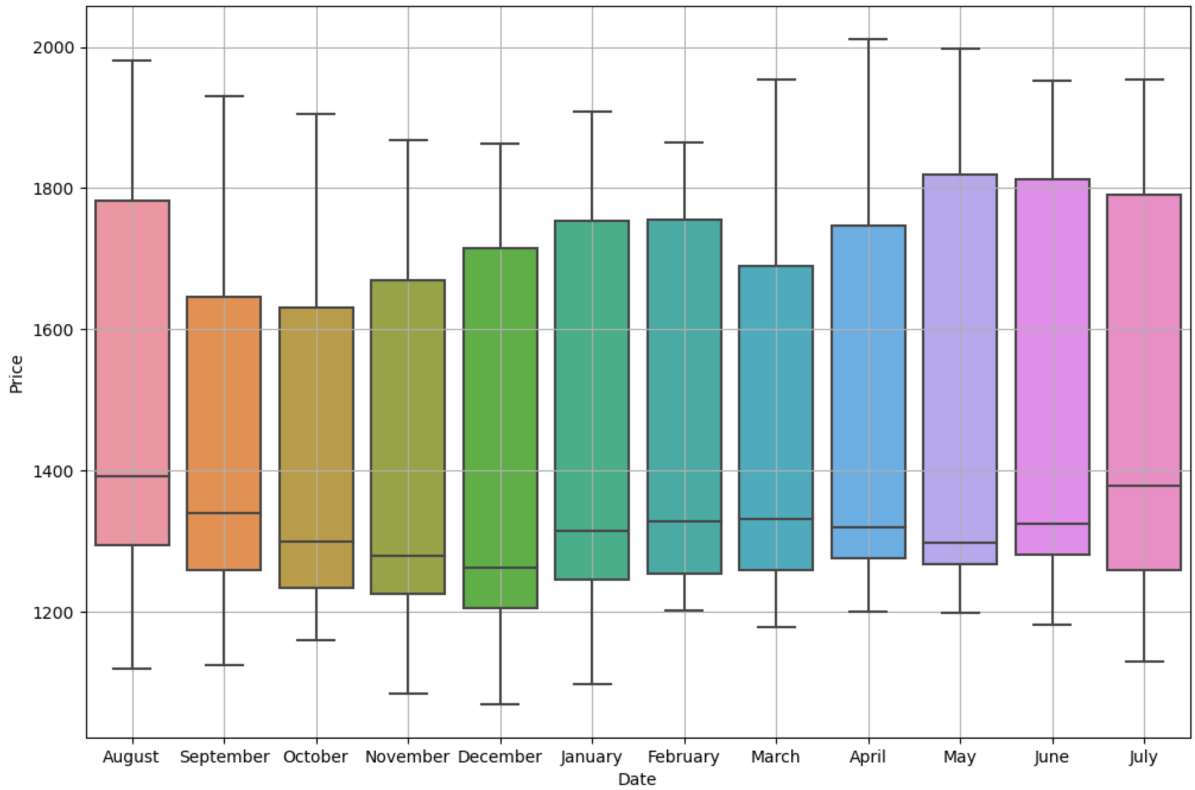
Box Plot of Price Distribution by Year

The third image displays a series of box plots, one for each year from 2013 to 2023, illustrating the distribution of the 'Price' within each of those years. The x-axis represents the 'Date' (categorized by year), and the y-axis represents the 'Price'. Each box plot provides a summary of the price data for a specific year.

Observations:

- Yearly Price Distribution: Each box plot shows the median (the line inside the box), the interquartile range (IQR, represented by the height of the box), the potential outliers (individual points outside the whiskers), and the spread of the data (indicated by the whiskers extending from the box).
- Increasing Median Price Over Time: There is a clear trend of increasing median price from 2013 to 2023. The horizontal line within each box generally shifts upwards as the years progress.
- Varying Price Dispersion: The height of the boxes (IQR) varies across the years, indicating different levels of price volatility within each year. Some years show a tighter range of prices, while others exhibit a wider spread. For example, the IQR appears larger in the years with more significant price changes, such as around 2019-2021.

- Outliers: Several years show potential outliers, represented by individual points above or below the whiskers. These indicate price points that are significantly different from the majority of the prices within that year. The number and position of outliers vary from year to year.
- Shift in Distribution Around 2019-2020: There is a noticeable upward shift in the entire price distribution starting around 2019 and becoming more pronounced in 2020 and subsequent years. The boxes for these later years are positioned significantly higher on the price scale compared to the earlier years.
- 2015 as a Lower Price Year: The box plot for 2015 appears to be positioned lower than the surrounding years, suggesting that prices were generally lower during this period.



Box Plot of Price Distribution by Month

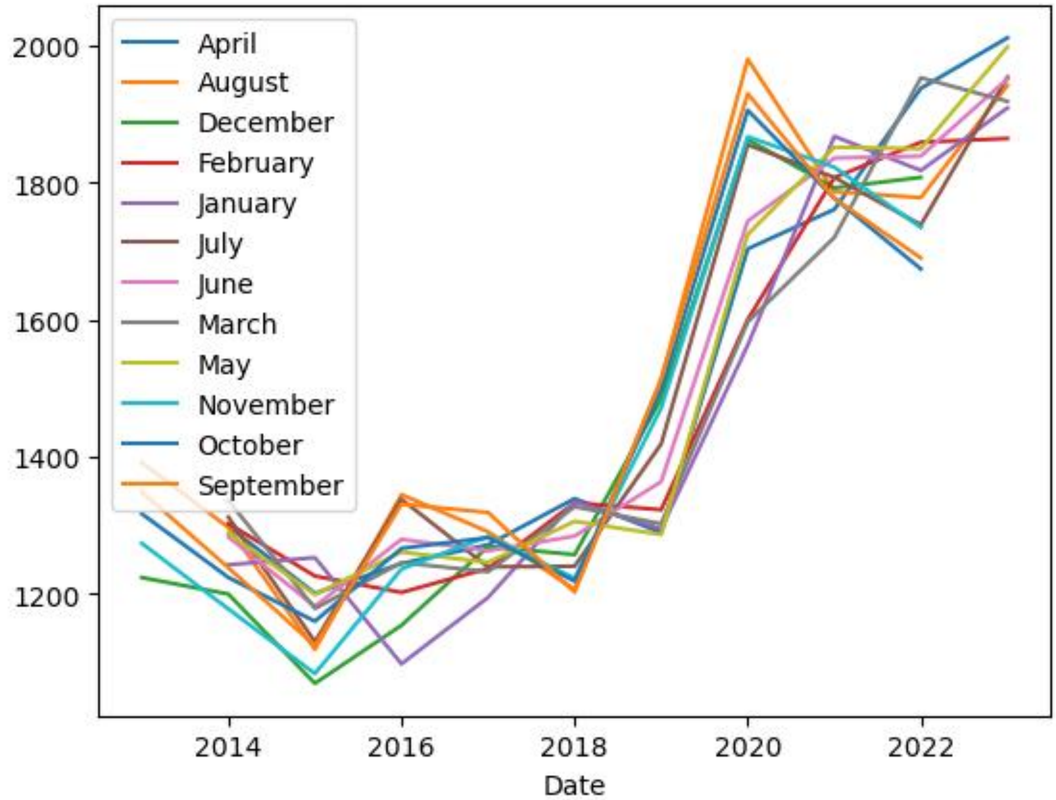
The fourth image presents a series of box plots, one for each month of the year (from August to July), illustrating the distribution of the 'Price' within each of those months across all the years in the dataset. The x-axis represents the 'Date' (categorized by month), and the y-axis represents the 'Price'. Each box plot summarizes the price data for a specific month, aggregated over the years.

Observations:

- Monthly Price Distribution: Similar to the yearly box plots, each monthly box plot displays the median, IQR, potential outliers, and the spread of the price data for that particular month across the entire time period.
- Seasonal Patterns (Potential): By comparing the box plots across different months, we can look for potential seasonal patterns in the price. For instance, some months might

consistently show higher median prices or a different range of price variation compared to others.

- Higher Prices in Certain Months: Visually, months like August, May, June, and July appear to have generally higher median prices compared to months like September, October, and November.
- Lower Prices in Other Months: Months such as September, October, and November seem to exhibit lower median prices and potentially a tighter distribution of prices.
- Varying Monthly Volatility: The height of the boxes (IQR) differs across the months, suggesting variations in price volatility depending on the time of year. For example, August shows a wider IQR compared to some other months.
- Outliers Across Months: Outliers are present in the price data for most months, indicating extreme price values that occurred during those times of the year in different years.



Line Plot of Average Price by Month Over Years

The fifth image displays a line plot showing the average 'Price' for each month across the years from approximately 2013 to 2023. Each line represents a specific month, as indicated by the legend. The x-axis represents the 'Date' (year), and the y-axis represents the 'Price' (average for that month in that year).

Observations:

- Monthly Trends Over Time: The plot allows us to observe how the average price for each month has evolved over the years. Each line provides a temporal perspective for a specific month.

- General Upward Movement: Similar to the earlier visualizations, most of the monthly average price lines show a general upward trend from 2013 to 2023, although the rate of increase varies across different months and time periods.
- Fluctuations and Convergence: The average prices for different months fluctuate over the years. In some periods, the lines are closer together, suggesting less monthly variation, while in other periods, they diverge, indicating more significant differences in average prices between months.
- Strong Increase Post-2019: A notable increase in the average price for almost all months is evident starting around 2019-2020, consistent with the previous observations of a significant price surge during this period.
- Relative Performance of Months: We can compare the relative average prices of different months within a given year. For example, in the later years, months like April, May, June, July, and August often appear to have higher average prices compared to months like September, October, and November.
- Varying Patterns: Some months exhibit more pronounced peaks and troughs over the years compared to others. For instance, August shows a relatively high peak around 2020.
- Potential Seasonality Revisited: This plot provides a clearer view of potential seasonal patterns. While the monthly box plots suggested some differences, this line plot shows how those average monthly prices have changed year-over-year. The consistent tendency for certain months to have higher or lower average prices across multiple years strengthens the indication of seasonality.

