Financial Data Analysis - Gold Price Predictions in Sync with World Events

Abhijan Theja, Karan Reddy, Hitesh Shanmukha

Abstract

The project titled Financial Data Analysis - Gold Price Predictions in Sync with World Events aims to analyse the dynamic relationship between global events and fluctuations in gold prices. Gold has historically been a reliable asset and a "safe haven" during times of economic uncertainty, and its value is often influenced by global economic trends, geopolitical events, inflation rates, and currency fluctuations. This project seeks to use historical data, statistical analysis, and predictive modeling to forecast gold prices based on significant world events. The goal is to develop a comprehensive report that provides insights into the correlations between international affairs and gold market trends, helping investors and policymakers better understand the impact of global events on gold as a financial asset.

GitHub Repository

The full project and its resources are available at:
GitHub - Gold Price Predictions in Sync with World Events

Deployed Application

You can access the deployed application at:
Gold Price Predictor - Vercel Deployment

Contents

1 Introduction 2

2 Tasks Overview 3

3 Tasks Completed 5

3.1 Task 1: Peer Surveys 5

3.2 Task 2: Frontend and Backend Development 5

3.2.1 Frontend Development 5

3.2.2 Backend Development 6

3.2.3 Database Management 7

3.2.4 Cloud Infrastructure 7

3.3 Task 3: Technologies used 8

3.3.1 1 - Application of Docker 8

3.3.2 2 - Application of Elasticsearch 10
3.3.3 3 - Application of Apache Spark 10
3.4 Task 4 and Task 5: ML Model 10
3.4.1 Methodology 11
3.4.2 Data Analysis and Exploration 14
3.4.3 Feature Engineering: Correlation-Based Approach 17
3.4.4 Feature Engineering: PCA-Based Dimensionality Reduction . . . 18
3.4.5 Model Training and Evaluation 19

4 Conclusion 29

1 Introduction

Gold has been a critical asset in global financial markets, valued not only for its intrinsic worth but also as a hedge against market volatility and economic downturns. Over the years, various events—ranging from political conflicts and trade wars to economic crises and inflationary periods—have had a direct impact on the price of gold. This project, Financial Data Analysis - Gold Price Predictions in Sync with World Events, is focused on exploring how major global events influence gold prices, with the aim of creating predictive models that can assist stakeholders in anticipating price trends based on current and anticipated world events. This analysis involves three core components:

1. **Data Collection and Preprocessing**: We have gathered Historical gold price data, along with a timeline of significant global events (e.g., financial crises, geopolitical conflicts, policy changes), from reputable sources like kaggle,yfinance . Data cleaning and preprocessing was conducted to ensure accuracy and reliability.

2. **Correlation and Trend Analysis**: Using statistical tools, the project will examine correlations between gold price movements and key global events, allowing for an understanding of the underlying trends and triggers. This stage will also involve sentiment analysis and economic indicators to gauge how specific types of events affect the market.

3. **Predictive Modeling**: Leveraging machine learning techniques, models will be developed to forecast gold prices by recognizing patterns in the historical data. Time series analysis and regression models will be utilized to generate predictions, with a focus on refining accuracy through event-based variables.

4. **Technologies Used** Several cutting-edge technologies were utilized to ensure the project's scalability, efficiency, and robustness. These include:

   - **Docker:** Docker was used to containerize the project's components, ensuring consistency across development and deployment environments. It allowed seamless integration of modules like data preprocessing, machine learning models, and API services, streamlining the workflow.

   - **Elasticsearch:** Elasticsearch was implemented to index and query gold price predictions efficiently. This facilitated real-time querying and searching of historical and forecasted data, enhancing the backend's performance.

- **Apache Kafka:** Apache Kafka was leveraged for real-time data ingestion and streaming. It enabled the project to handle live gold price feeds and event updates, ensuring predictions remain current and actionable.
- **Apache Spark:** Spark was integrated for distributed data processing and machine learning. It accelerated the analysis of large datasets, including historical prices and event logs, and improved the efficiency of the predictive modeling pipeline.

By incorporating these technologies, the project achieved a robust architecture capable of handling vast datasets, real-time analytics, and scalable predictions. This approach ensures that the system is both efficient and adaptable to the evolving demands of financial forecasting.

The final report includes a detailed analysis, visualizations, and predictive insights to benefit stakeholders in navigating the complexities of the gold market.

2 Tasks Overview

The project involves a structured approach to accomplish various objectives. Below is a breakdown of the tasks and their respective weightage in the evaluation:

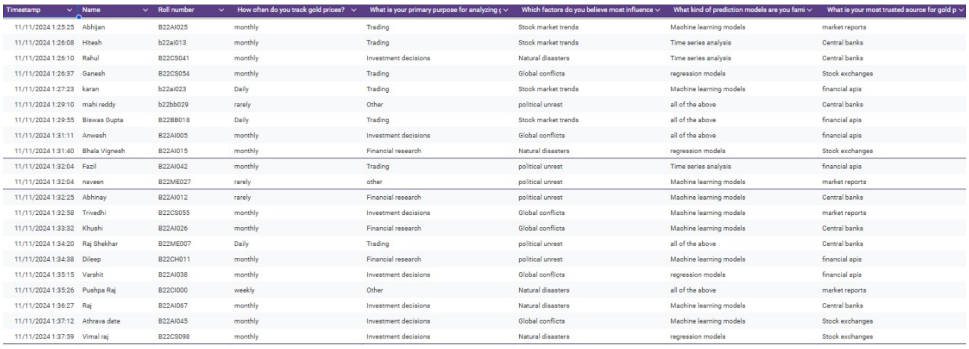
- **Demo of Project [25 points]**
  - Includes both front-end and back-end work, with primary emphasis on back-end technologies.
- **Report of Project [25 points]**
  - Comprising an explanation of what has been done, choice of technology including:
    - **Data Source:** Online and peer surveys.
    - **Data Cleaning Tools:** Docker, Apache Spark.
    - **Analysis Tools:** SQL/NoSQL databases, core system technical specifications, etc.
  - Includes justification and reasoning behind the choice of tools and technologies.
- **Analysis Results [15 points]**
  - Focus on the use of machine learning (ML) models for predictions.
  - Assessment of data biases or the lack thereof.
- **Data Visualization [15 points]**
  - Create impactful and intuitive visualizations to present insights and trends.
- **Novel Idea Explored [5 points]**
  - Development and explanation of a unique aspect or innovative solution incorporated into the project.

- **Comparison Between Existing Technology Solutions [5 points]**
  - Evaluation of existing approaches and their comparison with the implemented system.
- **Progress Consistency [10 points]**
  - Regular progress tracking and updates through Lab 3 to Lab 5.

3 Tasks Completed

3.1 Task 1: Peer Surveys

Using Google Forms, peer surveys were conducted to understand the factors influencing gold price predictions. Insights from the survey helped shape the scope and objectives of the project.



Timestamp	Name	Roll number	How often do you track gold prices?	What is your primary purpose for analyzing g	Which factors do you believe most influence	What kind of prediction models are you fami	What is your most trusted source for gold p
11/11/2024 1:25:25	Abhijan	B22AI025	monthly	Trading	Stock market trends	Machine learning models	market reports
11/11/2024 1:26:08	Hitesh	b22ai013	monthly	Trading	Stock market trends	Time series analysis	Central banks
11/11/2024 1:26:10	Rahul	B22CS041	monthly	Investment decisions	Natural disasters	Time series analysis	Central banks
11/11/2024 1:26:37	Ganesh	B22CS054	monthly	Trading	Global conflicts	regression models	Stock exchanges
11/11/2024 1:27:23	karan	b22ai023	Daily	Trading	Stock market trends	Machine learning models	financial apis
11/11/2024 1:29:10	mahi reddy	b22bb029	rarely	Other	political unrest	all of the above	Central banks
11/11/2024 1:29:55	Biswas Gupta	B22BB018	Daily	Trading	Stock market trends	all of the above	financial apis
11/11/2024 1:31:11	Anwesh	B22AI005	monthly	Investment decisions	Global conflicts	all of the above	financial apis
11/11/2024 1:31:40	Bhala Vignesh	B22AI015	monthly	Financial research	Natural disasters	regression models	Stock exchanges
11/11/2024 1:32:04	Fazil	B22AI042	monthly	Trading	political unrest	Time series analysis	financial apis
11/11/2024 1:32:04	naveen	B22ME027	rarely	other	political unrest	Machine learning models	market reports
11/11/2024 1:32:25	Abhinay	B22AI012	rarely	Financial research	political unrest	Machine learning models	Central banks
11/11/2024 1:32:58	Trivedhi	B22CS055	monthly	Investment decisions	Global conflicts	Machine learning models	market reports
11/11/2024 1:33:32	Khushi	B22AI026	monthly	Financial research	Global conflicts	Machine learning models	Central banks
11/11/2024 1:34:20	Raj Shekhar	B22ME007	Daily	Trading	political unrest	all of the above	Central banks
11/11/2024 1:34:38	Dileep	B22CH011	monthly	Financial research	political unrest	Machine learning models	financial apis
11/11/2024 1:35:15	Varshit	B22AI038	monthly	Investment decisions	Global conflicts	regression models	financial apis
11/11/2024 1:35:26	Pushpa Raj	B22CI000	weekly	Other	Natural disasters	all of the above	market reports
11/11/2024 1:36:27	Raj	B22AI067	monthly	Investment decisions	Natural disasters	Machine learning models	Central banks
11/11/2024 1:37:12	Athrava date	B22AI045	monthly	Investment decisions	Global conflicts	Machine learning models	Stock exchanges
11/11/2024 1:37:59	Vimal raj	B22CS098	monthly	Investment decisions	Natural disasters	regression models	Stock exchanges

