**Data mining project using CRISP-DM**



Names: Umed Nuri 1674823 & Basmalah Alhamaideh

Teacher: Witek ten Hove

Class: T06- Data mining

Date: 04-04-2024

Table of Contents

[Business Understanding 2](#_Toc194514591)

[Project Objectives 3](#_Toc194514592)

[Background 3](#_Toc194514593)

[Success Criteria 3](#_Toc194514594)

[Data Understanding 4](#_Toc194514595)

[Data Collection 4](#_Toc194514596)

[Data Description 5](#_Toc194514597)

[Initial exploration 6](#_Toc194514598)

[Data Preparation 7](#_Toc194514599)

[Data Cleaning 7](#_Toc194514600)

[Data Integration 7](#_Toc194514601)

[Feature Engineering 7](#_Toc194514602)

[Modeling 8](#_Toc194514603)

[Model Selection 8](#_Toc194514604)

[Parameter Tuning 9](#_Toc194514605)

[Training & Validation 9](#_Toc194514606)

[Evaluation 10](#_Toc194514607)

[Performance Metrics 10](#_Toc194514608)

[Model Interpretability 10](#_Toc194514609)

[Comparison with Baseline 10](#_Toc194514610)

[Deployment 11](#_Toc194514611)

[Deployment Strategy 11](#_Toc194514612)

[Monitoring 11](#_Toc194514613)

[Future work 12](#_Toc194514614)

[Conclusion 13](#_Toc194514615)

[Reverences 14](#_Toc194514616)

# 

# Business Understanding

In this section of the report, we will outline the key objectives of the project and define the primary business problem it aims to address. We’ll begin by providing background and context to help frame the problem and then move on to discuss the criteria for success, along with the key constraints that must be taken into account.

## Project Objectives

The primary objective of this project is to design and develop a machine learning model capable of accurately predicting the volatility of the gold market, using the Average True Range (ATR) as the key metric. Volatility, in this context, refers to the degree of variation in gold prices over time, which ATR effectively captures.

While the long-term vision for this project includes leveraging the volatility predictions to inform options trading strategies specifically straddle trading the immediate focus is on building a highly accurate and reliable prediction model. This involves data collection, feature engineering, model selection, training and rigorous fine-tuning to ensure optimal performance.

## Background

Volatility in the gold market presents both risk and opportunity for traders and investors. Predicting this volatility measured using Average True Range (ATR) can improve trading strategies, particularly options strategies like straddles, which profit from large price movements regardless of direction.

The core problem is to develop a machine learning model that can accurately forecast future gold market volatility. A high-performing predictive model could be used in the future to inform trading decisions, optimize risk management and improve returns on strategies that depend on price movement rather than direction.

## Success Criteria

The success of this project is based on developing a machine learning model that can accurately predict the future volatility of the gold market, as measured by the Average True Range (ATR). ATR is a volatility indicator measuring the average price movement of an asset over a specified time period (Finserv, 2024).

Key performance indicators include prediction accuracy, with metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R² score used to evaluate the model. These metrics we will explain Furter in this report.

Beyond accuracy, the model must be robust across various market conditions and generalize well to unseen data. It should also produce timely predictions that are suitable for real-world financial applications, such as daily or intraday forecasting. Equally important is the model’s interpretability, allowing stakeholders to understand the key factors driving its predictions. Finally, the model should be scalable and easy to maintain, enabling regular updates and seamless integration into future trading or decision-making systems.

Several constraints must be considered throughout the development process. Data quality and availability are foundational accurate historical gold price data and related features must be sourced, cleaned and aligned appropriately.

Although the long-term goal is to apply the model in trading strategies, such as straddles, the current phase strictly focuses on predicting volatility, excluding any trading logic. Looking ahead, ethical and regulatory considerations may come into play if the model is used in actual financial environments.

# Data Understanding

In this section of the report, we will describe the data sources used and the methods employed to collect the data. We will assess the reliability of these sources and highlight any potential data quality issues. The key attributes in the dataset will be listed and explained, providing a clear understanding of the available features. Finally, we will present a summary of our initial data exploration, supported by descriptive statistics and visualizations.