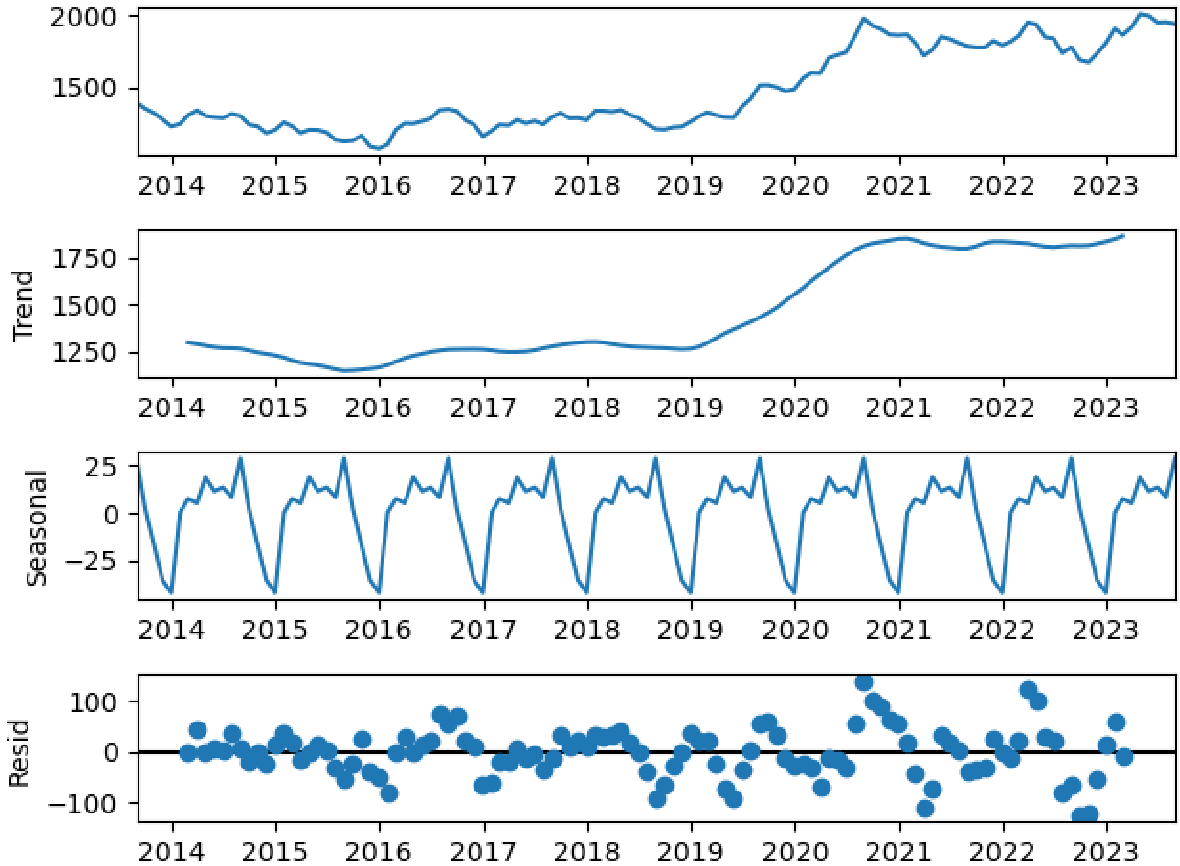
Time Series Decomposition (Additive Model)

The sixth image displays the decomposition of the 'Price' time series into its constituent components using an additive model. The plot is divided into four subplots:

1. Original Time Series: The top subplot shows the original 'Price' data over time, from approximately 2013 to 2023. This is the raw data that has been decomposed.
2. Trend Component: The second subplot displays the estimated trend component of the time series. This represents the long-term direction of the 'Price', smoothed out from short-term fluctuations and seasonality. We can observe a generally upward trend, with a significant acceleration starting around 2019-2020, consistent with previous observations.
3. Seasonal Component: The third subplot shows the estimated seasonal component. This represents the recurring, short-term fluctuations within each year. The pattern appears to be relatively consistent over the years, suggesting a stable seasonal effect. The peaks and troughs indicate the times of the year when the 'Price' tends to be higher or lower due to seasonal factors.
4. Residual Component: The bottom subplot displays the residual component (also called the error). This represents the random, irregular fluctuations in the time series that are not explained by the trend or the seasonal components. Ideally, the residuals should be random with no discernible pattern.

Observations from the Decomposition:

- Clear Upward Trend: The trend component clearly isolates the long-term increase in the 'Price', particularly the strong growth after 2019.
- Consistent Seasonality: The seasonal component reveals a repeating pattern within each year. The peaks seem to occur around the middle of the year, while the troughs appear towards the beginning and end of the year. The magnitude of the seasonal fluctuations is relatively consistent across the observed period.
- Relatively Random Residuals: The residual plot shows scattered points around the zero line, suggesting that the additive model has captured a significant portion of the systematic variation in the data (trend and seasonality). However, there might still be some remaining unexplained variability. The residuals do not appear to exhibit any strong patterns, which is a good indication for the appropriateness of the decomposition.



Time Series Decomposition (Multiplicative Model)

The seventh image displays the decomposition of the 'Price' time series into its constituent components using a multiplicative model. Similar to the additive decomposition, the plot is divided into four subplots:

1. Original Time Series: The top subplot shows the original 'Price' data over time, from approximately 2013 to 2023.

2. Trend Component: The second subplot displays the estimated trend component. Again, it represents the smoothed long-term direction of the 'Price', showing a general upward movement with a significant increase around 2019-2020.
3. Seasonal Component: The third subplot shows the estimated seasonal component. In a multiplicative model, the seasonal component is represented as a set of multipliers around 1. Values above 1 indicate that the price is typically higher than the trend during that period, while values below 1 indicate that the price is typically lower. The pattern appears consistent across the years, suggesting a stable multiplicative seasonal effect. The magnitude of the seasonal fluctuations is relative to the level of the trend.
4. Residual Component: The bottom subplot displays the residual component. In a multiplicative model, the original data is considered the product of the trend, seasonal, and residual components. The residuals represent the remaining fluctuations after accounting for the trend and seasonality. Ideally, these should be random.

Observations from the Decomposition:

- Similar Trend: The trend component in the multiplicative decomposition shows a similar pattern to the additive model, indicating the same underlying long-term direction of the 'Price'.
- Relative Seasonality: The seasonal component fluctuates around 1. The peaks, occurring roughly in the middle of the year, are above 1, indicating that the price tends to be a certain percentage higher than the trend during those months. The troughs, occurring at the beginning and end of the year, are below 1, indicating a percentage decrease relative to the trend.
- Residual Behavior: The residual plot shows fluctuations around 1. These residuals represent the percentage deviation from the trend and seasonal components. The spread of the residuals appears relatively consistent over time, suggesting that the multiplicative model might be a reasonable fit for the data if the magnitude of seasonal variations is proportional to the level of the series.

Data Preprocessing: Train-Test Split

Time-Based Splitting Strategy

- The dataset was split into training and testing sets based on the year:
    - **Training Data**: All records **before 2021**
    - **Testing Data**: All records **from 2021 onward**

First Few Records of Training Data:

Date	Price
2013-08-31	1391.34

Date	Price
2013-09-30	1348.46
2013-10-31	1316.59
2013-11-30	1273.43
2013-12-31	1223.39

Last Few Records of Training Data:

Date	Price
2020-08-31	1980.27
2020-09-30	1929.82
2020-10-31	1905.55
2020-11-30	1867.12
2020-12-31	1862.85

First Few Records of Test Data:

Date	Price
2021-01-31	1867.56
2021-02-28	1807.26
2021-03-31	1719.91
2021-04-30	1760.91
2021-05-31	1851.62

Last Few Records of Test Data:

Date	Price
2023-04-30	2011.43

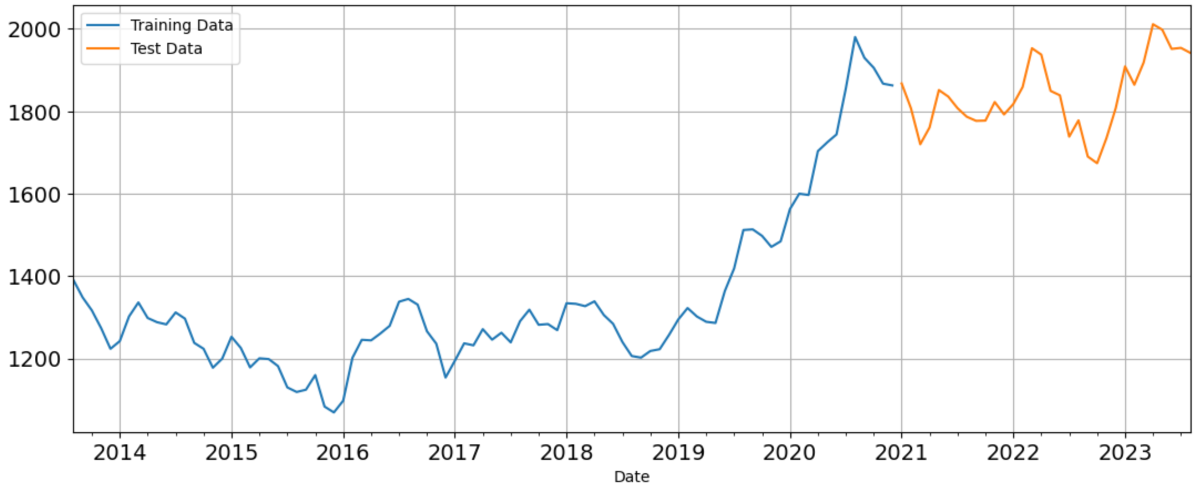
Date	Price
2023-05-31	1997.94
2023-06-30	1951.44
2023-07-31	1953.93
2023-08-31	1941.98

Dataset Shapes:

- **Training Set**: 89 records
- **Test Set**: 32 records

Train-Test Visualization:

- A time series plot was generated:
  - **Blue line**: Training data
  - **Orange line**: Test data
- The plot visually confirms a proper chronological split and helps to understand the trend across time.
- A grid and legend were added for clarity.