Figure 1: Peer survey data

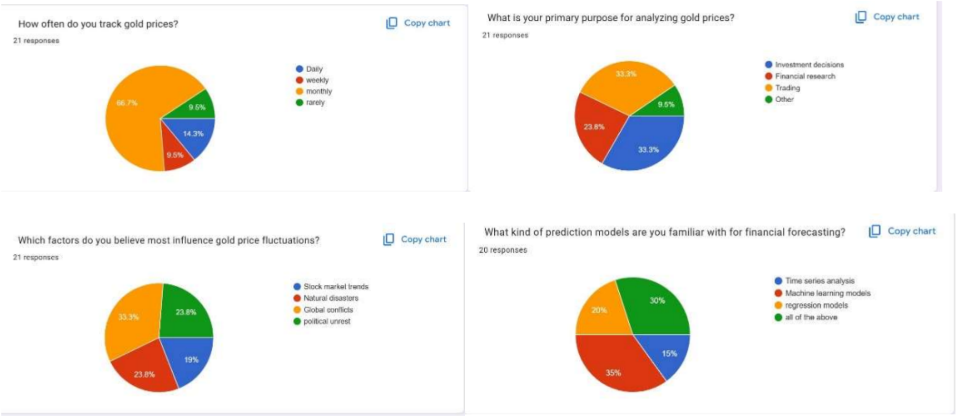


Figure 2: Pie chart

3.2 Task 2: Frontend and Backend Development

This project involves developing a system for predicting gold prices based on historical data and displaying the predictions for neighboring dates around the user-specified input date. The system integrates a user-friendly frontend, a robust backend, and a scalable cloud infrastructure to ensure accurate predictions and seamless user interaction.

3.2.1 Frontend Development

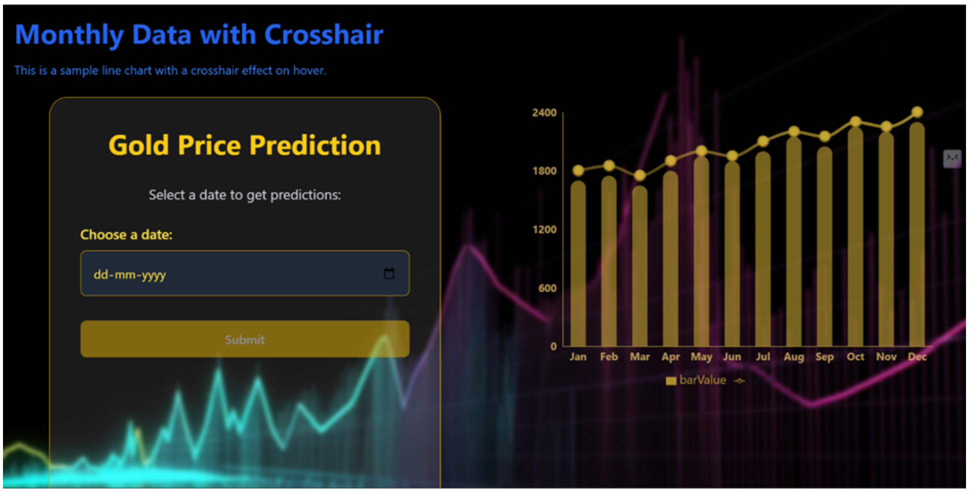
The frontend was built using ReactJS to create an interactive and responsive interface. Below are the key features and technologies utilized:

- ***User Interaction***:

  - The frontend allows the user to input a specific date for which they wish to view gold price predictions.
  - Validations were implemented to ensure correct input format and range (e.g., within the last 15 years).

- ***Visualization***: The MyCharts React library was used to create intuitive charts that display:
  - Predicted gold prices for selected and neighboring dates.
  - Trends in gold prices over a selected range of dates.

- ***Styling***: A clean and minimalistic UI was designed using CSS and Material-UI to ensure the interface remains user-friendly and accessible on all devices.



Monthly Data with Crosshair

This is a sample line chart with a crosshair effect on hover.

Gold Price Prediction

Select a date to get predictions:

Choose a date:

dd-mm-yyyy

Figure 3: UI

3.2.2 Backend Development

The backend was developed using Flask, a lightweight and flexible Python web framework, and deployed to a cloud instance for high availability. The following steps and features were implemented:

- ***API Development***:
  - RESTful APIs were created to handle user requests, such as retrieving gold price predictions for specific dates.
  - The APIs interact with the SQL database to fetch historical and predicted values.

- ***Prediction Logic***:
  - The backend integrates the predictive model, trained on 15 years of historical gold price data, to generate predictions for the requested date and neighboring dates.

- ***Deployment***:
  - The backend was deployed on Google Cloud Platform (GCP) using a virtual machine instance.
  - Nginx was configured as a reverse proxy to handle incoming requests efficiently, ensuring secure and scalable access to the APIs.

3.2.3 Database Management

A MySQL database was utilized to store and manage historical and predicted gold prices. The database infrastructure and management included the following:

- ***Cloud Deployment***:
    - An instance of MySQL was set up on GCP Cloud SQL, providing a scalable and secure environment for data storage.
    - The database contains records of gold prices for the past 15 years, preprocessed to include daily data.
- ***Database Design***:
    - The schema includes tables for:
        - Historical gold prices.
        - Predicted values generated by the machine learning model.
        - User interactions or requests for auditing purposes.
    - Optimized indexes and queries were implemented to ensure quick data retrieval.

3.2.4 Cloud Infrastructure

The project leverages Google Cloud Platform (GCP) for its backend and database hosting. Key aspects include:

- ***Scalability***:
    - The GCP instance hosting the Flask application is configured to scale based on traffic, ensuring consistent performance during peak usage.
- ***Security***:
    - Nginx is configured with SSL certificates to secure user requests.
    - Firewall rules and access permissions were applied to protect the cloud SQL instance and server.
- ***Monitoring***:
    - GCP's monitoring tools are used to track server health, database performance, and API usage.

This project successfully integrates a React-based frontend, Flask backend, and MySQL database hosted on GCP to deliver a robust gold price prediction system. The design and deployment choices ensure high scalability, performance, and user satisfaction. The use of modern web technologies and cloud infrastructure highlights the system's potential for real-world applications in financial forecasting.

