Gold Price Prediction Using Time Series

by

Jainisha Choksi (MST03-0039)

Submitted to

Scifor Technologies

Meta Scifor Technologies

Script. Sculpt. Socialize

Under Guidance of

Ms.Urooj Khan

TABLE OF CONTENT

Abstract

This project aims to predict gold prices using historical data from web scrapping and implementing various forecasting models. After collecting data from a reputable financial website, we conducted exploratory data analysis(EDA) to uncover insights and patterns in the gold price time series. We then implemented several forecasting models, including ARIMA and Seasonal ARIMA models, to predict future gold prices. Our findings suggest modeling techniques, providing valuable insights for investors and stakeholders in the financial markets.

Keywords: Gold Price, ARIMA, SARIMA

INTRODUCTION

Gold has been a crucial commodity throughout history, serving as a store of value, a medium of exchange, and a hedge against economic uncertainty. As one of the oldest forms of currency, its value is influenced by many factors, including economic indicators, geopolitical events, and market sentiment. Given its significance in the global economy, accurately predicting gold prices has long interested investors, traders, and policymakers alike.

This report uses historical data from web scrapping techniques to predict gold prices. Before delving into the prediction models, it's essential to understand the various components of the gold price dataset and how they relate to each other. The dataset typically contains several key variables, each providing valuable insights into the behavior of gold prices over time.

1. Close Price: The closing price represents the final traded price of gold at the end of a trading session. It is one of the most widely used metrics for assessing the performance of an asset over a specific period. In the context of gold price prediction, the close price serves as the target variable that we aim to forecast accurately.
2. Open Price: The opening price refers to the first gold price traded at the beginning of a trading session. It provides valuable information about market sentiment and investor expectations at the start of the trading day.
3. High Price: The high price represents the highest trade price of gold during a particular trading session. It indicates the maximum price level reached by gold within the given time frame and can help identify potential resistance levels in the market.
4. Low Price: The low price is the lowest trade price of gold during a specific trading session. It reflects the minimum price level reached by gold and can help identify support levels in the market.
5. Volume: Volume refers to the total number of shares or contracts traded during a given period. In gold trading, volume represents the amount of gold exchanged hands within a specified time frame. High

volume often accompanies significant price movements, indicating increased market participation.

6. Change: The change represents the difference between the current and previous closing prices. It provides insight into the direction and magnitude of price movements, allowing investors to assess the performance of gold over time.

TECHNOLOGY USED

Programming Language: Python
Machine Learning Algorithms:

- Linear Regression
- Decision Tree
- Random Forest

Time Series Machine Learning Algorithms:

- ARIMA
- SARIMA

Libraries:

- Yfinance: Finance Dataset
- Streamlit: Deployment of Website
- Numpy: Mathematical Operations
- Pandas: Handling Dataset
- Matplotlib: Data Visualization
- Sklearn: Machine Learning Algorithms
- Statsmodels: Time Series Algorithms

DATASET INFORMATION

The dataset we used here for Gold Price prediction is from investing.com. It has columns like Closing Price, Opening Price, High Price, Low Price, Change, and Volume. The data collected was from 01st January 2010 to 08th April 2024.