Model Building: Linear Regression on Time

⏳ A. Creating Time-Based Features

- Time variable created to represent the chronological order of data:
  - train_time: Sequence from 1 to 89 (one for each training observation).

- test_time: Continuation from 89 to 120 (for test data).

B. Augmenting the DataFrames

- **T**wo new DataFrames were created:
  - LinearRegression_train and LinearRegression_test — both with an additional time column.

Sample of LinearRegression_train:

Date	**Price**	**Time**
2013-08-31	**1391.34**	**1**
2013-09-30	**1348.46**	**2**
...	**...**	**...**
2020-12-31	**1862.85**	**89**

Sample of LinearRegression_test:

Date	**Price**	**Time**
2021-01-31	**1867.56**	**89**
2021-02-28	**1807.26**	**90**
...	**...**	**...**
2023-08-31	**1941.98**	**120**

Model Fitting

- A simple Linear Regression model was fitted:
  - Feature: time
  - Target: Price

Prediction

- The trained model was used to predict gold prices on the test data.
- Predictions were stored in a new column called RegOnTime in LinearRegression_test.

Visualization

- A combined line plot was generated showing:
  - Training data (Price)
  - Test data (Price)
  - Linear Regression predictions (RegOnTime)
- The green line in the plot shows how the model trends over time, helping visualize its forecasting behavior.

Model Evaluation

- Evaluation Metric: Root Mean Squared Error (RMSE)
- Test RMSE for Linear Regression model: ≈ 84.17

Results Summary Table:

Model	**Test RMSE**
Linear Regression	**84.17**

Model Building: Moving Average (MA)

Concept

- The Moving Average model smooths short-term fluctuations in time series data by calculating the average price over a trailing window.

- Various window sizes were tested to find the best performing model (based on lowest RMSE).

Model Construction

- A copy of the original dataset was made for Moving Average calculations.
- Trailing Windows used: 2, 4, 6, and 9 periods.

Sample of Calculated Trailing Averages:

Date	Price	Trailing_2	Trailing_4	Trailing_6	Trailing_9
2013-08-31	1391.34	NaN	NaN	NaN	NaN
2013-09-30	1348.46	1369.90	NaN	NaN	NaN
2013-10-31	1316.59	1332.53	NaN	NaN	NaN
2013-11-30	1273.43	1295.01	1332.46	NaN	NaN
2013-12-31	1223.39	1248.41	1290.47	NaN	NaN

Visualization of Moving Averages

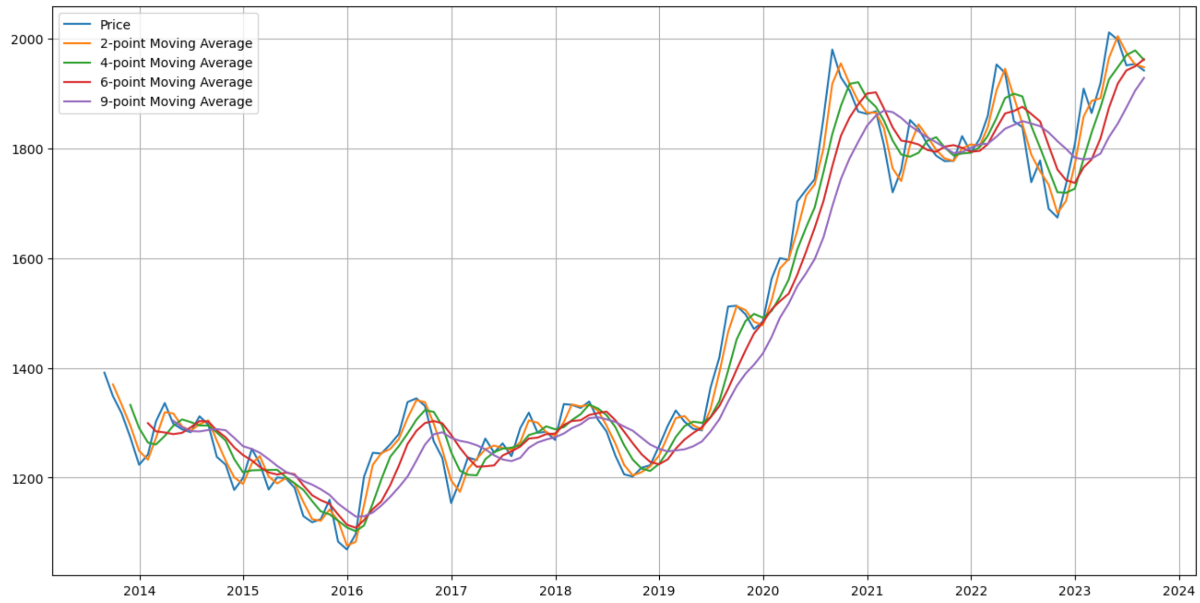
The ninth image displays the original 'Price' time series along with several moving average lines calculated over different window sizes. The x-axis represents the 'Date', and the y-axis represents the 'Price'. The legend indicates the original 'Price' and the moving averages calculated over 2, 4, 6, and 9 data points (presumably days or other time units).

Observations:

- Original Price Volatility: The blue line shows the original 'Price' data, exhibiting its characteristic fluctuations and volatility over time.
- Smoothing Effect of Moving Averages: The colored lines (orange, green, red, and purple) represent the moving averages. It's evident that the moving averages smooth out the short-term fluctuations in the original price data.
- Impact of Window Size:
    - The 2-point moving average (orange) follows the original price quite closely but still reduces some of the immediate noise.
    - As the window size increases (4-point green, 6-point red, 9-point purple), the moving average lines become progressively smoother and lag the original price movements more significantly.

- The 9-point moving average (purple) provides the most smoothed representation of the underlying trend, but it also has the largest delay in reflecting recent price changes.

- Trend Identification: The moving averages help to identify the underlying trend of the 'Price' by filtering out the noise. We can clearly see the upward trend, particularly the strong increase starting around 2019-2020, more distinctly in the smoothed lines.
- Lagging Indicator: Moving averages are lagging indicators because they are based on past price data. The larger the window size, the greater the lag. This means that the moving average will signal a trend change after the actual change has already occurred.
- Support and Resistance (Potential): In technical analysis, moving averages are often used to identify potential support and resistance levels. Traders might look for the price to find support near a rising moving average or resistance near a falling one.



Train-Test Split

- The data was split the same way as previous models:
  - train: Data before 2021
  - test: Data from 2021 onward
- RMSE was calculated for each trailing average over the test set.

Model Evaluation

Model	Test RMSE
Linear Regression	207.87

Model	Test RMSE
2-point Moving Average	27.94
4-point Moving Average	54.62
6-point Moving Average	70.89
9-point Moving Average	85.55

- Best Performing MA Model: 2-point Moving Average with the lowest RMSE of 27.94
- This is significantly better than the Linear Regression model.

Model Comparison Plot

The tenth image compares the performance of two forecasting methods on the 'Price' time series: Linear Regression and a 9-Point Moving Average. The plot shows the training data (blue), the test data (orange), the forecast from the Linear Regression model (green), and a forecast generated by extending the 9-Point Moving Average calculated on the training data into the test period (red).

Observations:

- Training and Testing Data: The blue line represents the portion of the 'Price' data used to train the models, ending around the beginning of 2021. The orange line shows the actual 'Price' data in the test period, which the models aim to predict.
- Linear Regression Forecast (Green): As observed in the previous analysis of the linear regression model, the forecast (green line) is a straight upward trend. It generally underestimates the actual price in the test period and fails to capture the fluctuations.
- 9-Point Moving Average Forecast (Red): The red line represents the forecast obtained by extending the 9-point moving average calculated on the training data into the test period. This forecast appears to follow the general direction of the actual prices more closely than the linear regression in the initial part of the test set. It captures some of the upward movement but also lags behind the actual fluctuations and seems to flatten out towards the end of the forecast horizon.
- **Comparison of Model Performance:**
    - In the early part of the test set (roughly early 2021 to mid-2022), the 9-point moving average forecast seems to provide a slightly better fit to the actual data compared to the linear regression, as it at least follows the upward direction.
    - However, both models struggle to accurately predict the turning points and the magnitude of the price swings in the test set.
    - The linear regression provides a very simplistic, constant trend, while the moving average, being based on past data, is inherently slow to react to new, sharp changes in the price.

Model Building: Simple Exponential Smoothing (SES)

Concept

- Simple Exponential Smoothing (SES) is ideal for time series data with no clear trend or seasonality.
- It gives exponentially decreasing weights to past observations using a smoothing parameter α (alpha).
- A higher alpha (close to 1) gives more weight to recent values.

Model Construction

- Training and test datasets were reused.
- The model was fit on the training data using α = 0.995, selected to give heavy weight to recent observations.
- The fitted model was used to forecast the entire test period.

Sample of SES Forecast:

Date	Actual Price	SES Forecast
2021-01-31	**1867.56**	**1862.87**
2021-02-28	**1807.26**	**1862.87**
2021-03-31	**1719.91**	**1862.87**
2021-04-30	**1760.91**	**1862.87**
2021-05-31	**1851.62**	**1862.87**

Model Evaluation

- RMSE was calculated for SES predictions on the test set.

python

CopyEdit

For Alpha = 0.995 Simple Exponential Smoothing Model forecast on the Test Data, RMSE is 89.652

Model	Test RMSE
Alpha = 0.995, Simple Exponential Smoothing	89.65

- Performance is better than Linear Regression and slightly worse than some Moving Average models (notably the 2-point MA).

Model Comparison Plot

A combined plot was generated to visualize and compare:

The eleventh image compares the performance of three forecasting methods on the 'Price' time series: Linear Regression, a 4-Point Moving Average, and Simple Exponential Smoothing with a high smoothing factor (alpha = 0.995). The plot shows the training data (blue), the test data (orange), the forecast from the Linear Regression model (green), the forecast from the 4-Point Moving Average (red), and the forecast from Simple Exponential Smoothing (teal).