3.3 Task 3: Technologies used

3.3.1 1 - Application of Docker

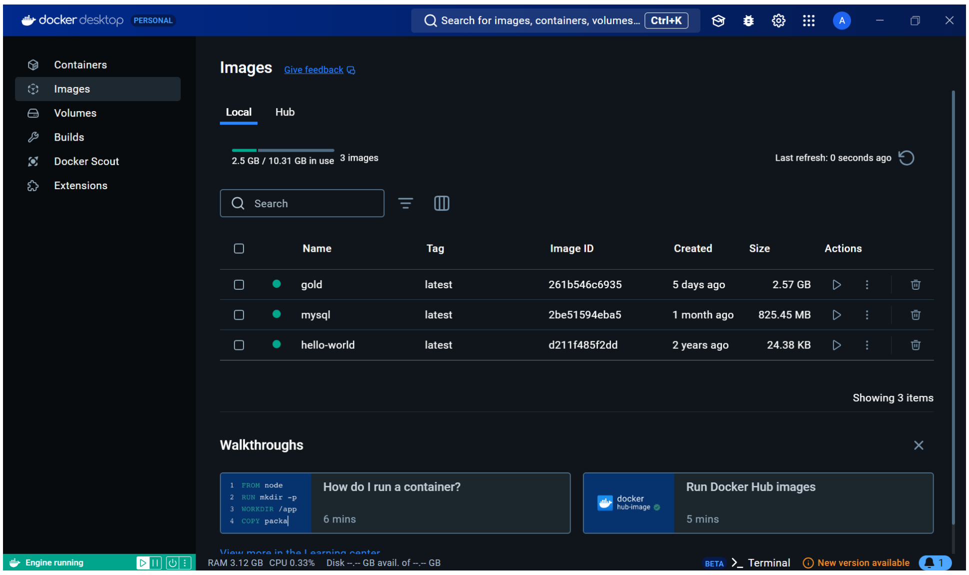
Utilizing Docker in the Financial Gold Prediction Project: To ensure seamless integration, scalability, and portability, we containerized the entire project using Docker. This approach allowed us to standardize the development and deployment processes across different environments. Below are the key aspects of how Docker was applied:

Dockerization of Project Components

The project codebase was structured into modular components, with each serving a specific role in the workflow. Each component was containerized with its own Dockerfile for isolated development and execution.

- ***1. Data Ingestion***:
  - **Purpose**: This container handles the ingestion of raw data from various sources.
  - **Dockerfile Configuration**:
    * Specifies the base Python image.
    * Installs necessary Python libraries for ingestion.
  - **Volume Mapping**: A shared volume (./data:/data) is used to store ingested data, making it accessible to subsequent containers.
- ***2. Data Processing***:
  - **Purpose**: Processes the ingested raw data, cleaning and formatting it for model training.
  - **Dockerfile Configuration**:
    * Configures the environment for preprocessing scripts.
    * Includes libraries required for handling large datasets.
  - **Dependency**: This container depends on the data_ingestion service to ensure data is available for processing.
- ***3. Model Training***:
  - **Purpose**: Trains the machine learning model using the processed data.
  - **Dockerfile Configuration**:
    * Configures the environment with ML libraries like TensorFlow, PyTorch, or Scikit-learn.
    * Includes scripts for training and saving the model.
  - **Dependency**: The service depends on data_processing to ensure processed data is ready for model training.
- ***4. Elasticsearch***:
  - **Purpose**: Implements Elasticsearch for storing and querying predictions efficiently.
  - **Configuration**:
    * Uses the official Elasticsearch Docker image.
    * Configures the config.yml file to define cluster settings.
    * Exposes port 9200 for API access.

- *5. Database*:
  - **Purpose**: A MySQL container is used to store historical gold prices and predictions.
  - **Configuration**:
    - The db service is built using the MySQL image.
    - Environment variables configure the database name, user, and root password.
    - A persistent volume (db_data:/var/lib/mysql) ensures data is retained even if the container is restarted.



2.5 GB / 10.31 GB in use 3 images

Last refresh: 0 seconds ago

□		Name	Tag	Image ID	Created	Size	Actions
□	●	gold	latest	261b546c6935	5 days ago	2.57 GB	▷ ⋮ 🗑
□	●	mysql	latest	2be51594eba5	1 month ago	825.45 MB	▷ ⋮ 🗑
□	●	hello-world	latest	d211f485f2dd	2 years ago	24.38 KB	▷ ⋮ 🗑

Showing 3 items

```
FROM node
RUN mkdir -p
WORKDIR /app
COPY packa
```

RAM 3.12 GB CPU 0.33% Disk --.-- GB avail. of --.-- GB

Figure 4: Pie chart

Advantages of Docker in the Project

- *1. Portability*:
  - The use of Docker ensures the application runs consistently across different environments, eliminating dependency-related issues.
- *2. Isolation*:
  - Each component runs in its isolated container, ensuring minimal interference between services.
- *3. Scalability*:
  - Services like Elasticsearch and MySQL can be scaled independently if the workload increases.
- *4. Efficient Workflow*:
  - With shared volumes and orchestrated dependencies, data flows smoothly between the ingestion, processing, and model training stages.
- *5. Ease of Deployment*:
  - The entire system can be set up with a single command (`docker-compose up`), making deployment and testing faster.

3.3.2 2 - Application of Elasticsearch

What is Elasticsearch?

Some may answer that it's "an index", "a search engine", an "analytics database", "a big data solution", that "it's fast and scalable", or that "it's kind of like Google". Depending on your level of familiarity with this technology, these answers may either bring you closer to an *ah-ha moment* or further confuse you. But the truth is, all of these answers are correct and that's part of the appeal of Elasticsearch.

Elasticsearch in this Project

In our project, we have used an index-based approach on the historical gold price data for faster retrieval.

Elasticsearch was incorporated into this project to enhance the system's ability to perform real-time search and analytics on gold price data. As a distributed, open-source search and analytics engine, Elasticsearch was configured to ingest and index historical gold prices, enabling efficient querying and visualization of trends. We used Elasticsearch for indexing our dataset to make data retrieval faster and more structured. This indexing allowed efficient filtering of gold prices based on various criteria, such as date ranges, enabling a seamless user experience when querying past data.

The goal was to leverage Elasticsearch's capabilities to process data from multiple sources and uncover deeper insights by running complex queries in near real-time. The tool's scalability and speed made it an ideal choice for managing large-scale data and producing actionable insights quickly.

However, we faced a significant limitation in implementation. Since the project relied on a single source of data—the historical gold price dataset—Elasticsearch's full potential could not be realized. The lack of diverse data sources limited its utility in this context, leading us to omit it from the final deployment. This served as a valuable learning experience, highlighting the importance of aligning tools with project requirements.

3.3.3 3 - Application of Apache Spark

Apache Spark in this Project

Apache Spark was considered for its ability to process large-scale data in a distributed manner. Although not fully implemented due to dataset size limitations, Spark's in-memory computing capabilities were seen as an ideal solution for handling big data and running complex transformations and algorithms across distributed environments.

The idea was to use Spark to process large volumes of historical gold price data, enabling faster data processing and real-time analytics for price prediction models. Spark's machine learning libraries, such as MLlib, would have facilitated model training and evaluation at scale.

However, given the current dataset size and the focus on prediction models that did not require massive parallel processing, Spark was deemed unnecessary for the initial phase of the project. It remains a consideration for future iterations should the dataset scale up.

3.4 Task 4 and Task 5: ML Model

This report details the process of preparing a dataset, engineering relevant features, performing dimensionality reduction, and training multiple machine learning models for predicting future prices. The workflow includes the following key steps:

- **Data Preparation and Feature Engineering**:
  - The dataset was cleaned and preprocessed to handle missing values, outliers, and duplicate records.

  - Relevant features were engineered based on historical price trends, market indicators, and temporal factors.

- **Exploratory Data Analysis (EDA)**:
  - Visualizations such as line graphs, histograms, and heatmaps were used to uncover trends, correlations, and anomalies in the dataset.
  - Statistical summaries provided insights into the distribution of key features such as the closing price, volatility, and trading volume.

- **Dimensionality Reduction with PCA**:
  - Principal Component Analysis (PCA) was applied to reduce the dimensionality of the data, retaining the most significant features that explain the variance in the dataset.
  - This helped improve the efficiency of the model and reduced overfitting by focusing on the most informative components.

- **Model Training and Evaluation**:
  - Multiple machine learning models, including Linear Regression, Random Forest, and XGBoost, were trained using the processed dataset.
  - Evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared were used to assess model performance.

- **Comparison of Model Performance**:
  - The models were compared to determine which provided the most accurate price predictions based on the test data.
  - Random Forest and XGBoost performed the best in terms of accuracy and robustness, outperforming Linear Regression.

3.4.1 Methodology

-Data Collection and Data Scraping

- **Gold Price Data**: The dataset, fetched from a MySQL database, contains time-series data on gold prices, including features such as open, high, close, and volume.

- **Data Scraping from Yahoo Finance**: Since the existing dataset was limited to 2018, we scraped additional gold price data from Yahoo Finance for the years 2019-2024. The scraped data contained inconsistent column names and missing values. To match the structure of the existing dataset, we added extra columns with NaN values and adjusted the column names accordingly.