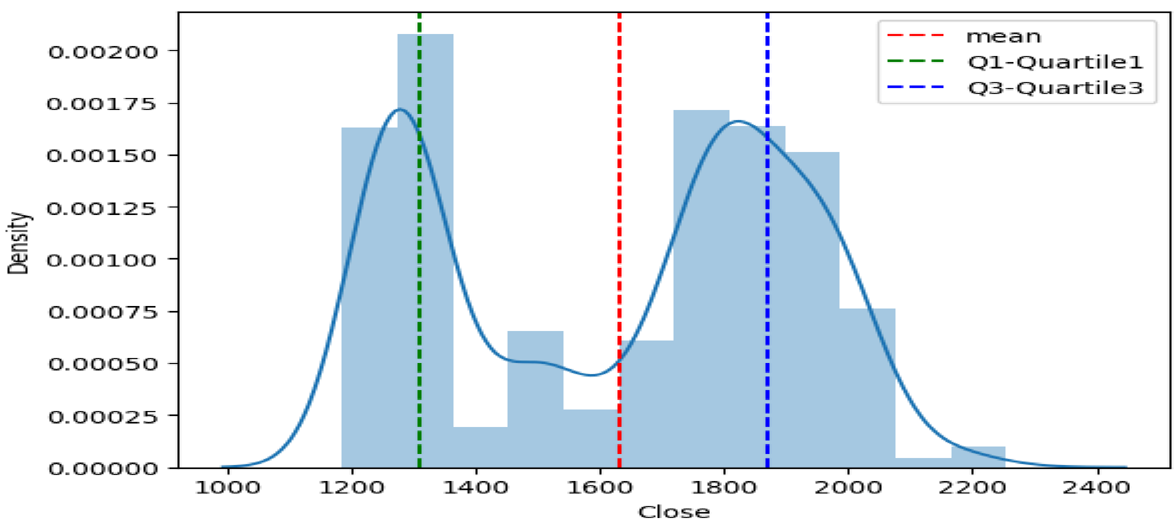
METHODOLOGY

Preprocessing:

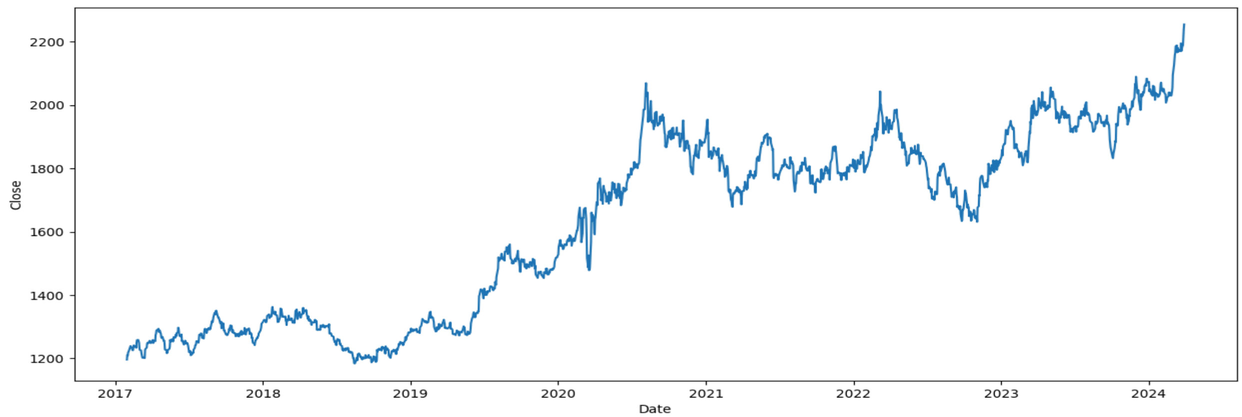
Column Naming: We renamed columns to ensure clarity and consistency after extracting the data. Common column names include "Date," "Close," "Open," "High," "Low," "Volume," and "Change."
Formatting Dates: Dates were formatted into a standard datetime format to facilitate time series analysis. This involved converting date strings into datetime objects and ensuring uniformity across the dataset.
Type Changing: We ensured appropriate data types for each column, such as numeric types for price and volume columns and datetime types for the date column.
Close Column Analysis: Recognizing the significance of the close price in gold price prediction, special attention was given to this column during preprocessing. Outliers, missing values, and inconsistencies were addressed to ensure the integrity of the close price data.