## Data Collection

For this project, we used 15-minute interval gold futures data, identified by the ticker symbol “GC=F,” which we retrieved from Yahoo Finance. The dataset spans the most recent 60-day period and includes standard market indicators such as open, high, low, close (OHLC) prices and trading volume. The data was downloaded in CSV format and imported into our environment for pre-processing and analysis.

Yahoo Finance is a widely used platform for accessing financial market data and is generally considered a reliable source for historical pricing information, particularly for research and modelling purposes. While it may not match the precision of premium platforms like Bloomberg or Refinitiv, it is suitable for academic and exploratory projects such as this one.

Upon importing the CSV file, we encountered a formatting issue: the file contained two header rows, forming a multi-index column structure. This required a pre-processing step to clean the dataset by removing the extra header row and simplifying the column names for ease of use in analysis and modelling. Additionally, as with most intraday data, occasional gaps in timestamps or entries with zero trading volume were present, especially during non-trading hours or weekends. These issues were addressed during the data cleaning phase to ensure a consistent and usable dataset.

## Data Description

**Below we will explain and list the attributes present in the dataset we used:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Purpose** |
| **Datetime** | Timestamp for the 15-minute interval | Track time sequence and enable time-based analysis |
| **Close** | Last traded price during the interval | Used in calculating indicators like returns and ATR |
| **High** | Highest price during the interval | Input for volatility and ATR calculations |
| **Low** | Lowest price during the interval | Used in price range and volatility analysis |
| **Open** | First traded price during the interval | Helps in identifying price trends and patterns |
| **Volume** | Total trading volume during the interval | Measures trading activity and confirms price moves |

## Candlestick - Definition, Explained, Patterns, Chart, TradingInitial exploration

Figure 1 Candles

This image shows how to read candlestick charts used in trading.

The green candle means the price went up during that time.

It opened low and closed high.

The red candle means the price went down.

It opened high and closed low.

Each candle has:

Body: The range between open and close.

Upper Shadow: The highest price reached.

Lower Shadow: The lowest price reached within that time period.

We wanted to see this information in the csv file. And we got exactly the data we needed as seen in figure 2.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 2 First rows of the CSV

The csv shows 4558 rows of data. This is roughly 60 days of data as we stated at the start. Keep in mind that the markets are open only on weekdays, so calculated backwards the csv has around 47 days of data.

# Data Preparation

In this section of the report, we detail the data cleaning and transformation steps applied to prepare the dataset for analysis. We explain how data from multiple sources were integrated, ensuring consistency and alignment. Finally, we describe any new features that were engineered from the original data to enhance the predictive power of the model.

## Data Cleaning

Only a few cleaning and transformation steps were applied to the dataset. First, the CSV file imported from the data source contained two header rows, creating a multi-index structure. To simplify the dataset, we removed the extra header row and retained a single-level header for easier access and processing. Additionally, any empty or missing fields in the dataset were identified and replaced with NaN values to standardize missing data handling.

## Data Integration

Although we initially explored data integration, we ultimately decided against using it in the final model. Specifically, we attempted to combine 15-minute interval data with 60-minute interval data in a single CSV file, with the intention of enhancing the model’s predictive capability. However, during testing, we found that this merged dataset did not improve forecasting performance and introduced additional complexity. As a result, we chose to proceed solely with the 15-minute interval data, which proved to be more consistent and suitable for our modeling approach.

## Feature Engineering

As part of our feature engineering process, we derived two important technical indicators from the original gold price data: the Average True Range (ATR) and the Relative Strength Index (RSI).

The Average True Range (ATR) is a widely used measure of market volatility. It captures the degree of price movement over a given period by considering the true range between high, low and closing prices (Hayes, 2024). In this project, ATR serves as the target variable for prediction, making it a central component of our modeling efforts.

The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements. It ranges from 0 to 100 and is commonly used to identify overbought or oversold market conditions (Fernando, 2024). By including RSI as a feature, we add contextual information about market momentum, which can help the model better understand underlying trends that may affect volatility.

We used these features because ATR directly aligns with our goal of predicting gold market volatility, while RSI provides additional insight into market behaviour that could influence volatility patterns. Together, these engineered features enhance the dataset’s informational depth and improve the model’s ability to make accurate forecasts.