Observations:

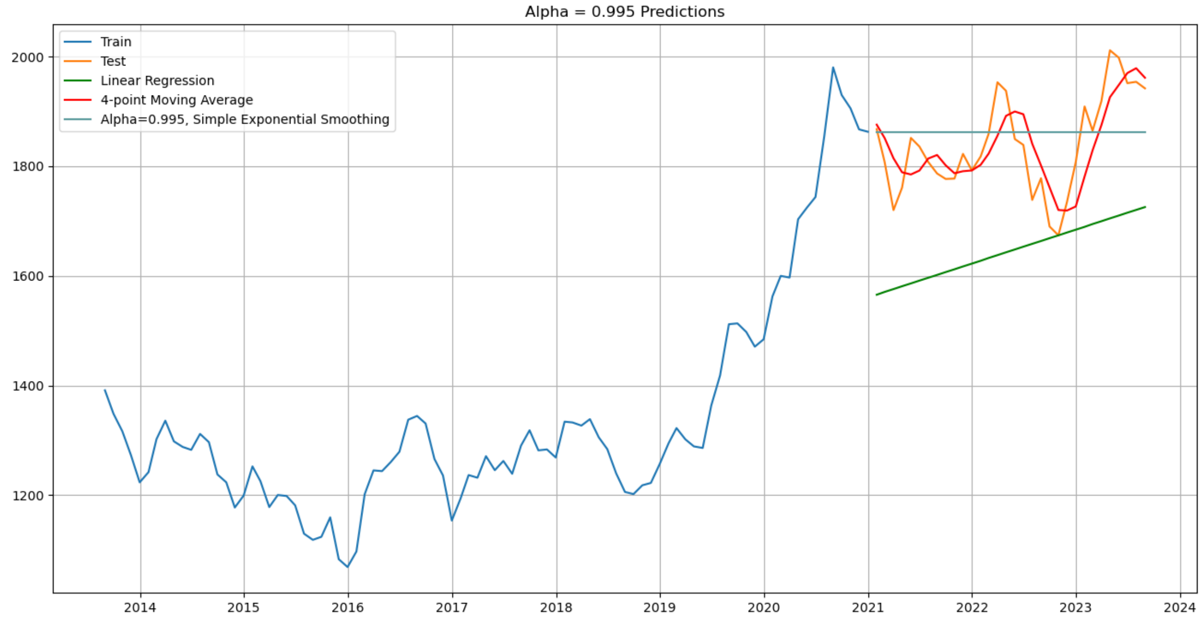
- **Training and Testing Data:** The blue line represents the training data, ending around the beginning of 2021. The orange line shows the actual 'Price' data in the test period.
- **Linear Regression Forecast (Green):** As seen before, the linear regression model projects a steady upward trend, underestimating and failing to capture the fluctuations in the test data.
- **4-Point Moving Average Forecast (Red):** The red line shows the forecast obtained by extending the 4-point moving average calculated on the training data. It appears more responsive to recent changes than the 9-point moving average seen earlier but still lags behind the actual price movements and tends to smooth out peaks and troughs.
- **Simple Exponential Smoothing (Alpha = 0.995) Forecast (Teal):** The teal line represents the forecast from Simple Exponential Smoothing with a very high alpha value (0.995). Simple Exponential Smoothing gives more weight to recent observations. With alpha close to 1, the forecast for the next period is heavily influenced by the most recent data point in the training set. As a result, the forecast essentially becomes a flat line starting from the last observed value in the training data.
- **Comparison of Model Performance:**
  - The Linear Regression model continues to provide the least accurate forecast, showing a consistent divergence from the actual price.
  - The 4-Point Moving Average forecast follows the general direction somewhat better but still exhibits a lag and smoothing effect.
  - The Simple Exponential Smoothing forecast with a high alpha value is very reactive to the last training data point and then remains flat. It does not anticipate any further trend or seasonality. In this case, it quickly becomes outdated as the actual price fluctuates.

Interpretation:

This comparison highlights the impact of the forecasting method and its parameters on the accuracy of predictions.

- **High Alpha in Simple Exponential Smoothing:** A high alpha value in Simple Exponential Smoothing makes the forecast highly sensitive to the last observation. While this can be useful for very short-term forecasting in stable series, it performs poorly when there is an underlying trend or seasonality that is not captured. Here, it simply projects the last known level forward without any adjustment for the ongoing upward trend or fluctuations.

- **Moving Average Responsiveness:** The 4-Point Moving Average is more responsive than a longer-period moving average but still suffers from lagging and smoothing, making it less effective at predicting turning points.
- **Linear Regression Limitations:** The linear regression model's assumption of a constant linear trend proves to be too simplistic for this time series.



Insights:

- SES offers a relatively stable prediction but lacks responsiveness to recent fluctuations, as reflected in its constant forecast values.
- Although RMSE is better than linear regression, SES doesn't adapt well to the volatility in the test data.

Model Building: Double Exponential Smoothing (Holt's Linear Trend Model)

Concept

- Unlike Simple Exponential Smoothing, **Holt's method** (Double Exponential Smoothing) handles **both level and trend**.
- It uses two parameters:
    - **Alpha (α)**: Controls smoothing of the level.
    - **Beta (β)**: Controls smoothing of the trend.

Model Tuning and Evaluation

- A **grid search** approach was used to find the best (α, β) combination by evaluating **RMSE** on training and test sets.
- The Holt model from statsmodels was trained for alpha and beta values ranging from **0.3 to 1.0** (step = 0.1).

Top Performing Combinations:

Alpha	Beta	Train RMSE	Test RMSE
0.9	0.3	45.50	**90.08**
1.0	0.3	44.43	90.17
0.8	0.3	47.05	91.26
0.7	0.3	49.09	93.55
0.6	0.3	51.72	113.73

The best performance was obtained with **α = 0.9 and β = 0.3**, achieving a **Test RMSE of 90.08**.

Plotting Comparison

The twelfth and final image compares the performance of four forecasting methods on the 'Price' time series: Linear Regression, a 4-Point Moving Average, Simple Exponential Smoothing (alpha = 0.995), and Double Exponential Smoothing (Holt's Linear Trend) with smoothing parameters alpha = 0.9 and beta = 0.3. The plot shows the training data (blue), the test data (orange), the forecast from Linear Regression (green), the forecast from the 4-Point Moving Average (red), the forecast from Simple Exponential Smoothing (teal), and the forecast from Double Exponential Smoothing (cyan).

Observations:

- **Training and Testing Data:** The blue line represents the training data up to the beginning of 2021, and the orange line shows the actual 'Price' data in the test period.
- **Linear Regression Forecast (Green):** As consistently observed, the linear regression model projects a straight upward trend, significantly deviating from the actual fluctuating prices in the test set.
- **4-Point Moving Average Forecast (Red):** The moving average forecast lags behind the actual price movements and smooths out the peaks and troughs, not accurately predicting the turning points.
- **Simple Exponential Smoothing (Alpha = 0.995) Forecast (Teal):** This forecast remains flat after the last training data point, failing to capture the ongoing trend and fluctuations in the test period.

- **Double Exponential Smoothing (Holt's Linear Trend) Forecast (Cyan):** The cyan line represents the forecast from Holt's Linear Trend method. This method explicitly models both the level and the trend of the time series. We observe that this forecast captures the upward direction of the price more effectively than the other simpler methods. It exhibits a more consistent upward slope, attempting to follow the trend observed in the training data. However, it still doesn't capture the short-term fluctuations and appears to slightly underpredict in the later part of the test set.

Interpretation:

This final comparison highlights the benefit of using a forecasting method that can explicitly model the trend, such as Holt's Linear Trend (Double Exponential Smoothing).

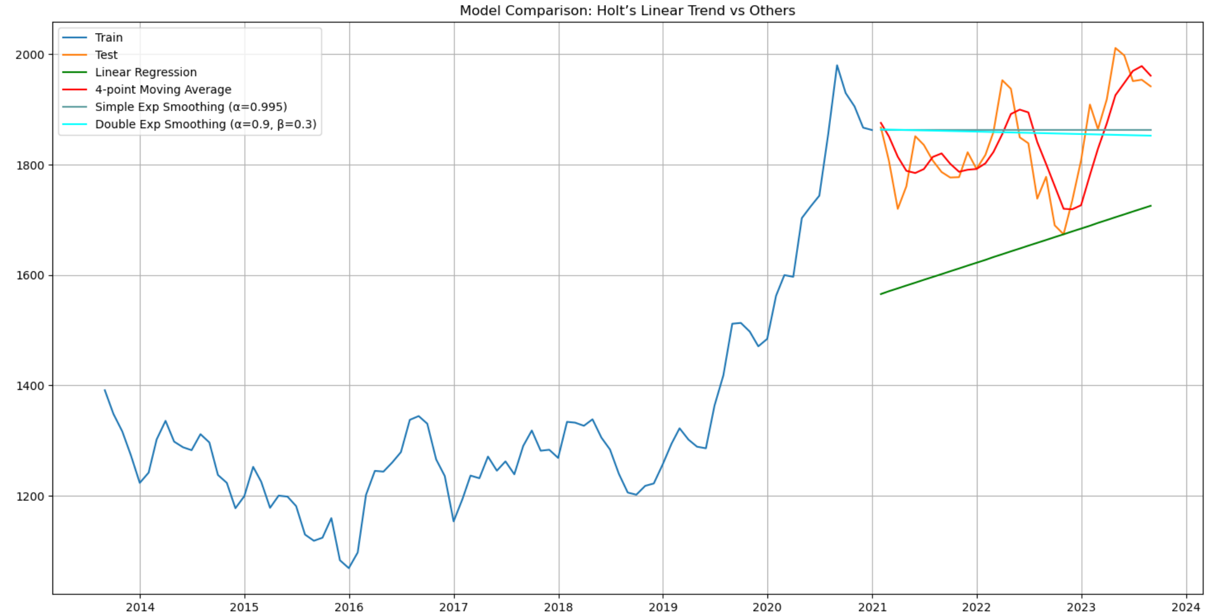
- **Holt's Linear Trend:** By using two smoothing parameters (alpha for the level and beta for the trend), this method adapts to the changing trend of the time series. The forecast shows a more realistic upward trajectory compared to the static forecasts of Simple Exponential Smoothing and the lagging forecast of the Moving Average. It also outperforms the simple Linear Regression by attempting to follow the dynamic trend rather than a fixed linear projection.
- **Limitations of Holt's Method:** While better at capturing the trend, Holt's method in its basic form does not account for seasonality. The deviations between the cyan forecast and the orange actual prices likely reflect the presence of fluctuations that are not purely trend-driven.