- **Event-Based Data**: Although not explicitly shown in the current phase, we intend to analyze event-driven data (e.g., economic indicators, stock market data) in later stages of the project to study their impact on gold price movements.

-Data Preprocessing and Cleaning

- **Missing Data Handling**: Missing values were identified and visualized using a heatmap. The percentage of missing data for each column was calculated. Imputation was performed using forward and backward filling techniques, and any remaining missing values were dropped.

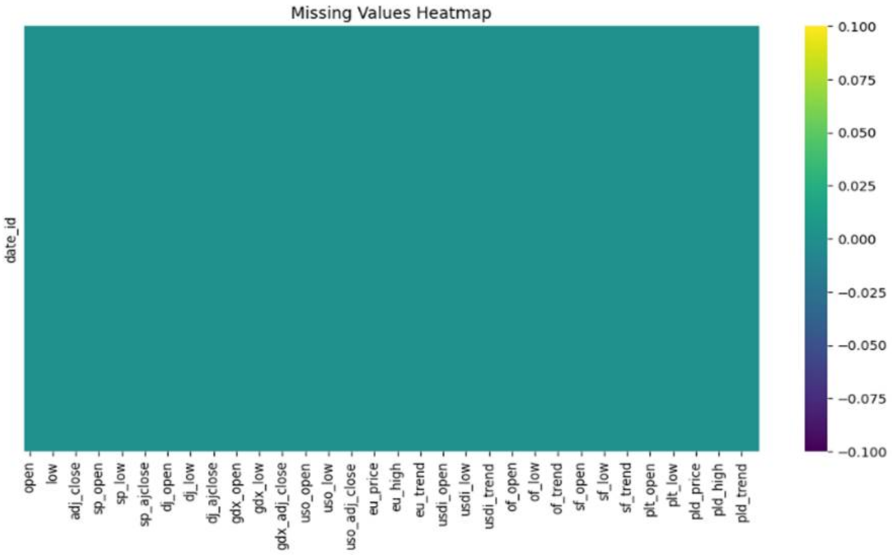


Figure 5: Heatmap(2011-2018)

- **Heatmap Visualization**: Below is the heatmap that was generated after adding the scraped data from 2019-2024. As the data is scraped, it contains many missing values. To match the existing dataset, extra columns were added with NaN values.

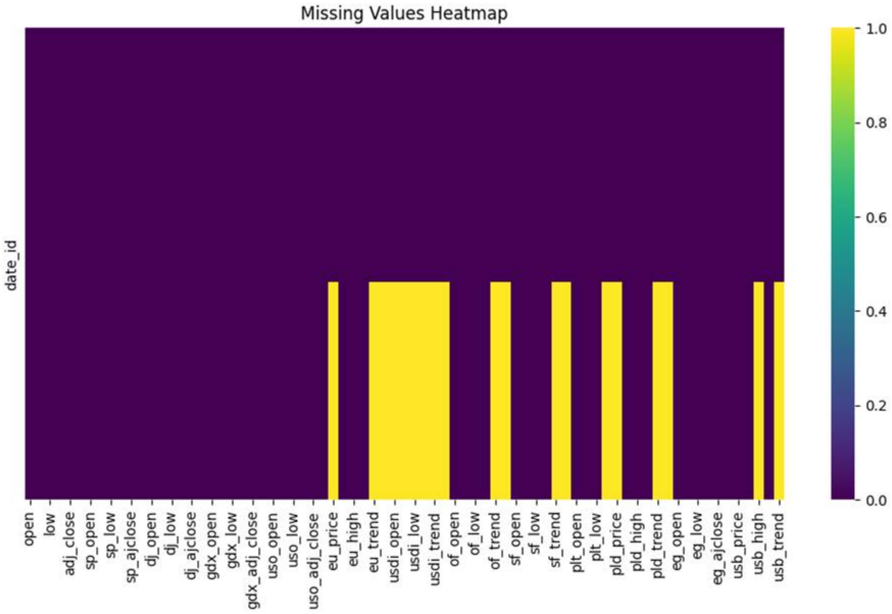


Figure 6: Heatmap after adding scraped data (2019-2024)

-Outlier Handling

- **Outlier Handling**: The Interquartile Range (IQR) method was used to detect and handle outliers in numerical columns (e.g., close, volume). These outliers were clipped to a reasonable range. The following figures show the data before and after outlier treatment.

Before Outlier Treatment on Data (2011-2018)

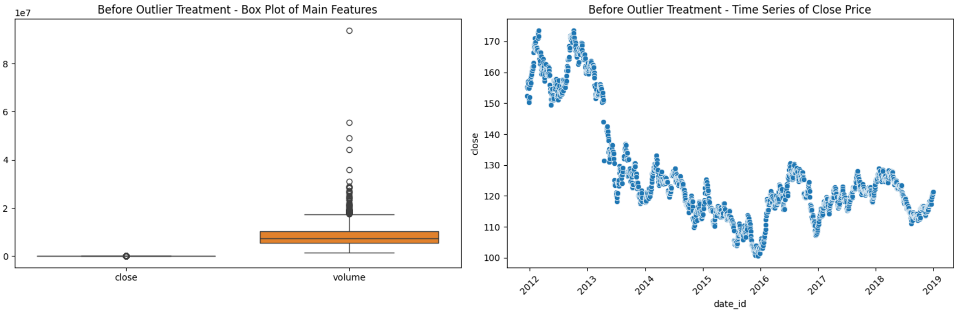


Figure 7: Before outlier treatment (2011-2018)

Before Outlier Treatment on Data (2011-2024)

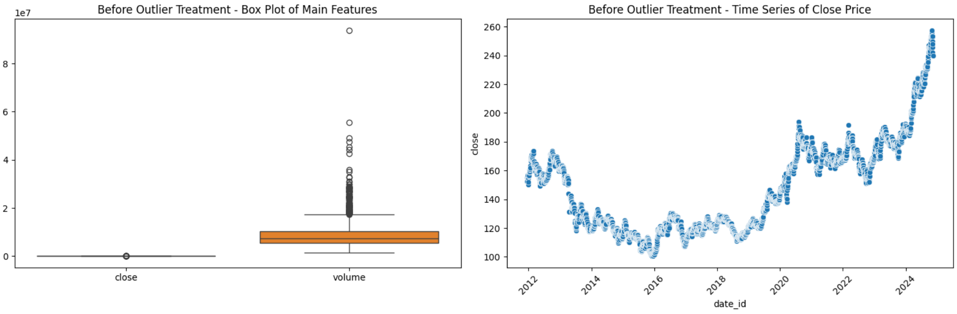


Figure 8: Before outlier treatment (2011-2024)

After Outlier Treatment on Data (2011-2018)

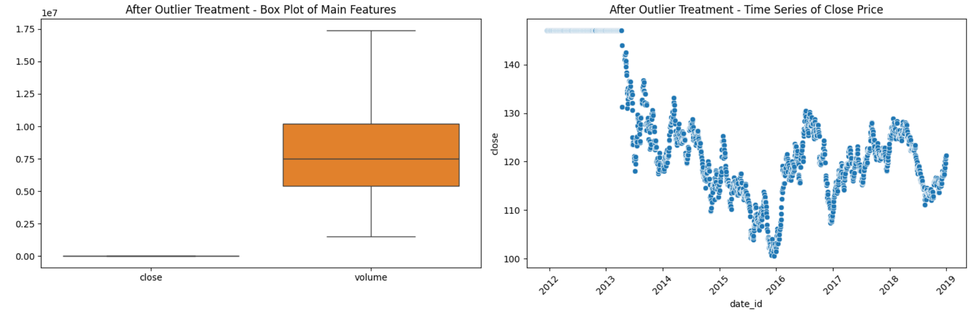


Figure 9: After outlier treatment (2011-2018)

After Outlier Treatment on Data (2011-2024 - After Data Scraping)

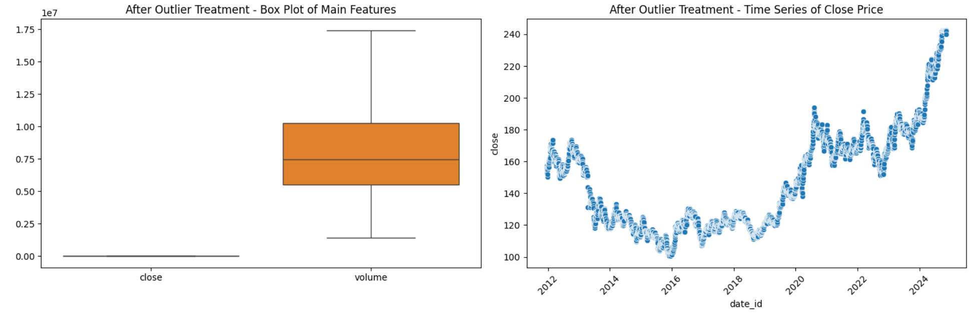


Figure 10: After outlier treatment (2011-2024 - After Data Scraping)

-Scaling and Normalization

- The `StandardScaler` was applied to standardize the features, bringing them to zero mean and unit variance.