# Modeling

In this section of the report, we outline the machine learning models considered for this project and explain the rationale behind our final model selection. We also provide details on the hyperparameters that were tuned, along with the strategy used for hyperparameter optimization. Lastly, we describe how the dataset was split into training, validation and testing sets to ensure a fair and robust evaluation of model performance.

## Model Selection

In this project, we explored multiple machine learning approaches to forecast gold market volatility. After reviewing different methods, we focused on two main models:

* **Random Forest (RF):**  
  A tree-based ensemble method known for its robustness and ability to model complex, non-linear relationships (*Machine Learning | Google for Developers*, n.d.). It was chosen for its simplicity and strong performance on structured data.
* **Long Short-Term Memory (LSTM):**  
  A type of recurrent neural network (RNN) specifically designed for time series data (Saxena, 2024). LSTM was selected because of its ability to capture patterns over time and handle sequential dependencies in financial data.

Initially, we hypothesized that combining RF and LSTM in an ensemble model might produce superior results by leveraging the strengths of both. However, after testing, we found that neither the Random Forest model nor the ensemble outperformed a simple naïve benchmark. As a result, we shifted our focus to improving the LSTM architecture. With further tuning and adjustments, the improved LSTM model successfully surpassed the benchmark and became our preferred approach.

To evaluate model performance, we used standard regression metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the R² score. These metrics helped us understand how well each model predicted the target variable—Average True Range (ATR)—and guided us in comparing different modeling approaches.

**Explanation for these metrics:**

|  |  |  |
| --- | --- | --- |
| **Metric** | **What it Measures** | **Why it’s useful** |
| **Mean Absolute Error (MAE)** | The average size of the errors between predicted and actual values, without considering direction (*What Is Mean Absolute Error? Formula & Significance*, 2024). | Provides a simple, interpretable measure of average prediction error (*What Is Mean Absolute Error? Formula & Significance*, 2024). |
| **Root Mean Squared Error (RMSE)** | The square root of the average of squared differences between predicted and actual values. Penalizes larger errors more heavily (*What Is Root Mean Square Error? Calculation & Importance*, 2024). | Highlights large prediction errors more clearly than MAE (*What Is Root Mean Square Error? Calculation & Importance*, 2024). |
| **R² Score (Coefficient of Determination)** | The proportion of variance in the target variable that is explained by the model (Fernando, 2024a). | Indicates how well the model fits the data. Values closer to 1 mean better fit (Fernando, 2024a). |

## Parameter Tuning

During model development, we manually tuned several key hyperparameters to improve the performance of our Long Short-Term Memory (LSTM) model. We started with a basic version and gradually enhanced it by increasing the model's depth and training duration. The improved LSTM model included two LSTM layers, with 64 units in the first and 32 in the second and we added dropout layers with a rate of 0.2 after each to help prevent overfitting. The model was trained over 40 epochs using a batch size of 32 and the Adam optimizer was used with a learning rate of 0.001. We also used a look-back window of 10-time steps to capture short-term trends in the time series data.

Our tuning strategy was exploratory and continual. Rather than using automated hyperparameter search techniques, we made adjustments step-by-step and observed their effect on validation performance using metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE). This trial-and-error approach helped us identify an effective configuration for the LSTM model that ultimately outperformed our baseline models.

## Training & Validation

To prepare the data for model training and evaluation, we defined the target variable as the Average True Range (ATR) of the next time step. This means the model is trained to predict future volatility based on current and past market data.

For splitting the data, we used an 80/20 train-test split, carefully maintaining the chronological order of the time series. This means the first 80% of the data was used for training and the last 20% was reserved for testing. We did not shuffle the data, as preserving time order is essential in time series forecasting to ensure that the model is tested on future data, closely mimicking how it would be used in real-world applications.

As a baseline for comparison, we also implemented a naïve benchmark model, which simply predicts the next ATR value as being equal to the previous one. This helps us assess whether our more advanced models, like Random Forest and LSTM, offer meaningful improvements over a simple, rule-based prediction.

# Evaluation

In this section, we compare four models: a Random Forest, an LSTM, an ensemble of both, an LSTM with Random Forest features and a naive benchmark. The models are evaluated using Mean Absolute Error (MAE) and Mean Squared Error (MSE), since this is a regression task.