Exploratory Data Analysis(EDA):

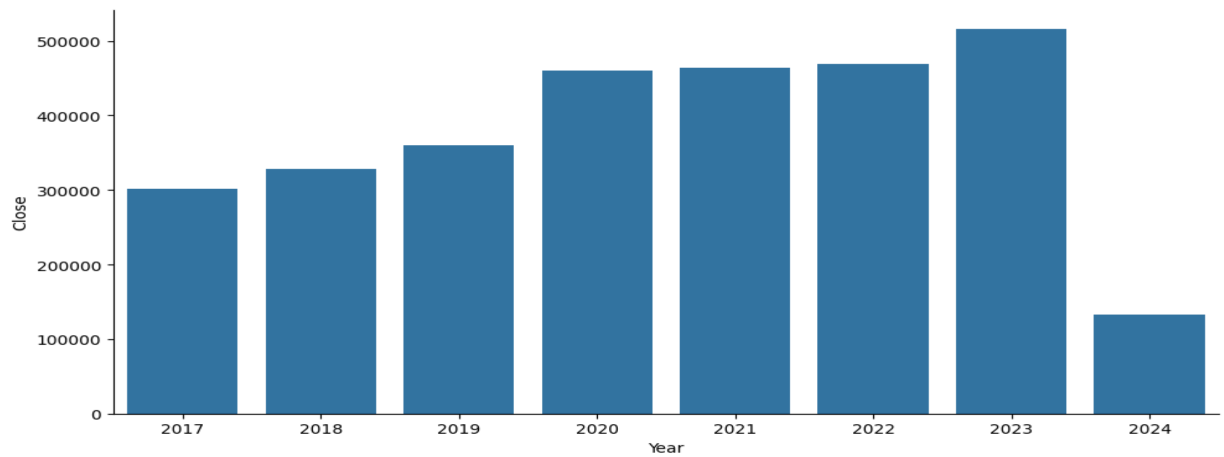
Checking for Normality: We conducted tests to assess the normality of the close price distribution, including graphical methods such as histograms.



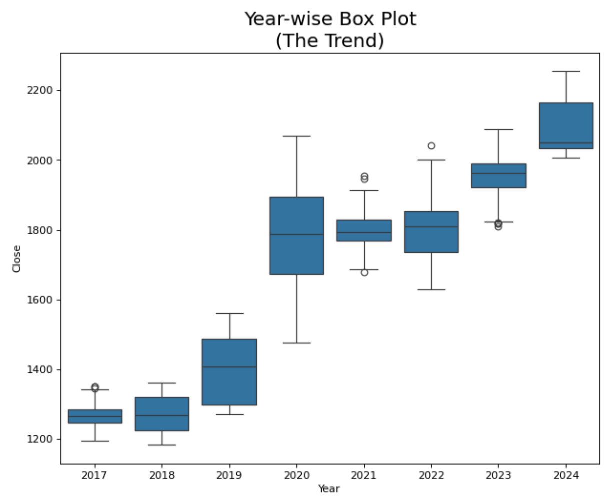
Checking for Trends in Data: Time series plots were created to visualize trends, seasonality, and irregularities in the gold price data over time. Trend analysis involves identifying long-term patterns or movements in the data.

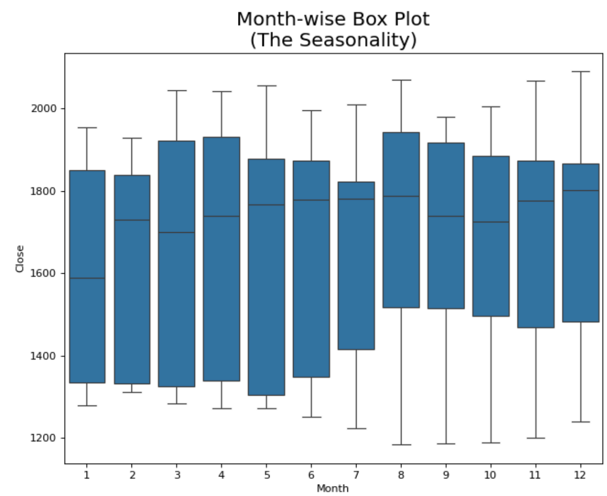


Yearly Sales: We calculated yearly gold price averages to observe overall trends and fluctuations in gold prices across different years.