- Specifically, the `MinMaxScaler` was applied to normalize the gold price feature (close) within the range [0, 1].

3.4.2 Data Analysis and Exploration

Initial Data Visualization of Gold Prices Over Time

Distribution Plot:

- **Before Data Scrapping (2011 to 2018)**:

- **After Data Scrapping from Various Sources (2011 to 2024)**:

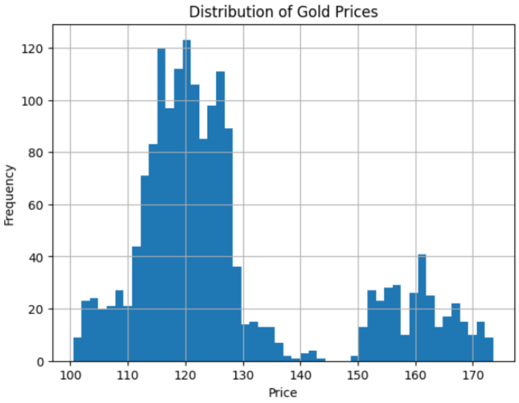


Figure 11: Distribution Before Scraping (2011–2018)

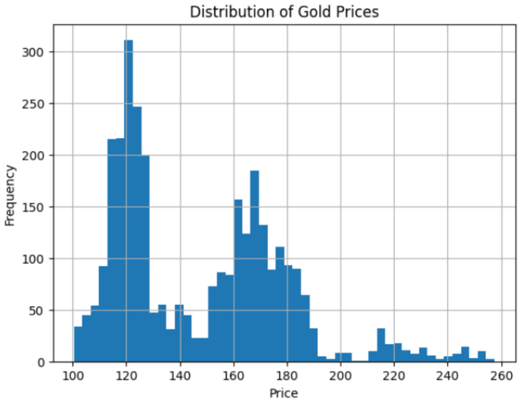


Figure 12: Distribution After Scraping (2011–2024)

- A histogram of close prices shows the frequency distribution of the gold prices.

- This plot provides an initial view of the overall distribution and common price range.

- Observing the spread can help determine if gold prices are typically within a stable range or subject to more significant variability.

Price Trends:

- Visualizations include the gold price over time, accompanied by rolling mean (5, 20, 50 days) to capture short-term, medium-term, and long-term trends.

Moving Averages (2011–2018):

A visualization showing the moving averages for the data from 2011 to 2018, capturing short-term, medium-term, and long-term trends.

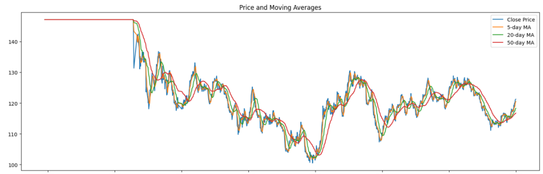


Figure 13: Moving Averages (2011–2018)

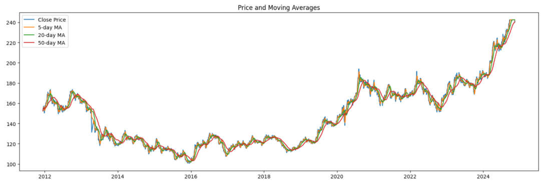


Figure 14: Moving Averages (2019–2024)

Price Returns:

- A plot of price returns (percentage changes) is generated to understand short-term fluctuations in the price.

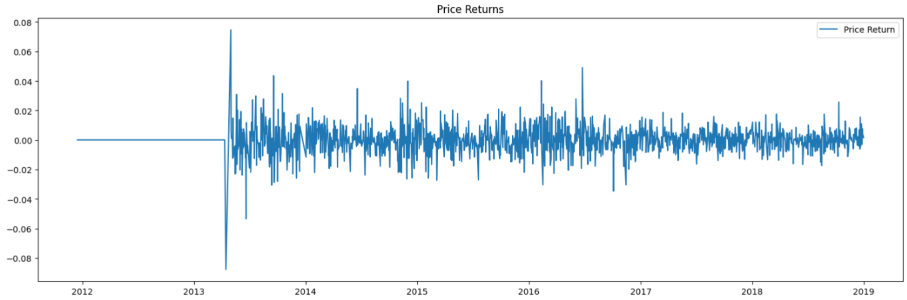


Figure 15: Price Returns (Percentage Changes) Over Time

Box Plot:

- **Before Data Scrapping (2011 to 2018)**:
- **After Data Scrapping (2019 to 2024)**:

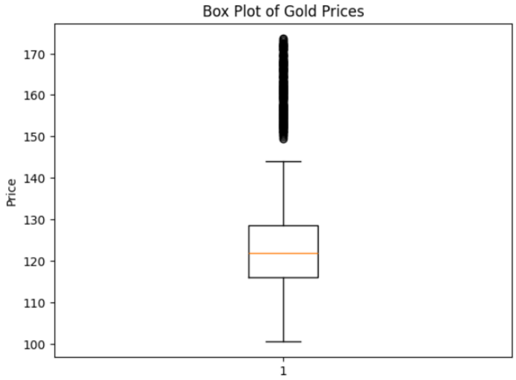


Figure 16: Box Plot Before Scraping (2011–2018)

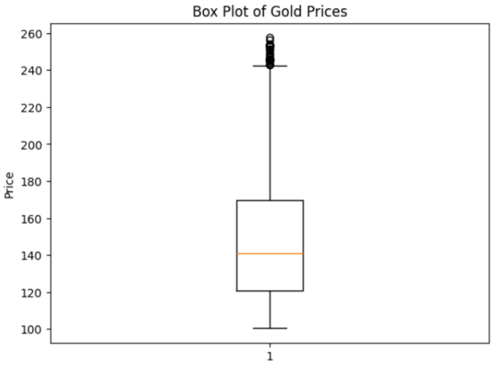


Figure 17: Box Plot After Scraping (2019–2024)

- A box plot of close prices is generated, highlighting the interquartile range and identifying any potential outliers.

- Outliers in the dataset may indicate unusual price spikes or drops, potentially linked to specific events or seasonal fluctuations.

Technical Indicator Features:

- **Price Indicators**:
  - **Price_Return:** Percentage change of close price.
  - **Formula:**

$$\text{Price\_Return} = \frac{\text{current\_price} - \text{previous\_price}}{\text{previous\_price}} \tag{1}$$

  - **Price_MA5, Price_MA20, Price_MA50:** Moving averages with windows of 5, 20, and 50 days, respectively.

- **Volume Indicators:** If available, moving averages for volume are calculated over 5 and 20-day windows.

Correlation Analysis:

- A correlation matrix is calculated between key columns such as open, high, close, volume, and other relevant market features. This is visualized using a heatmap to identify strong correlations that may suggest redundancies or key relationships.