## Performance Metrics

**Random Forest**

The Random Forest model scored a MAE of 6.62 and a MSE of 136.45. It’s fast to train and relatively easy to interpret (e.g. using feature importance). However, it doesn’t capture time-based relationships, which can be important for time series data.

**LSTM**

The LSTM model performed better, with a MAE of 5.81 and a MSE of 106.03. It can learn patterns over time, which is useful for predicting a sequence like ATR. On the downside, it takes longer to train and is harder to interpret compared to tree-based models.

**Ensemble Model**

Combining the predictions from the Random Forest and LSTM resulted in a MAE of 6.20 and a MSE of 121.21. The performance was in between the two individual models. In this case, the ensemble didn’t outperform the best standalone model (LSTM).

**Ensemble (LSTM + Random Forest average)**

MAE: 6.20

MSE: 121.21

Averaging the predictions gave middle-ground results. It didn’t beat the standalone LSTM and added complexity.

## Model Interpretability

• The Random Forest is easy to explain but lacks time awareness.

• The LSTM handles temporal patterns better but is less interpretable.

• The ensemble did not provide clear improvement over LSTM.

## Comparison with Baseline

The naive benchmark used the previous ATR value as the prediction. Surprisingly, this simple method scored very well: MAE of 0.66 and MSE of 1.18.

The LSTM was the best performing of the trained models, but none of the advanced methods beat the naive benchmark. This suggests that the target variable (ATR) doesn’t change much over time, making it hard for complex models to add value. Using simple methods might be more effective here.

# Deployment

## Deployment Strategy

To deploy the model in a real-world setting, it would need to be integrated into a live system that can automatically process incoming market data and generate volatility predictions. This could be done by setting up a web API or a scheduled script that runs at regular intervals such as every 15 minutes to fetch the latest gold price data, pre-process it and feed it into the trained model. The model would then output the predicted Average True Range (ATR), which could be displayed in a dashboard or used as input for a larger trading or risk management system.

A key part of deployment involves setting up a reliable data pipeline that handles data collection, feature calculation (like RSI), scaling and sequence generation for the LSTM model. The system should also monitor the model's performance over time and trigger retraining when market conditions shift. This helps ensure the model stays accurate and relevant as new data comes in.

However, several challenges may arise during deployment. Live financial data can sometimes be delayed or unavailable, which could disrupt the prediction process. The model may also lose accuracy over time if market behaviour changes, requiring regular updates or retraining.

Building and maintaining the full pipeline can also be technically complex, especially when ensuring it runs smoothly in real-time. Additionally, since this model may influence financial decisions, it's important to ensure transparency and interpretability, especially for users like traders or risk managers. Finally, security, compliance and system reliability are also crucial, particularly if the model is used in a regulated environment or handles sensitive financial data.

## Monitoring

Once the model is deployed, it's important to monitor its performance regularly to ensure it continues to make accurate predictions. This involves comparing the model’s forecasts, such as predicted ATR values, with the actual values as they become available. Key metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) can be calculated in real time to track how well the model is performing. Setting up a simple dashboard can help visualize these metrics over time and alert systems can be put in place to notify the team if the model's accuracy drops unexpectedly.

To maintain and update the model, it should be retrained regularly using the latest data. This helps the model stay up to date with changing market conditions a process known as handling “model drift.” Automated workflows can be created to retrain and redeploy the model on a schedule, ensuring it's always using the most relevant information. Tools like version control and automated testing can help manage updates and make sure new versions perform better before they go live.

Finally, explainability tools can be used to check whether the model is still relying on the right features to make its predictions. Explainability tools are software and systems that provide transparency into how an AI algorithm reaches its decisions (Maayan, n.d.). If the model starts focusing on unusual patterns or if performance drops, it might be time to investigate and update it. By monitoring and maintaining the model this way, you can ensure it stays reliable and useful in real-world trading or forecasting scenarios.

## Future work

Right now, our models are trying to predict the absolute ATR value (e.g. “What will ATR be tomorrow?”).

When a value barely changes, complex models like LSTMs have trouble learning meaningful patterns. They often just learn to “repeat” the last value, which is what the naive model already does.

Instead, we would try to predict the change in ATR, for example:

* Will ATR go up or down tomorrow?
* By how much will it change compared to today?