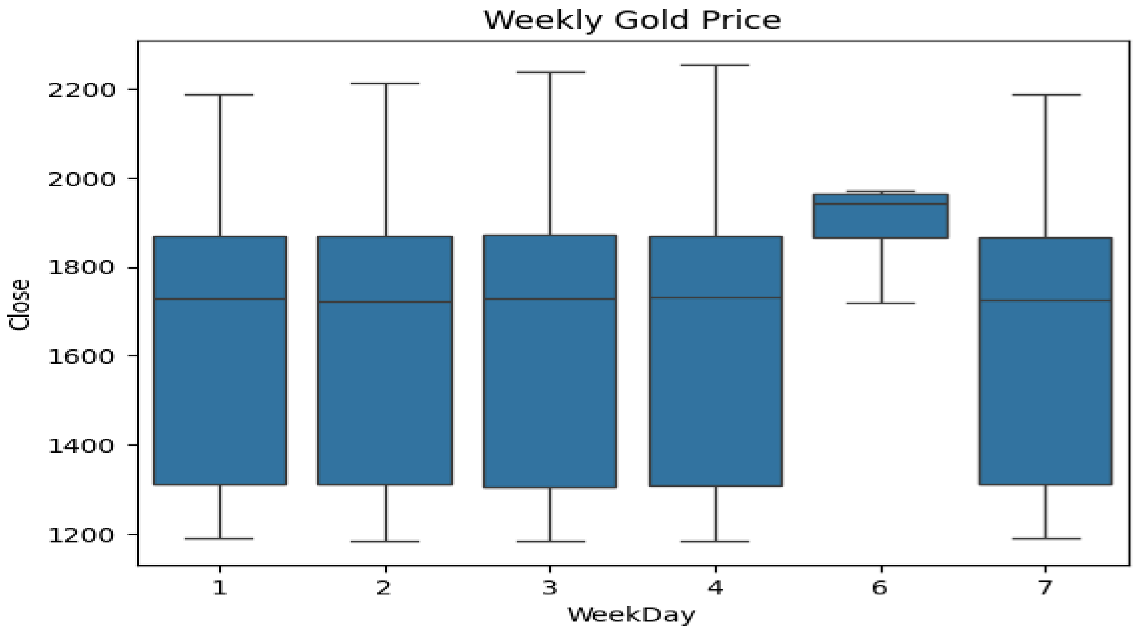


Year-wise and Month-wise Price Averages: Analyzing gold price averages on a year-wise and month-wise basis provided insights into seasonal variations and cyclicality in gold prices.