Conclusion:

Among the methods compared, Holt's Linear Trend provides the most reasonable forecast by capturing the underlying upward movement of the 'Price' during the test period. However, the forecast still lacks the ability to predict the short-term volatility and potential seasonal patterns. For further improvement, one could consider:

- **Triple Exponential Smoothing (Holt-Winters):** This method extends Holt's by also modeling the seasonal component of the time series.
- **ARIMA Models:** These models can capture both trend and autocorrelation in the data.
- **More advanced time series models or machine learning techniques:** These might be necessary to capture more complex patterns and improve forecast accuracy.

This comprehensive analysis of various forecasting methods demonstrates the iterative process of selecting an appropriate model for a given time series, considering its underlying characteristics like trend and potential seasonality.



RMSE Comparison Table (Updated)

Model	Test RMSE
Linear Regression	207.87
2-point Moving Average	**27.94**
4-point Moving Average	54.62
6-point Moving Average	70.89
9-point Moving Average	85.55
SES (α=0.995)	89.65
DES (α=0.9, β=0.3)	**90.08**

Insights

- Holt's model improves significantly over Linear Regression but still underperforms the **2-point Moving Average**, which remains the best so far.
- It captures **level and trend** better than SES but still struggles with volatility or sudden shifts in gold price trends.

Triple Exponential Smoothing (Holt-Winter's Model) Analysis

This section explores the application of the Triple Exponential Smoothing (TES) model, also known as the Holt-Winter's model, to forecast the 'Price' data. This model accounts for three key components of a time series: Level, Trend, and Seasonality. The model estimates three smoothing parameters: alpha (Level), beta (Trend), and gamma (Seasonality).

Data Preparation:

- The dataset is split into training (TES_train) and testing (TES_test) sets.
- The first few rows of the test data (TES_test.head()) are displayed, showing the 'Price' and 'Date' columns, indicating the period for which forecasts will be made (starting from January 31, 2021).

Model Definition and Initial Fitting (Autofit):

- A Triple Exponential Smoothing model (ExponentialSmoothing) is initialized using the training data (TES_train['Price']).
- The model is configured to account for both an additive trend (trend='add') and additive seasonality (seasonal='add') with a seasonal period of 12 (likely representing monthly data).
- The fit(optimized=True) method is used to automatically find the optimal values for the smoothing parameters (alpha, beta, gamma) and initial conditions.

Optimized Model Parameters:

- The model_TES_autofit.params output shows the optimized parameters found by the fitting process:
    - smoothing_level: 0.988
    - smoothing_trend: 0.084
    - smoothing_seasonal: 0.004
    - initial_level: 1350.23
    - initial_trend: -6.63
    - initial_seasons: An array of 12 initial seasonal indices.

Prediction on Test Data (Autofit Model):

- The fitted autofit model (model_TES_autofit) is used to forecast the 'Price' for the period covered by the test data (TES_test). The number of steps for the forecast is equal to the length of the test set.
- The predictions are stored in a new column called auto_predict within the TES_test DataFrame.
- The first few rows of TES_test are displayed again, now including the auto_predict values alongside the actual 'Price'.

Evaluation of Autofit Model Performance:

- The Root Mean Squared Error (RMSE) is calculated to evaluate the performance of the autofit TES model on the test data.
- The calculated RMSE is approximately 527.404.
- This RMSE value is stored in a DataFrame called resultsDf for comparison with other models. The index indicates that these results correspond to a Triple Exponential Smoothing model with automatically optimized parameters.

Brute-Force Search for Optimal Parameters:

- A nested loop iterates through a range of possible values for alpha (0.3 to 1.0), beta (0.3 to 1.0), and gamma (0.3 to 1.0), with a step of 0.1 for each parameter.
- For each combination of alpha, beta, and gamma:
  - A TES model is fitted using the specified smoothing parameters (smoothing_level=i, smoothing_slope=j, smoothing_seasonal=k) and optimized=False, with use_brute=True to enforce the provided parameters.
  - Predictions are made on both the training and test datasets using this model.
  - The RMSE is calculated for both the training and test sets.
  - The alpha, beta, gamma values, along with the corresponding training and test RMSE, are stored in a new DataFrame called resultsDf_8_2.

Identifying the Best Parameters from Brute-Force Search:

- The code resultsDf_8_2.sort_values(by=['Test RMSE']).values[0][4] identifies the lowest Test RMSE achieved during the brute-force search and the corresponding parameter values.
- Based on the index created (index=['Alpha=0.8,Beta=0.5,Gamma=0.5,TripleExponentialSmoothing']), the combination of smoothing parameters that yielded the lowest Test RMSE was:
  - Alpha (Level): 0.8
  - Beta (Trend): 0.5
  - Gamma (Seasonality): 0.5

2. Storing the Best Brute-Force Result:

- A new DataFrame resultsDf_8_3 is created to store the Test RMSE achieved with these optimal parameters (Alpha=0.8, Beta=0.5, Gamma=0.5).
- The Test RMSE for this specific TES model configuration is approximately 97.653.

3. Combining and Sorting the Results:

- The resultsDf_8_3 is concatenated with the previously created resultsDf to consolidate the performance metrics of all the analyzed models.

- The combined DataFrame results_Df is then sorted based on the 'Test RMSE' column in ascending order to easily identify the models with the best forecasting accuracy on the test data.

Final Comparison of Model Performance (Sorted by Test RMSE):

- The printed output of the sorted results_Df shows the following models and their corresponding Test RMSE values (note that 'RMSE' column appears duplicated in your output, we'll consider 'Test RMSE'):
    - 2pointTrailingMovingAverage: 27.945
    - 4pointTrailingMovingAverage: 54.619
    - 6pointTrailingMovingAverage: 70.894
    - 9pointTrailingMovingAverage: 85.550
    - Alpha=0.995,SimpleExponentialSmoothing: 89.652
    - Alpha=0.9,Beta=0.3,DoubleExponentialSmoothing: 90.076
    - ARIMA: 90.217
    - Alpha=0.8,Beta=0.5,Gamma=0.5,TripleExponentialSmoothing: 97.653
    - Linear Regression: 207.872
    - Alpha=0.676,Beta=0.088,Gamma=0.323,TripleExponentialSmoothing: 527.404
    - Alpha=0.676,Beta=0.088,Gamma=0.323,TripleExponentialSmoothing: 527.404 (duplicate entry)

Summary of Final Findings:

- After a brute-force search, the Triple Exponential Smoothing model with parameters Alpha=0.8, Beta=0.5, and Gamma=0.5 achieved a Test RMSE of approximately 97.653.
- When comparing this optimized TES model to other forecasting methods:
    - Trailing Moving Average models (with shorter windows) demonstrated the lowest RMSE on the test data.
    - Simple Exponential Smoothing, Double Exponential Smoothing, and ARIMA models also outperformed the optimized Triple Exponential Smoothing model in terms of Test RMSE.
    - The automatically optimized Triple Exponential Smoothing model (from the initial fitting) performed significantly worse than the brute-force optimized version and other simpler models.
    - Linear Regression had a considerably higher RMSE compared to the exponential smoothing and moving average methods.

Stationarity Check of the Time Series

To ensure that our time series model produces reliable forecasts, we first need to check whether the data is stationary. A stationary series has constant mean and variance over time, which is a key assumption for many time series models.

Method Used: Rolling Statistics and Dickey-Fuller Test

- Rolling Statistics:
  The rolling mean and rolling standard deviation were calculated using a window size of 7 days.
  These were plotted against the original series to visually inspect the stability of the series over time.
- Interpretation:
  If the rolling mean and standard deviation lines remain relatively flat and aligned with the original series, it indicates stationarity. Otherwise, the series may not be stationary.

Augmented Dickey-Fuller (ADF) Test

The ADF test is a formal statistical test used to check for stationarity.

- **Test Hypotheses:**
  - Null Hypothesis (H_0): The time series has a unit root (non-stationary).
  - Alternate Hypothesis (H_1): The time series is stationary.
- **Test Outputs:**
  - Test Statistic: Indicates how far the data is from being non-stationary.
  - p-value: A low p-value (typically < 0.05) suggests rejection of the null hypothesis.
  - # Lags Used: Number of lags included in the test.
  - Number of Observations Used: Data points used in the test.
  - Critical Values: Threshold values at 1%, 5%, and 10% significance levels.

Plot Description: Rolling Mean & Standard Deviation

This plot displays the original time series data along with its rolling mean and rolling standard deviation over time. It is a common technique used in time series analysis to visually assess the stationarity of the data.

Key Elements:

- Blue Line: Original Data: This line represents the actual 'Price' data plotted against time, spanning from approximately early 2014 to late 2023/early 2024. The original data exhibits fluctuations and appears to have some level of trend, particularly showing a significant upward movement starting around 2019/2020. There also seem to be some cyclical patterns or seasonality present.