We then make classifications:

• Class 0: ATR stays the same or goes down

• Class 1: ATR goes up

This is simpler and can still be useful. We could even add a third class for “no significant change.”

Also, another thing we like to mention is that volatility can be captured in different ways. In the future, we would also explore letting the LSTM model decide for itself which data to use for predicting volatility through unsupervised learning.

We could also introduce more features for the models to use. Or even leave some out, but that must be tested by trial and error.

We are aware that this project Is far from implementation, but we still have learnt a lot. We developed a business understanding of the stock market and indicators people use for their trading strategies.

We know that there are more ways to try to predict volatility using machine learning. For this project we are happy with what we learnt along the way, even though we didn’t exactly end up where we initially expected.

# Conclusion

This project set out to build a reliable model for forecasting gold market volatility using ATR as the key metric. Our initial experiments showed that a simple naïve benchmark predicting the next ATR using the previous one performed surprisingly well, highlighting the persistent nature of market volatility.

We evaluated both Random Forest and basic LSTM models, but neither outperformed the benchmark. Ensemble attempts and incorporating Random Forest predictions as features also failed to deliver meaningful improvements.

This report has documented our process from data cleaning and feature engineering to model selection and evaluation, providing a clear rationale behind each step.

# Reverences

Fernando, J. (2024a, November 13). *R-Squared: Definition, Calculation and interpretation*. Investopedia. https://www.investopedia.com/terms/r/r-squared.asp

Fernando, J. (2024b, November 19). *Relative Strength Index (RSI) indicator explained with formula*. Investopedia. https://www.investopedia.com/terms/r/rsi.asp

Finserv, B. (2024, December 25). *Average True Range (ATR)*. www.bajajfinserv.in. https://www.bajajfinserv.in/average-true-range-atr#:~:text=The%20Average%20True%20Range%20(ATR,over%20a%20specified%20time%20period.&text=The%20Average%20True%20Range%20(ATR)%20is%20a%20volatility%20indicator%20providing,assets%20over%20a%20specified%20period.

Hayes, A. (2024, April 11). *Average True Range (ATR) formula, what it means and how to use it*. Investopedia. https://www.investopedia.com/terms/a/atr.asp

Maayan, V. a. P. B. G. D. (n.d.). *Explainable AI: 5 Open-Source Tools You should know*. TDAN.com. https://tdan.com/explainable-ai-5-open-source-tools-you-should-know/31589#:~:text=Explainable%20AI%20tools%20are%20software,fine%2Dtuning%20of%20AI%20systems.

*Machine Learning | Google for Developers*. (n.d.). Google for Developers. https://developers.google.com/machine-learning/decision-forests/random-forests#:~:text=A%20random%20forest%20(RF)%20is,form%20of%20decision%20tree%20ensemble.

Saxena, S. (2024, December 30). *What is LSTM? Introduction to Long Short-Term Memory*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

StatQuest with Josh Starmer. (2018, February 5). *StatQuest: Random Forests Part 1 - Building, using and Evaluating* [Video]. YouTube. https://www.youtube.com/watch?v=J4Wdy0Wc\_xQ

StatQuest with Josh Starmer. (2020a, January 15). *StatQuest: Random Forests Part 2: Missing data and clustering* [Video]. YouTube. https://www.youtube.com/watch?v=sQ870aTKqiM

StatQuest with Josh Starmer. (2020b, August 31). *The essential main ideas of neural networks* [Video]. YouTube. https://www.youtube.com/watch?v=CqOfi41LfDw

StatQuest with Josh Starmer. (2022a, July 11). *Recurrent neural networks (RNNs), clearly explained!!!* [Video]. YouTube. https://www.youtube.com/watch?v=AsNTP8Kwu80

StatQuest with Josh Starmer. (2022b, November 7). *Long Short-Term Memory (LSTM), clearly explained* [Video]. YouTube. https://www.youtube.com/watch?v=YCzL96nL7j0

*What is Mean Absolute Error? Formula & Significance*. (2024, May 27). Deepchecks. https://www.deepchecks.com/glossary/mean-absolute-error/

*What is Root Mean Square Error? Calculation & Importance*. (2024, May 27). Deepchecks. https://www.deepchecks.com/glossary/root-mean-square-error/