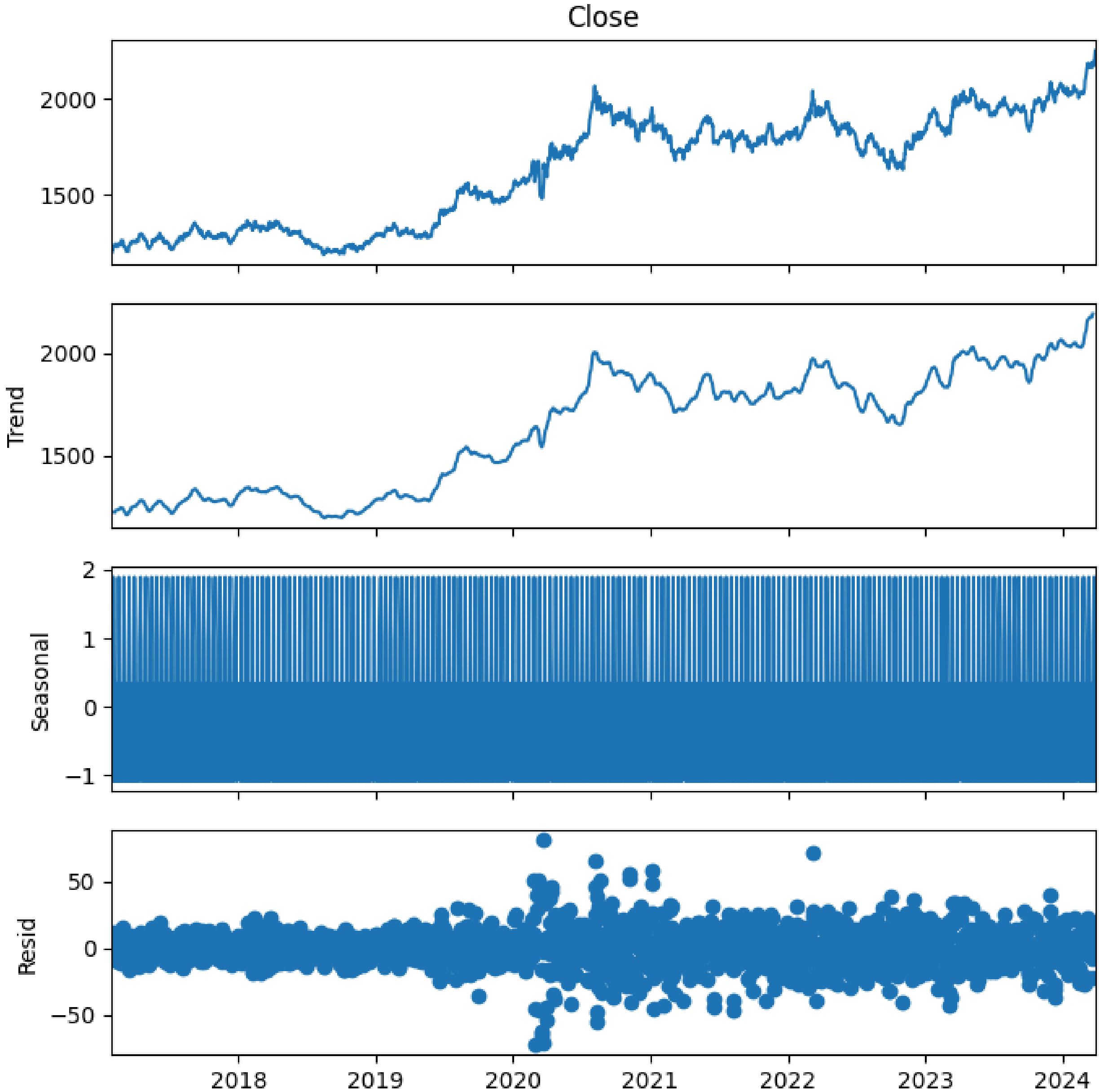




Weekly Gold Price Plot: Weekly gold price plots were generated to identify intra-week patterns and detect any recurring patterns or anomalies.



Time Series Decomposition: Time series decomposition techniques such as seasonal decomposition of time series (STL) were applied to decompose the gold price data into its trend, seasonal, and residual components, aiding in understanding underlying patterns and identifying seasonality.



Methods:

Naive Method: This method is like the naive method but predicts the last observed value of the same year's season. This method works for highly seasonal data. Before diving into sophisticated algorithms, the time series data must be plotted to gain intuition and make direct predictions. Several trends occur in time series analysis; they are seasonal trends(increase or decrease at equal intervals and are associated with some aspect of the calendar) and cyclical (increase or decrease at irregular intervals); cyclical trends can be observed in the stock market where bull market is an uptrend and bear market represents the downtrend. Cyclical patterns are tough to predict as they could be very random.

Before we get to the more advanced time-series forecasting methods, let's look at a basic method - Seasonal Naive. It can serve as a quick calculation to get a baseline until something better can come. Or, perhaps there is very little variance in the data, so that this method can be good enough.

It is a naive method that considers the patterns by looking at what happened the same time the previous day. For example, if we want to predict sales for January 2021, the naive method will assume the previous sales for the previous day. Fortunately, we have at least one year of sales data, so this method might not make sense otherwise.

ARIMA and SARIMA:

ARIMA:

ARIMA models are flexible and widely used in time series analysis. ARIMA combines three processes: autoregressive (AR), differencing to strip off the integration (I) of the series, and moving averages (MA). Each of the three process types has its characteristic way of responding to a random disturbance. Autoregressive integrated moving average (ARIMA) models predict future values based on past values. ARIMA makes use of lagged moving averages to smooth time series data. They are widely used in technical analysis to forecast future security prices. The practical difference is that ARIMA packages are built to assume time series data, whereas most regression packages make no special allowances for time-dependent data. The ARIMA model uses differenced data to make the data stationary, which means there's data consistency over time. This function removes the effect of trends or seasonality, such as market or economic data.