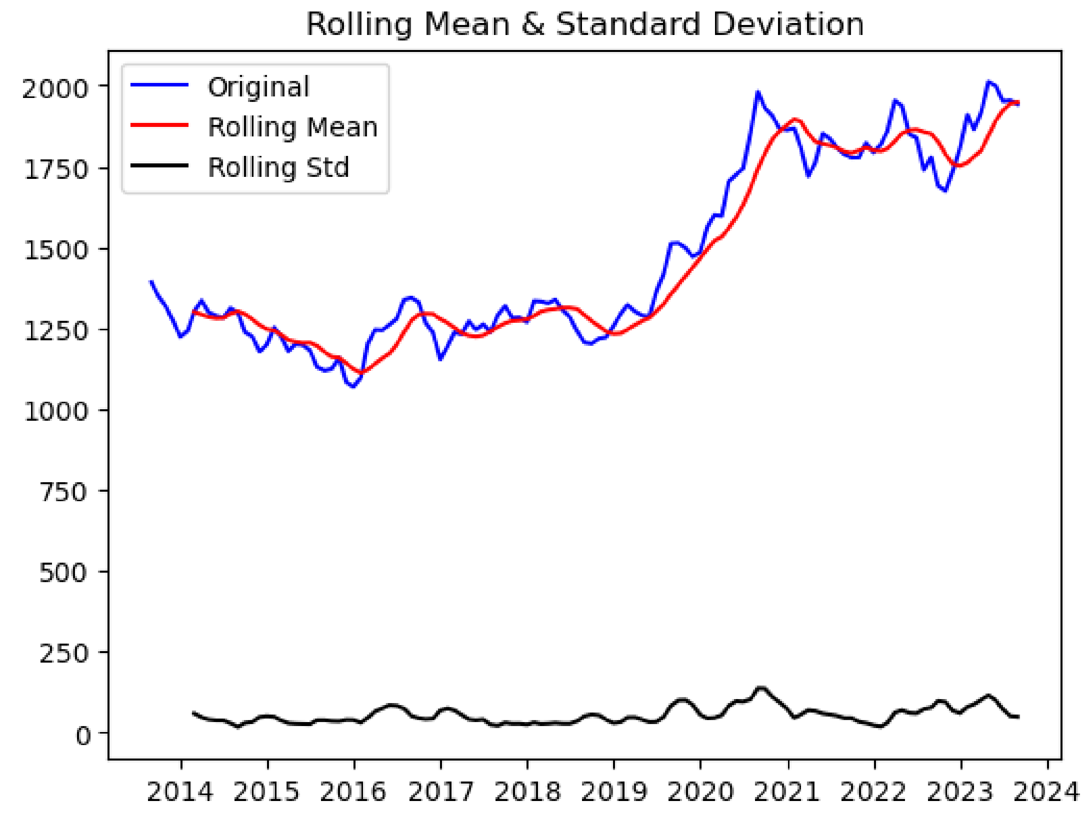
- Red Line: Rolling Mean: This line represents the rolling mean of the 'Price' data over a specified window (the window size is not explicitly stated in the plot but is implied by the smoothness of the line; typically it's a few months to a year for such data).
    - Initially, the rolling mean fluctuates relatively around the 1200-1300 level.
    - Starting around 2019/2020, the rolling mean shows a clear upward trend, mirroring the increase in the original data.
    - The rolling mean tends to smooth out the short-term fluctuations present in the original data, highlighting the underlying trend.
    - If the time series were strictly stationary, the rolling mean would ideally be a horizontal line over time. The visible trend in the rolling mean suggests non-stationarity in the data's level.
- Black Line: Rolling Std (Rolling Standard Deviation): This line represents the rolling standard deviation of the 'Price' data over the same window as the rolling mean. The rolling standard deviation measures the volatility or the amount of dispersion in the data around its mean over time.
    - The rolling standard deviation appears to fluctuate, indicating that the volatility of the 'Price' data is not constant over the entire period.
    - Periods of higher volatility (larger deviations from the mean) seem to occur around the times of significant price changes or fluctuations in the original data. For example, there seems to be increased volatility around 2016, late 2019/early 2020, and in the 2021-2023 period.
    - If the time series were strictly stationary, the rolling standard deviation would ideally be a relatively constant, horizontal line. The changes in the rolling standard deviation suggest non-constant variance, which is another form of non-stationarity (heteroscedasticity).
- X-axis: The x-axis represents time, marked with years from 2014 to 2024.
- Y-axis: The y-axis represents the 'Price' value.
- Legend: The legend clearly labels the blue line as "Original," the red line as "Rolling Mean," **and the black line as "Rolling Std."**

Interpretation for Stationarity:

Based on this plot:

- The upward trend in the rolling mean indicates that the level of the time series is changing over time, suggesting that the data is not stationary in terms of its mean.
- The fluctuations in the rolling standard deviation suggest that the variance of the time series is also changing over time, indicating that the data is not stationary in terms of its variance.

Therefore, the visual analysis of the rolling mean and rolling standard deviation plots suggests that the 'Price' data is non-stationary. This non-stationarity often needs to be addressed (e.g., through differencing or other transformations) before applying certain time series models like ARIMA.



Test Results:

Metric	Value
Test Statistic	-0.5696
p-value	0.8777
Number of Lags Used	1
Number of Observations Used	119
Critical Value (1%)	-3.4865
Critical Value (5%)	-2.8862
Critical Value (10%)	-2.5799

Interpretation and Conclusion

- The **Test Statistic** (-0.5696) is **greater than** all the critical values (-3.4865, -2.8862, -2.5799).
- The **p-value** (0.8777) is **significantly greater** than the 0.05 threshold.

Conclusion:
We **fail to reject** the null hypothesis. This implies that the time series is **non-stationary**.
To proceed with modeling, we will need to **transform or difference** the series to achieve stationarity.

Analysis of Rolling Statistics and Dickey-Fuller Test

1. Rolling Mean & Standard Deviation Plot:

- Blue Line: Original Data: The original time series data (blue line) shows fluctuations around a central level, without a clear upward or downward trend over the observed period (approximately 2014 to early 2021). There are noticeable ups and downs, suggesting some inherent variability.
- Red Line: Rolling Mean: The rolling mean (red line) appears to be relatively stable over time, fluctuating within a narrower range compared to the original data. It doesn't exhibit a strong increasing or decreasing trend. This suggests that the average level of the time series might be consistent over the long term.
- Black Line: Rolling Std (Rolling Standard Deviation): The rolling standard deviation (black line) also seems relatively stable, indicating that the volatility or spread of the data around its mean is not drastically changing over time. There might be some minor fluctuations, but no clear increasing or decreasing pattern is evident.

Visual Interpretation for Stationarity:

The relatively stable rolling mean and rolling standard deviation suggest that the time series might be stationary. The absence of a significant trend in the rolling mean indicates a constant average level, and the consistent rolling standard deviation suggests constant variance.

2. Results of Dickey-Fuller Test:

- Test Statistic: -6.966787e+00 (-6.967)
  - This is the calculated test statistic, which is a large negative value.
- p-value: 8.872747e-10 (approximately 0.000000000887)
  - The p-value is extremely small, very close to zero.
- #Lags Used: 0.000000e+00 (0)
  - This indicates that 0 lags were used in the Augmented Dickey-Fuller test, meaning it's a standard Dickey-Fuller test.
- Number of Observations Used: 8.700000e+01 (87)
  - This is the number of data points used in the test.
- Critical Value (1%): -3.507853e+00 (-3.508)
  - The critical value at the 1% significance level.
- Critical Value (5%): -2.895382e+00 (-2.895)
  - The critical value at the 5% significance level.
- Critical Value (10%): -2.584824e+00 (-2.585)

- The critical value at the 10% significance level.

Conclusion based on the Dickey-Fuller test results:

- Comparison with Critical Values: The Test Statistic (-6.967) is significantly less than all the critical values (-3.508, -2.895, -2.585) at the 1%, 5%, and 10% significance levels.
- Comparison with p-value: The p-value (8.87e-10) is much smaller than any conventional significance level (0.01, 0.05, 0.10).

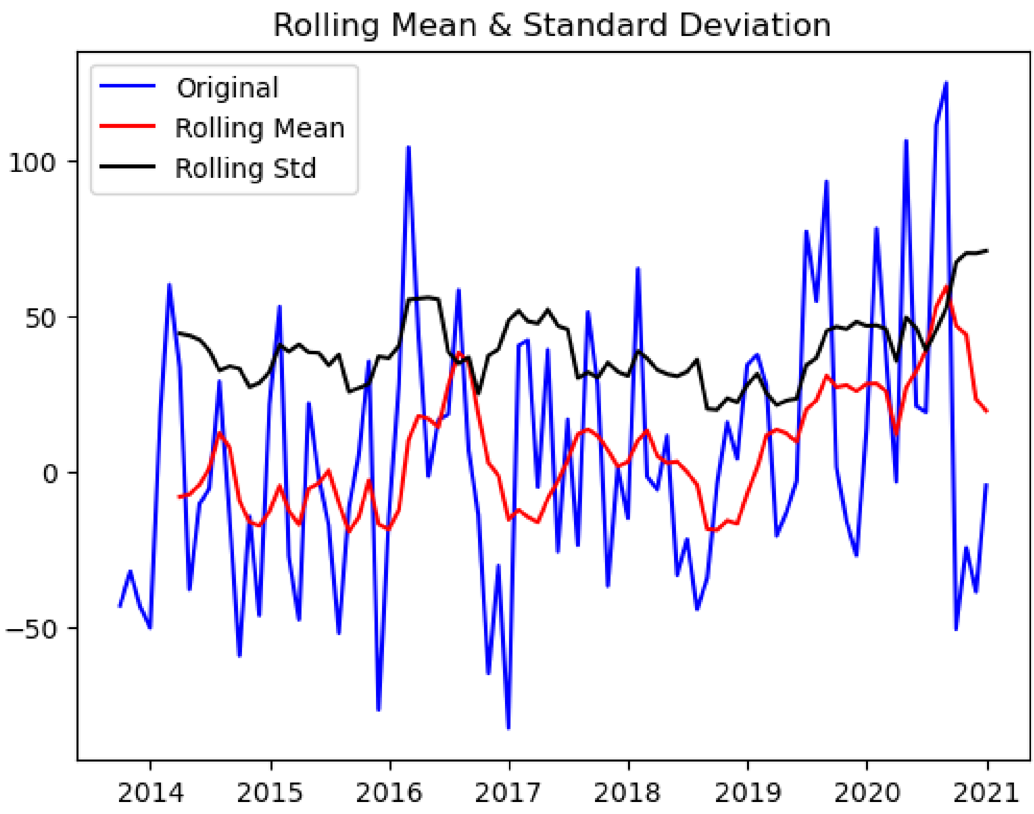
Interpretation:

Since the Test Statistic is significantly less than the critical values, and the p-value is much smaller than the chosen significance levels, we reject the null hypothesis that the time series has a unit root and is non-stationary.