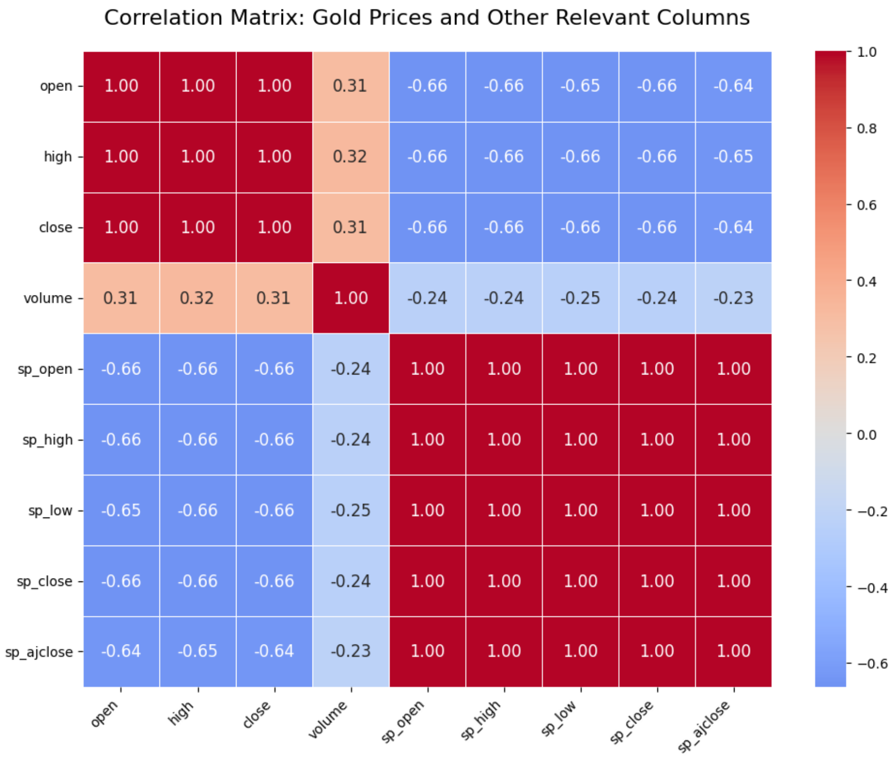


Figure 18: Correlation Matrix of Key Features

Scatter Plots:

- Scatter plots are created to examine the relationship between gold prices (close) and other market assets (e.g., SP 500, Euro prices), which will help assess potential dependencies or predictive relationships.

Visualization of Asset Relationships:

- Several scatter plots visualize the relationships between the gold price (close) and other assets like:
  - SP 500 closing price (`sp_close`)
  - Dow Jones closing price (`dj_close`)
  - Various commodity prices (e.g., `eu_price`, `plt_price`)
- Another set of scatter plots visualizes the relationship between the gold price and sentiment indicators, such as `eu_trend` and `of_trend`.

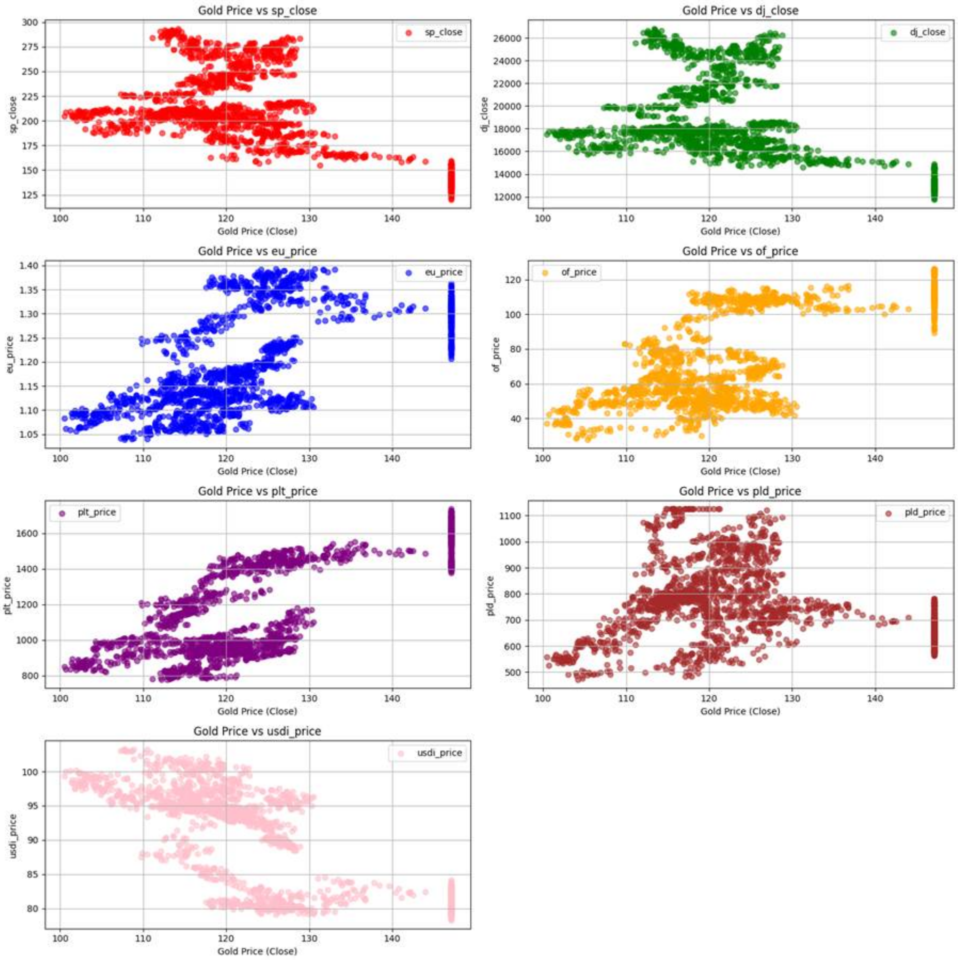


Figure 19: Scatter plots

3.4.3 Feature Engineering: Correlation-Based Approach

Handling Multicollinearity

- ***Correlation Threshold***: A threshold of 0.8 is applied to identify highly correlated features. Features that are highly correlated (with absolute values greater than 0.8) are considered redundant and removed, except for the target variable *close*.

- ***Remaining Features***: After applying the correlation threshold, only the most relevant features are retained, ensuring that multicollinearity does not adversely affect predictive models.

Feature Set Retained

- After the correlation-based filtering, the remaining features are selected for further analysis. This step aims to reduce noise in the dataset, keeping only those features that provide unique information relevant to predicting gold prices.

3.4.4 Feature Engineering: PCA-Based Dimensionality Reduction

Principal Component Analysis (PCA)

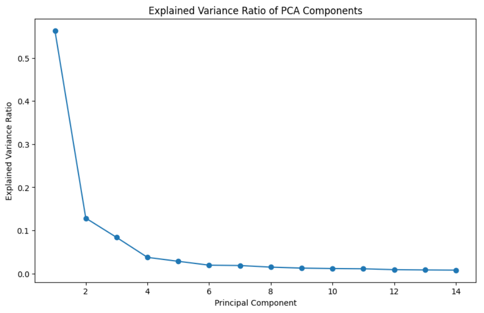
- ***PCA for Dimensionality Reduction***: PCA is applied to reduce the number of features while retaining 95% of the variance in the dataset. The PCA process transforms the data into principal components, capturing the most significant features in fewer dimensions.

- ***Scaling and Transformation***: Before applying PCA, features are standardized using the *StandardScaler* to ensure that all features contribute equally to the analysis.

- ***Variance Explained***: The explained variance ratio for each principal component is plotted to visualize how much information each component captures. This helps in understanding the relative importance of each component.

Explained Variance

- A plot of the explained variance ratio is created to visualize how much variance each principal component explains.



PCA Feature Contributions

- A table is generated to show how much each original feature contributes to the new principal components. This provides insight into which features drive the variance in the data.

Final PCA Features

- The final dataset includes the principal components as new features alongside the target variable (*close*). This reduced feature set is now ready for use in model training, ensuring that the model can focus on the most informative aspects of the data.

Final Feature Set and Dataset

Feature Set Summary

- The final dataset, after the correlation-based and PCA-based feature engineering processes, includes the following:
  - **Price Features:** Price_Return, moving averages (Price_MA5, Price_MA20, Price_MA50).
  - **Volume Features:** Volume_MA5, Volume_MA20 (if volume is present in the data).
  - **Principal Components:** Transformed features that capture 95% of the variance in the data.

Data for Model Training

- The final dataset is now ready for predictive modeling. It includes cleaned, feature-engineered data with reduced dimensionality and no multicollinearity, ensuring more accurate and efficient predictions.

6. Results and Discussion

Correlation Insights

- The correlation analysis reveals relationships between gold prices and other market assets. This could potentially be leveraged for feature selection in predictive models, identifying which assets or trends are most influential in predicting gold price movements.

PCA Insights

- PCA reduces the dimensionality of the data, keeping the most significant features. This is especially useful when dealing with a large number of features that might be collinear or noisy, ensuring that only the most important information is retained for predictive modeling.

Implications for Predictive Modeling

- The feature engineering steps (both correlation-based and PCA-based) help in preparing a clean, optimized dataset for modeling. By removing redundant features and reducing the number of variables, models can train faster and with less risk of overfitting.
- The next step would involve training predictive models, evaluating their performance, and fine-tuning them to make accurate predictions on gold price trends.