Step 1: Check stationarity

Before going further into our analysis, our series must be made stationary.

Stationarity is the property of exhibiting constant statistical properties (mean, variance, autocorrelation, etc.). If the mean of a time series increases over time, then it's not stationary.

The mean across many periods is only informative if the expected value is the same. If these population parameters can vary, what are we estimating by taking an average across time?

Stationarity requires that the statistical properties be the same across time, making the sample average a reasonable way to estimate them.

Methods to Check Stationarity
Plotting rolling statistics: Plotting rolling means and variances is the first good way to visually inspect our series. Suppose the rolling statistics exhibit a clear trend (upwards or downwards) and varying variance (increasing or decreasing amplitude). In that case, you might conclude that the series is very likely not to be stationary.

Augmented Dickey-Fuller Test: This test is used to assess whether or not a time series is stationary. It gives a result called a "test statistic," based on which you can say, with different levels (or percentages) of confidence, if the time series is stationary. The test statistic is expected to be negative; therefore, it has to be more negative(less) than the critical value for the hypothesis to be rejected and conclude that the series is stationary.

ACF and PACF plots: An autocorrelation (ACF) plot represents the autocorrelation of the series with lags of itself. A partial autocorrelation (PACF) plot represents the amount of correlation between a series and a lag of itself that is not explained by correlations at all lower-order lags. Ideally, we want no correlation between the series and lags of itself. Graphically speaking, we would like all the spikes to fall in the blue region.

Step 2: Differencing
Differencing:
Seasonal or cyclical patterns can be removed by subtracting periodical values. If the data is 12-month seasonal, substracting the series with a 12-lag difference series will give a "flatter" series. Since we have aggregated the data for each day level, we will shift by 1.

Step 3: Model Building
Interpreting the AR(p), I(d), MA(q) values: Determining I(d):

Taking the first-order difference makes the time series stationary. Therefore, I(d) = 1.

Determining AR(p): If the lag-1 autocorrelation of the differenced series PACF is negative and/or there is a sharp cutoff, choose an AR order of 1.

From the PACF plot, we can observe that the AR is significant within six lags. Therefore, we can use AR(p) = 6 (6 lines crossed the blue lines, so six past days are required to predict).

Determining MA(q): If the lag-1 autocorrelation of the differenced series ACF is negative and/or there is a sharp cutoff, choose an MA order 1.

From the ACF plot, we see a negative spike at lag one therefore, we can use MA(q) = 1

Determine Error, Trend, and Seasonality
An ETS model has three main components: error, trend, and seasonality. Each can be applied either additively, multiplicatively, or not at all. We will use the above Times Series Decomposition Plot to determine the additive or multiplicative property of the three components.

Trend - If the trend plot is linear, we apply it additively (A). If the trend line grows or shrinks exponentially, we apply it multiplicatively (M). No trend component is included (N) if there is no clear trend.

Seasonal - If the peaks and valleys for seasonality are constant over time, we apply it additively (A). If the size of the seasonal fluctuations tends to increase or decrease with the time series level, we apply it multiplicatively (M). If there is no seasonality, it is not applied (N).

Error - If the error plot has constant variance over time (peaks and valleys are about the same size), we apply it additively (A). If the error plot fluctuates between large and small errors over time, we apply it multiplicatively (M).

We see a linear trend plot and a constant seasonality over time for our Gold price data so that we will apply trend and seasonality additively. Your data is nonstationary.