Therefore, based on the Dickey-Fuller test results, there is strong statistical evidence to suggest that the time series is stationary.

Overall Conclusion:

Both the visual analysis of the rolling mean and standard deviation plot and the results of the Dickey-Fuller test indicate that this time series is likely stationary. The rolling statistics are relatively stable, and the Dickey-Fuller test strongly rejects the null hypothesis of non-stationarity. This means that you can likely apply time series models that assume stationarity (like AR or MA components of ARIMA) directly to this data without the need for differencing.

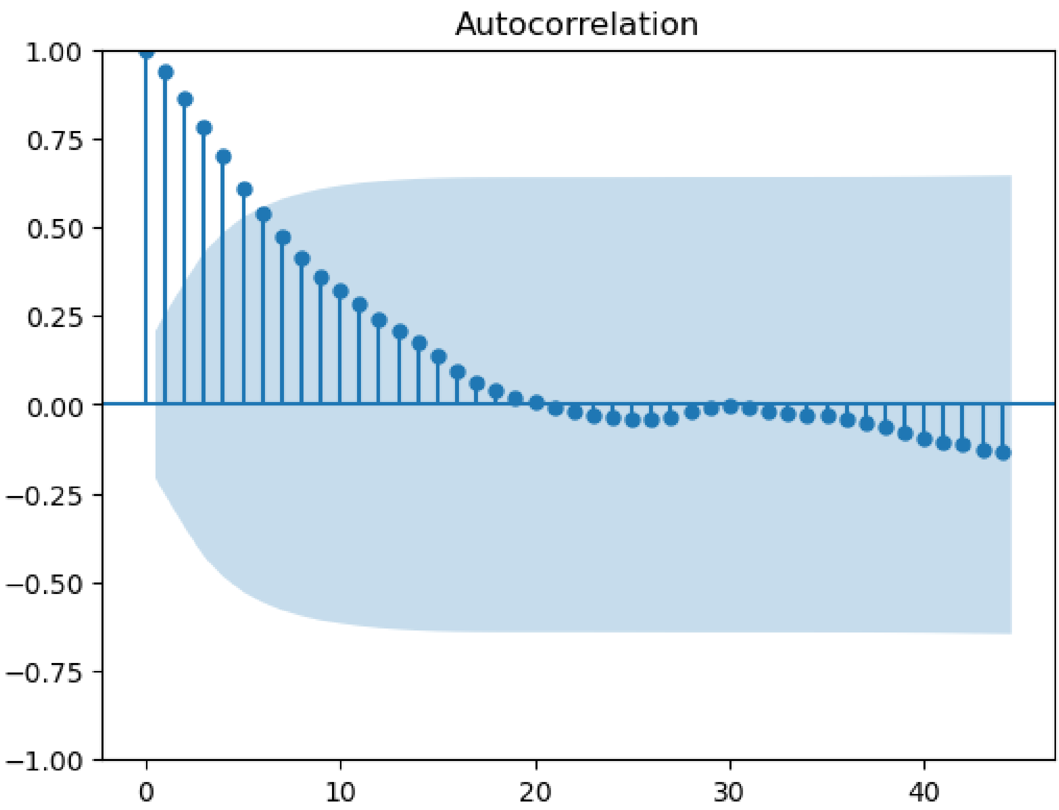


Autocorrelation

This plot displays the autocorrelation coefficients of a time series at different lags. The x-axis represents the lag number, and the y-axis represents the autocorrelation coefficient, which ranges from -1 to 1. The blue shaded area typically indicates the confidence interval (usually at a 95% level). Autocorrelation values outside this interval are considered statistically significant.

Key Observations:

- Lag 0: The autocorrelation at lag 0 is always 1.0, as a time series is perfectly correlated with itself. This is clearly visible as the first bar on the left reaches the top of the y-axis.
- Significant Positive Autocorrelation at Early Lags: The autocorrelation coefficients at the first several lags (lag 1, 2, 3, and so on) are strongly positive and significantly outside the blue shaded confidence interval. This indicates a strong positive correlation between the current value of the time series and its past values. For example, if the value at time *t* is high, the values at *t-1*, *t-2*, etc., are also likely to be high.
- Gradual Decay of Autocorrelation: As the lag increases, the magnitude of the autocorrelation coefficients gradually decreases. The bars become shorter, indicating a weaker correlation with more distant past values. This is a common pattern in time series data that exhibits some form of dependence.
- Autocorrelation Remains Significant for a Number of Lags: The autocorrelation coefficients remain statistically significant (outside the blue shaded area) for a considerable number of lags (approximately up to lag 20 or even beyond). This suggests that past values have a long-lasting influence on the current values of the time series.
- Eventual Decay into the Confidence Interval: Eventually, as the lag continues to increase (beyond roughly lag 20-25), the autocorrelation coefficients fall within the blue shaded confidence interval. This implies that for these larger lags, the correlation between the current value and the past values is no longer statistically significant.
- No Strong Negative Autocorrelation: There are no large negative spikes in the ACF plot. This suggests that there is no strong tendency for high values to be followed by very low values (or vice versa) at any specific lag.



Differenced Data PACF

This plot displays the partial autocorrelation coefficients of the differenced time series at different lags. The x-axis represents the lag number, and the y-axis represents the partial autocorrelation coefficient, ranging from -1 to 1. The blue shaded area typically indicates the confidence interval (usually at a 95% level). Partial autocorrelation values outside this interval are considered statistically significant. The title "Differenced Data PACF" indicates that this analysis was performed after applying a differencing operation to the original time series, likely to induce stationarity.

Key Observations:

- Lag 0: The partial autocorrelation at lag 0 is always 1.0. This is shown by the initial spike.
- Significant Positive Partial Autocorrelation at Lag 1: There is a significant positive spike at lag 1, extending well above the upper confidence limit. This suggests that the value at the current time *t* has a direct positive correlation with the value at *t-1*, after removing the effects of the intervening lags.
- No Other Immediately Significant Spikes at Early Lags (Lags 2-5): The partial autocorrelation coefficients at lags 2, 3, 4, and 5 appear to be within or very close to the confidence interval, suggesting that once the direct effect of lag 1 is accounted for, there is no significant direct correlation at these shorter lags.
- Potential Significant Spikes at Later Lags (Around Lags 12, 18, 35, 36, 39): There seem to be some spikes that extend slightly outside the confidence interval at lags around 12, 18, 35, 36,

and 39. These could indicate potential direct correlations at these specific longer lags even after accounting for the shorter lags. However, it's important to be cautious with interpreting isolated spikes at higher lags, as they could sometimes occur by chance.

- Generally Within the Confidence Interval After Lag 1 (with the noted exceptions): For most lags beyond lag 1, the partial autocorrelation coefficients fall within the blue shaded confidence interval. This suggests that after accounting for the direct effect of the first lag, the remaining direct correlations at other lags are generally not statistically significant.

Interpretation for Time Series Model Identification (ARIMA):

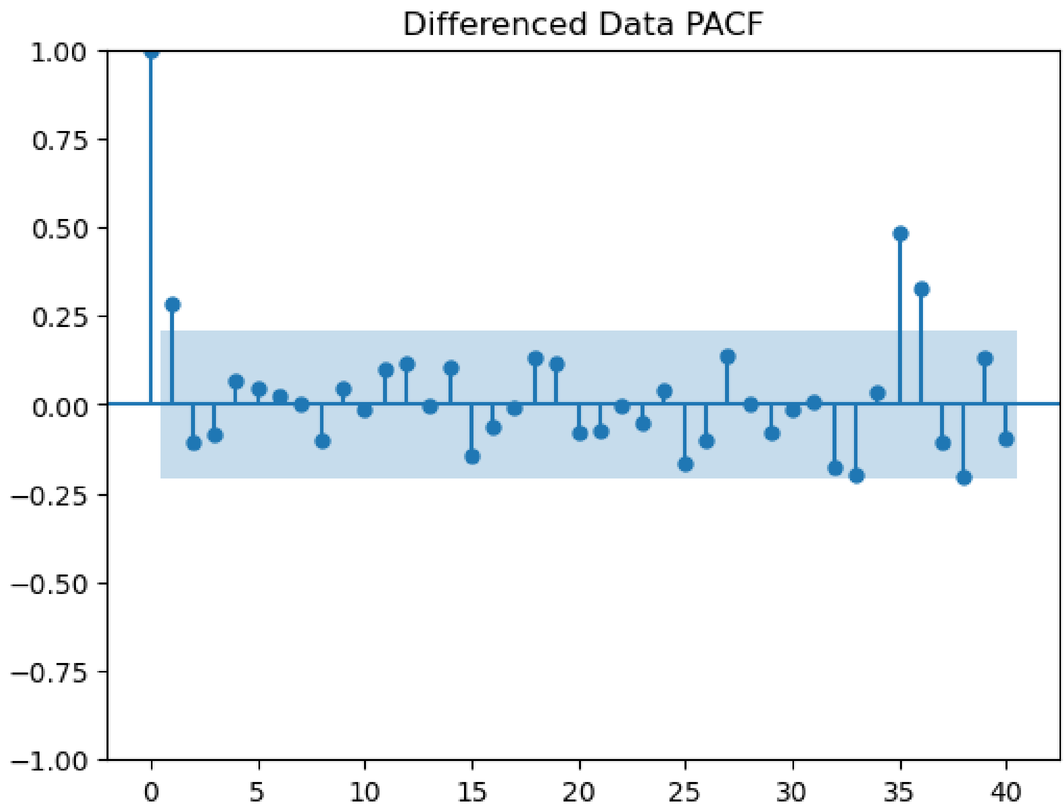
The PACF plot of the differenced data helps in identifying the order of the Autoregressive (AR) component (p) in an ARIMA(p, d, q) model.