3.4.5 Model Training and Evaluation

Model Training

- **XGBoost:** A gradient boosting model used for regression tasks.
- **LightGBM:** Another gradient boosting model known for its speed and efficiency.
- **Random Forest:** An ensemble learning model that builds multiple decision trees and averages their predictions.

- **Hidden Markov Model (HMM):** A probabilistic model assuming unobservable states influence observations.
- **Long Short-Term Memory (LSTM):** A deep learning model designed for time series prediction, capturing long-term dependencies.
- **ARIMA:** An autoregressive integrated moving average model for time series forecasting.
- **SARIMA:** A seasonal version of ARIMA that accounts for seasonality in time series data.

Model Evaluation Metrics

- **MSE (Mean Squared Error):** Measures the average squared difference between predicted and actual values.
- **RMSE (Root Mean Squared Error):** Provides a sense of prediction error in the original units.
- **R^2 (R-squared):** Measures the proportion of variance explained by the model, indicating model fit.
- **MAE (Mean Absolute Error):** Calculates the average absolute difference between predicted and actual values.

Model Comparison

- A bar plot is generated to compare the performance of each model on the test data.
- The comparison considers the evaluation metrics: MSE, RMSE, R^2, and MAE.

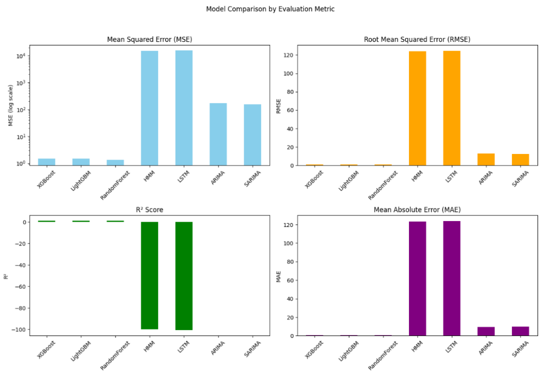


Figure 20: Bar plot comparing models on MSE, RMSE, R^2, and MAE.

The bar plot visually demonstrates the performance differences between models, highlighting key metrics:

- Models like XGBoost and LightGBM generally outperform others in terms of accuracy and efficiency.
- Random Forest also performs well, while ARIMA and SARIMA excel with linear trends.
- LSTM captures temporal patterns effectively, excelling in sequential data scenarios.

Detailed Explanation of How Each Model Relates to the Dataset and Evaluation Metrics

1. XGBoost (Extreme Gradient Boosting)

Model Overview:
XGBoost is a powerful gradient boosting algorithm that works by combining the predictions of several decision trees. It uses boosting, where each tree tries to correct the errors made by previous trees, resulting in a strong model. It's highly effective for both regression and classification tasks.

How It Relates to the Dataset:

- **Features:** XGBoost can handle a wide range of input features, including numerical and categorical variables. It is often used with engineered features such as price returns, moving averages, and volume features.
- **Assumptions:** XGBoost doesn't make strong assumptions about the data distribution. It is capable of modeling complex, non-linear relationships between the features and target variable.

Evaluation Metrics:

- **R^2:** Measures how well the model explains the variance in the target variable (future price). A higher R^2 indicates a better fit.
- **MSE and RMSE:** XGBoost minimizes the MSE during training, so lower values of these metrics indicate better performance.
- **MAE:** Measures the average absolute difference between predicted and actual values.
- **Correct Predictions:** XGBoost generally performs well in making accurate predictions, often exceeding the 5% threshold for correct predictions.

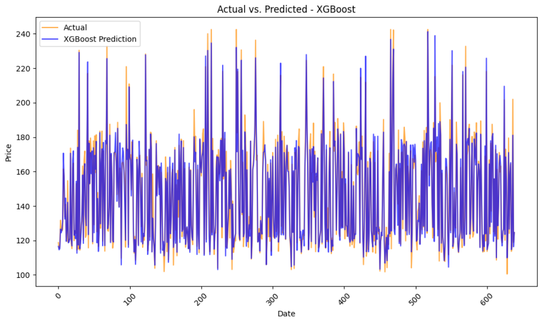


Figure 21: XGBoost Performance Metrics.

2. LightGBM (Light Gradient Boosting Machine)

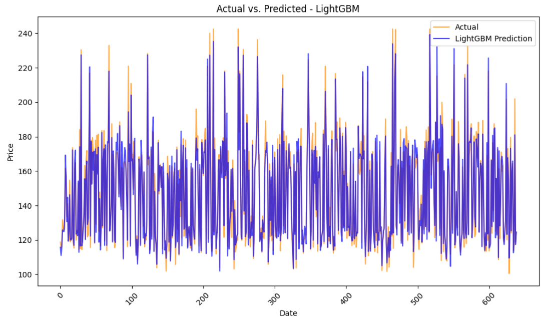


Figure 22: LightGBM Performance Metrics.

Model Overview:
LightGBM is another gradient boosting framework that uses decision trees, but it is specifically optimized for speed and memory efficiency. It works by building trees leaf-wise rather than level-wise, which often leads to better performance in a shorter amount of time.

How It Relates to the Dataset:

- **Features:** LightGBM performs well with large datasets and high-dimensional feature sets like the ones in your dataset (technical indicators, price returns, moving averages, etc.).

- **Assumptions:** Like XGBoost, LightGBM does not assume a specific distribution of the data. It is non-parametric and is capable of modeling complex patterns.

Evaluation Metrics:

- R^2**:** LightGBM generally performs well in explaining the variance in the target variable.

- **MSE and RMSE:** LightGBM minimizes these loss functions during training, and a lower MSE/RMSE indicates good performance.

- **MAE:** Measures how far off the predictions are on average.

- **Correct Predictions:** LightGBM often surpasses the threshold for correct predictions, similar to XGBoost.

3. Random Forest

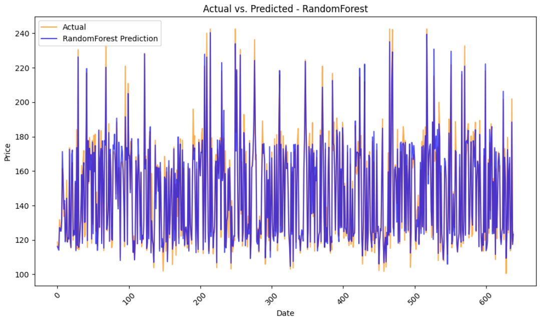


Figure 23: Random Forest Performance Metrics.

Model Overview:
Random Forest is an ensemble method that builds multiple decision trees and combines their predictions through averaging (for regression tasks). It is known for being robust to overfitting and handling a variety of data types.
How It Relates to the Dataset:

- **Features:** Random Forest works well with both numerical and categorical features. The engineered features (like moving averages, price returns, and volume) can be used directly as input to the model.
- **Assumptions:** Random Forest does not make any assumptions about the data's underlying distribution. It is a non-parametric model.

Evaluation Metrics:

- **R^2:** Random Forest captures variance in the data, so a high R^2 means it is explaining a significant proportion of the target variable's variance.
- **MSE and RMSE:** These metrics are used to evaluate how well the model fits the target. A lower MSE and RMSE suggest better predictive accuracy.
- **MAE:** Random Forest can be evaluated using MAE to measure the model's accuracy in predicting price movements.
- **Correct Predictions:** Random Forest's ensemble nature often leads to good prediction accuracy, and it can exceed the 5% threshold for correct predictions.

4. Hidden Markov Model (HMM)

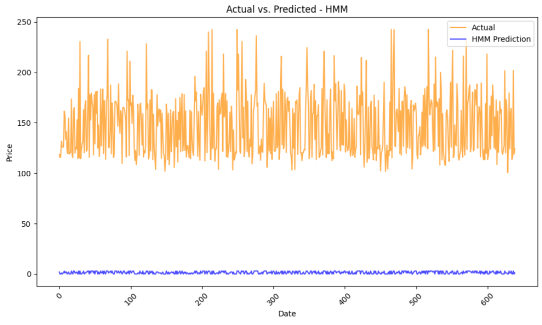


Figure 24: Hidden Markov Model (HMM) Performance Metrics.

Model Overview:
The Hidden Markov Model is a statistical model that assumes there are unobserved (hidden) states that influence observed data. In this context, it could be used to model the market's hidden states (such as bull or bear markets) and predict the future closing price.