RESULT AND DISCUSSION

The performance of the gold price prediction models was evaluated using various accuracy metrics, including Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Error (ME), Mean Absolute Error (MAE), Mean Percentage Error (MPE), and Min-Max Error (MinMax). The results are summarized below:

Naive Method:

RMSE: 17.933
ARIMA Model:

RMSE: 96.811
Accuracy Metrics:

MAPE (Mean Absolute Percentage Error): 9.210
ME (Mean Error): 58.211
MAE (Mean Absolute Error): 149.551
MPE (Mean Percentage Error): 0.046
RMSE (Root Mean Squared Error): 203.208
Min-Max Error: 0.079

The performance of the gold price prediction models was evaluated using various accuracy metrics, including Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Error (ME), Mean Absolute Error (MAE), Mean Percentage Error (MPE), and Min-Max Error (MinMax). The results are summarized below:

Naive Method:

RMSE: 17.933
ARIMA Model:

RMSE: 96.811
Accuracy Metrics:

MAPE (Mean Absolute Percentage Error): 9.210
ME (Mean Error): 58.211
MAE (Mean Absolute Error): 149.551
MPE (Mean Percentage Error): 0.046

RMSE (Root Mean Squared Error): 203.208
Min-Max Error: 0.079

Date	Linear Regression Forecast	Decision Tree Forecast	Random Forest Forecast
2024-02-07 00:00:00	62,959	62,995.7402	62,982.7802
2024-04-06 00:00:00	71,856.7598	71,856.7598	71,737.4551
2024-02-14 00:00:00	61,596.2794	61,633.0196	61,936.0242
2024-02-27 00:00:00	62,792	62,995.7402	62,927.5371
2024-04-01 00:00:00	69,411.8824	70,437.2598	69,783.9896
2024-03-01 00:00:00	64,425.2598	64,425.2598	64,327.6315
2024-02-02 00:00:00	62,995.7402	62,995.7402	62,992.3666

ARIMA and SARIMA Forecast Comparison

	Date	ARIMA Forecast	SARIMA Forecast
0	2024-04-09 00:00:00	69,402.7255	69,411.8824
1	2024-04-10 00:00:00	69,396.0171	69,411.8824
2	2024-04-11 00:00:00	69,409.6768	69,411.8824
3	2024-04-12 00:00:00	69,411.5863	69,411.8824
4	2024-04-13 00:00:00	69,412.3414	69,411.8824
5	2024-04-14 00:00:00	69,412.5385	69,411.8824
6	2024-04-15 00:00:00	69,412.5979	69,411.8824

Machine Learning Models Evaluation Metrics

Linear Regression MAE: `1.0224487466205443e-07`

Linear Regression MSE: `1.494756382941233e-14`

Decision Tree MAE: `134.0770263671875`

Decision Tree MSE: `90924.16718418537`

Random Forest MAE: `107.1326958356606`

Random Forest MSE: `25376.2876856981`

The accuracy metrics for the ARIMA model demonstrate a mixed performance. While the MAPE of 9.210 suggests that, on average, predictions deviate by approximately 9.210 from the actual values, the ME of 58.211 indicates a bias in the predictions. The MAE of 149.551 represents the average absolute difference between predicted and actual values, while the MPE of 0.046 indicates a small percentage error on average. The RMSE of 203.208 reflects the model's overall accuracy, considering bias and variance. Additionally, the Min-Max Error of 0.079 signifies the range of errors observed across predictions relative to the actual range of gold prices.

CONCLUSION

In conclusion, while the Naive method and the ARIMA model offer valuable insights into gold price prediction, there is no one-size-fits-all approach. The choice of model depends on various factors, including the availability of data, the time horizon of predictions, and the desired level of accuracy. By critically evaluating the strengths and limitations of each approach, stakeholders can make informed decisions and better navigate the dynamic and ever-changing landscape of the gold market.

REFERENCES

- Xiaohui Yang, ”The Prediction of Gold Price Using ARIMA Model,” 2nd International Conference on Social Science, Public Health and Education 2019

- geeks for geeks/gold-price-prediction-using-machine-learning

- V. G. S and H. V. S, "Gold Price Prediction and Modelling using Deep Learning Techniques," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, 2020, pp. 28-31, doi: 10.1109/RAICS51191.2020.9332471. keywords: {Gold;Recurrent neural networks;Time series analysis;Predictive models;Market research;Data models;Investment;LSTM;Exponential;CNN;RNN;Prediction;Estimation},

- Investing.com Gold Historical Data