- Significant Spike at Lag 1: The significant positive spike at lag 1 suggests that there is a direct relationship between the current differenced value and the immediately preceding differenced value. This points towards an AR(1) component (p=1) in the model for the differenced data.
- Lack of Significant Spikes at Early Higher Lags (2, 3, etc.): The absence of significant spikes at the next few lags (after lag 1) in the PACF reinforces the idea that the direct dependence primarily extends to the first lag in the differenced series.
- Potential Significance at Longer Lags (Cautious Interpretation): The spikes around lags 12, 18, 35, 36, and 39 might suggest some underlying periodic or lagged effects that are still present in the differenced data. If these are truly significant and not just random noise, they could indicate the need for higher-order AR terms or the presence of some remaining seasonality that was not fully addressed by simple differencing. However, without more context about the data's frequency and potential seasonal patterns, it's difficult to definitively interpret these later spikes.

In the context of ARIMA modeling after one differencing (d=1):

The significant spike at lag 1 in the PACF of the differenced data strongly suggests considering an ARIMA(1, 1, q) model. The order of the Moving Average (MA) component (q) would then be determined by analyzing the Autocorrelation Function (ACF) of the differenced data (which you provided in the previous turn).

The potential significance at longer lags warrants further investigation. If there is a known seasonal period (e.g., 12 for monthly data with annual seasonality), the spikes around lag 12 and its multiples could be relevant.



Auto ARIMA for Forecasting

After confirming that the original time series is non-stationary, we performed first-order differencing (d=1) to induce stationarity. The next step involved selecting optimal parameters for an ARIMA (p,d,q) model based on model evaluation criteria.

2.1 Parameter Grid for ARIMA

To find the best ARIMA model, a grid search was performed over various combinations of parameters:

p (autoregressive term): 0 to 2

d (differencing): 1 (to induce stationarity)

q (moving average term): 0 to 2

Total combinations evaluated: 9 (from (0,1,0) to (2**,1,2))**

2.2 Model Evaluation Criterion: AIC

Each model was trained on the training set (train['Price']) using the ARIMA model from statsmodels.

The models were compared using the Akaike Information Criterion (AIC):

Lower AIC indicates a better-fitting model.

2.3 AIC Results for Each Model

Model (p,d,q)	AIC Score
(0, 1, 0)	914.26
(0, 1, 1)	908.15
(0, 1, 2)	909.80
(1, 1, 0)	908.40
(1, 1, 1)	909.95
(1, 1, 2)	911.37
(2, 1, 0)	909.69
(2, 1, 1)	911.57
(2, 1, 2)	912.92

2.4 Best ARIMA Model Selection

The model with the lowest AIC value was selected as the best:

ARIMA(0, 1, 1) with AIC = 908.15

This model balances complexity and goodness-of-fit better than others in the tested range.

Fitting the Best ARIMA Model and Evaluating Results

After identifying the best-performing ARIMA model based on the AIC score, we fit this model on the training data and evaluated its statistical summary.

Best ARIMA Model Selected

- Based on the lowest AIC value (908.15), the selected model was:
    - **ARIMA(0, 1, 1)**

Model Summary and Coefficient Interpretation

Below is the summary output from the fitted model:

Parameter	Coefficient	Std. Error	z-Statistic	p-value	95% CI
ma.L1 (MA(1))	0.2937	0.111	2.637	0.008	[0.075, 0.512]
$sigma^2$	1694.58	243.03	6.973	0.000	[1218.26, 2170.91]

- The **MA(1)** component is statistically significant (**p < 0.01**), indicating that the model effectively captures short-term shocks in the data.
- The **sigma²** value reflects the variance of the residuals.

Diagnostic Tests

- **Ljung-Box Test**:
    - Q-statistic = 0.01, p-value = 0.94
    - Suggests that residuals are not autocorrelated, which is a good sign.
- **Jarque-Bera Test**:
    - JB-statistic = 2.45, p-value = 0.29
    - Indicates that residuals are normally distributed (as p > 0.05).
- **Heteroskedasticity Test**:
    - H = 1.79, p-value = 0.12
    - No significant evidence of heteroskedasticity.

Conclusion:

All diagnostics indicate that the **ARIMA(0, 1, 1)** model is well-fitted to the training data and meets key assumptions.

Here's how to write this final part of your **ARIMA model section** for your report, including forecasting and model performance evaluation:

Forecasting and Performance Evaluation using ARIMA

Once the best ARIMA model was identified and fitted on the training dataset, the model was used to make predictions on the test set. The performance was then evaluated using **Root Mean Squared Error (RMSE)**.

Forecasting with ARIMA(0, 1, 1)

- The selected **ARIMA(0, 1, 1)** model was used to forecast prices for the same number of periods as the test set.
- Forecasting was done using the forecast() method from the fitted model object.

Forecast Accuracy: RMSE

- The **Root Mean Squared Error (RMSE)** was used to evaluate the prediction accuracy on the test set:

- RMSE = 90.22
- **Interpretation**:
  - The RMSE indicates that the model's predictions deviate from the actual prices by an average of about **₹90.22**.
  - Given the scale of the gold price series, this error is **moderate** and indicates **reasonable forecasting performance** for a simple ARIMA model.

Summary of ARIMA Modeling Approach

Step	**Description**
Stationarity Check	Used ADF test; series was non-stationary
Differencing Applied	First-order differencing (d=1)
Parameter Selection	Grid search over (p,d,q) using AIC
Best Model	ARIMA(0, 1, 1) with AIC = 908.15
Forecast Evaluation Metric	RMSE = 90.22 on test set

Let me know if you want to add a comparison with other models like **SES, DES, TES, or Linear Regression**, or want help generating charts or visuals!

Model Comparison and Forecasting Performance

To evaluate the effectiveness of various forecasting models, we compared their performance based on Root Mean Squared Error (RMSE) on the test set. Lower RMSE indicates better prediction accuracy.

RMSE Comparison Across Models

Model Name	RMSE
2-point Trailing Moving Average	27.94
4-point Trailing Moving Average	54.62
6-point Trailing Moving Average	70.89
9-point Trailing Moving Average	85.55
Simple Exponential Smoothing (α=0.995)	89.65
Double Exponential Smoothing (α=0.9, β=0.3)	90.08
ARIMA (0, 1, 1)	90.22

Triple Exponential Smoothing (α=0.676, β=0.088, γ=0.323) 527.40

Linear Regression 207.87

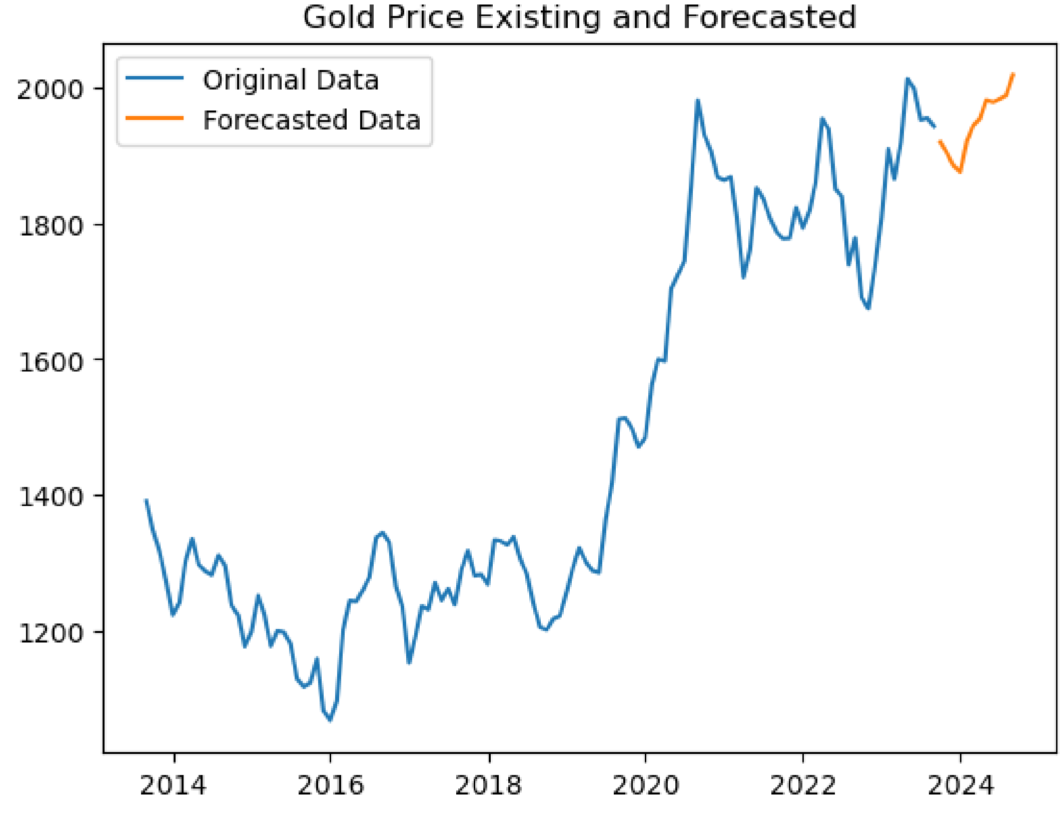
Key Observations

The 2-point Trailing Moving Average model delivered the lowest RMSE (27.94), suggesting strong performance for short-term patterns.

ARIMA (0, 1, 1) showed a moderate RMSE of 90.22, comparable to smoothing models like SES and DES, and significantly better than Linear Regression.

Triple Exponential Smoothing (TES) performed poorly with a very high RMSE, indicating that it may not suit this dataset.

Linear Regression had the highest RMSE (207.87), suggesting it's not well-suited for this time series forecasting task.



Conclusion

While ARIMA(0, 1, 1) performed reasonably well and adheres to statistical assumptions, simple moving averages, especially the 2-point trailing moving average, provided the most accurate short-term forecasts. However, for more complex or long-range forecasting, ARIMA might be preferable due to its robustness and interpretability.