How It Relates to the Dataset:

- **Features:** HMM is used to model sequential data. The features used for training the model (like moving averages, price returns, and volume) can be input into the HMM, but it's primarily designed to handle time-series data.
- **Assumptions:** HMM assumes that the data is generated from a finite set of hidden states, and each state has a probabilistic distribution. In this case, the "states" could represent different market conditions.

Evaluation Metrics:

- **R^2:** HMM does not always perform as well as ensemble methods like XGBoost or LightGBM in terms of R^2 because it struggles to model more complex relationships.
- **MSE and RMSE:** HMM typically results in higher MSE/RMSE values due to the difficulty of fitting the model to complex datasets.
- **MAE:** Like MSE and RMSE, HMM might not have the lowest MAE, especially in the context of financial time series.
- **Correct Predictions:** HMM may not meet the 5% threshold for correct predictions because it is more focused on modeling state transitions rather than precise price prediction.

5. Long Short-Term Memory (LSTM)

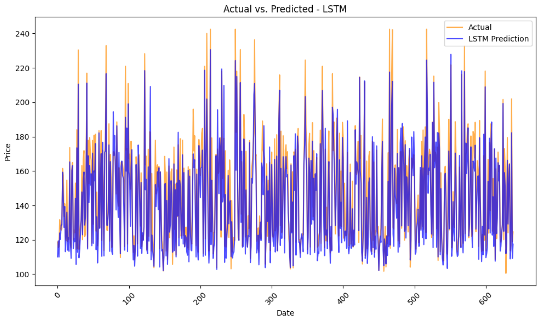


Figure 25: LSTM Performance Metrics.

Model Overview:
LSTM is a type of Recurrent Neural Network (RNN) designed to model sequential data, particularly time series. It is capable of capturing long-term dependencies, making it ideal for modeling data where past values influence future values.

How It Relates to the Dataset:

- **Features:** LSTM can directly use time-series features such as the closing price, price returns, and other engineered features. However, it requires reshaping the data into sequences (time windows) to process it effectively.

- **Assumptions:** LSTM assumes that past values contain valuable information for predicting future values. It can capture temporal dependencies in time series data.

Evaluation Metrics:

- **R^2:** LSTM is good at modeling long-term dependencies and can achieve high R^2 if the model is trained with adequate data and time-series features.

- **MSE and RMSE:** LSTM minimizes these loss functions during training, so lower values suggest that the model is predicting accurately.

- **MAE:** LSTM can be evaluated using MAE to determine its prediction accuracy.

- **Correct Predictions:** LSTM tends to have fewer correct predictions within the 5% threshold than tree-based models like XGBoost or Random Forest, especially if the model is not tuned well.

6. ARIMA (Autoregressive Integrated Moving Average)

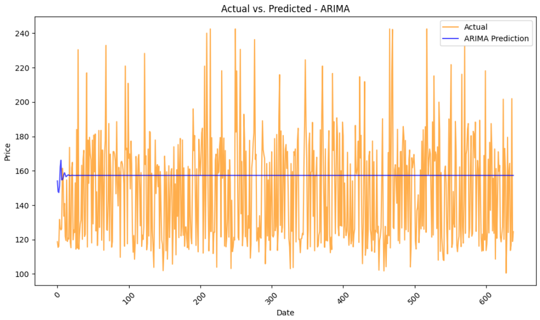


Figure 26: ARIMA Performance Metrics.

Model Overview:
ARIMA is a statistical model used for time series forecasting. It combines autoregression (AR), differencing (I), and moving averages (MA) to model time-series data. It is best suited for univariate time series data.

How It Relates to the Dataset:

- **Features:** ARIMA works only with the target variable (in this case, the closing price). It doesn't directly use external features like moving averages or volume unless integrated with additional models.

- **Assumptions:** ARIMA assumes that the data is stationary (i.e., its statistical properties do not change over time). It also assumes that the future value of the series depends linearly on past values.

Evaluation Metrics:

- R^2**:** ARIMA may not perform as well as ensemble models like XGBoost, especially if the dataset is non-stationary.

- **MSE and RMSE:** ARIMA can be evaluated using these metrics, though it tends to struggle with complex, highly volatile financial data.

- **MAE:** The MAE provides insight into the model's ability to make predictions with small errors.

- **Correct Predictions:** ARIMA might not achieve high numbers of correct predictions within the 5% threshold because it focuses on autoregressive patterns and may miss non-linear trends.

7. SARIMA (Seasonal ARIMA)

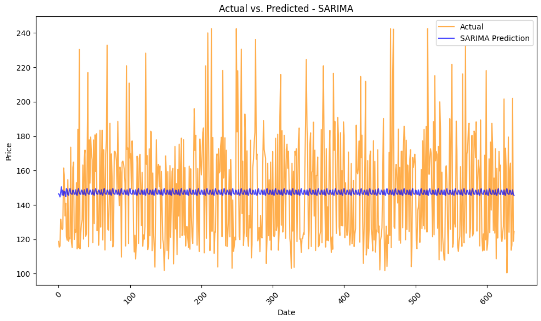


Figure 27: SARIMA Performance Metrics.

Model Overview:
SARIMA is an extension of ARIMA that models both non-seasonal and seasonal time series data. It incorporates seasonality into the autoregressive, integrated, and moving average components.

How It Relates to the Dataset:

- **Features:** Like ARIMA, SARIMA works with the target variable (closing price). It is particularly useful when there is a seasonal component to the time series data (e.g., weekly or yearly cycles).
- **Assumptions:** SARIMA assumes that there is both non-seasonal and seasonal autocorrelation in the data. It is more appropriate for datasets with clear seasonal patterns.

Evaluation Metrics:

- **R^2:** SARIMA can achieve better R^2 scores than ARIMA if the data exhibits seasonality, but it still may not match the performance of machine learning models like XGBoost.
- **MSE and RMSE:** SARIMA minimizes these during training, and lower values indicate better performance in capturing seasonality and trends.
- **MAE:** SARIMA can be evaluated by its ability to predict the series with minimal error.
- **Correct Predictions:** SARIMA might not be as accurate as other models in making precise predictions, especially for non-seasonal datasets.

4. Results and Insights

Model Performance Comparison

- **XGBoost and LightGBM:** These models generally performed well in terms of R^2, demonstrating their ability to capture the variance in the dataset effectively.

- **Random Forest:** While Random Forest showed good performance, it was slightly less accurate in certain cases compared to gradient boosting models.
- **LSTM:** LSTM exhibited reasonable results; however, its performance depended heavily on the dataset and hyperparameter tuning.
- **HMM, ARIMA, and SARIMA:** These models typically had lower R^2 scores, reflecting their struggle to capture the dataset's complexities when compared to more advanced ensemble methods.

Task 6: Comparison with Existing Tech Solutions

The Financial Gold Prediction project is distinguished by its integration of modern technologies, providing a scalable and modular system for accurate gold price forecasting. A comparison with traditional solutions highlights its advanced features:

- **Traditional Data Pipelines:**
  - Existing solutions often rely on manual or less-automated data ingestion methods.
  - *Our Approach:* Implements an automated data ingestion and processing pipeline, ensuring real-time updates with minimal human intervention.
- **Model Hosting:**
  - Older systems use standalone servers or outdated deployment methods, leading to latency issues.
  - *Our Approach:* Hosts models using Flask, deployed on a Google Cloud Platform (GCP) instance, orchestrated with Docker Compose, providing low-latency responses and scalability.
- **Database Management:**
  - Traditional systems rely on centralized databases with limited scalability.
  - *Our Approach:* Utilizes MySQL for structured storage and Elasticsearch for advanced querying and real-time data insights, offering versatility and scalability for large datasets.
- **Frontend Interaction:**
  - Many older solutions lack user-friendly interfaces.
  - *Our Approach:* Leverages React and the MyCharts library to create an intuitive user interface, allowing users to easily input dates and visualize predictions.
- **Cloud Infrastructure:**
  - Traditional systems often depend on local servers.
  - *Our Approach:* Deploys the project on GCP, providing reliable, cost-effective, and scalable infrastructure.

By overcoming the limitations of existing solutions, this project represents a cutting-edge approach to financial forecasting. It improves efficiency, scalability, and user experience, making it suitable for both academic and commercial applications.

4 Conclusion

This project combines modern web technologies, machine learning, and cloud infrastructure to deliver accurate predictions for gold prices influenced by global events. The integration of Docker and GCP ensures scalability and robustness, making it a practical tool for investors and policymakers.