

## **Gold Price Prediction using Machine Learning**

The goal of this project is to build a predictive model for the gold price (GLD) using market and economic indicators such as the S&P 500 Index (SPX), Crude Oil ETF (USO), Silver ETF (SLV), and EUR/USD exchange rate.

Gold is a crucial commodity for investors, central banks, and governments, and its price is influenced by various macroeconomic variables. Accurate prediction of gold prices helps in portfolio management, risk mitigation, and strategic investment decisions.

The problem is approached as a regression task, where historical market data is used to train machine learning models, specifically Random Forest Regressor and Gradient Boosting Regressor to predict the price of GLD. The models are evaluated on accuracy and generalization using metrics like  $R^2$ , MAE, and RMSE, and deployed as an interactive web application for real-time predictions.

### **Motivation & Purpose**

Understanding the financial markets, particularly commodities like gold, is critical for investors, economists, and policy makers. This project was built to:

- a. Explore the correlation between macroeconomic indicators and gold prices.
- b. Develop a tool that allows interactive experimentation with input values.
- c. Provide insights into model behavior and accuracy through visualizations.

### **What Problem Does It Solve?**

Gold is a globally traded asset whose price is influenced by various economic indicators.

This tool:

- a. Predicts gold prices using real market data and advanced ML models.
- b. Helps users understand how inputs like oil prices and exchange rates affect gold value.
- c. Provides a quick way for users to run simulations and visualize prediction accuracy.

## **What I Learned**

- a. Implementing and tuning ensemble models like Random Forest and Gradient Boosting.
- b. Building an interactive dashboard using Streamlit.
- c. Visualizing machine learning predictions and interpreting regression metrics.
- d. Creating clean, modular, and user-focused Python apps.

## **What Makes This Project Stand Out?**

- a. Use of real-world financial data for meaningful insights.
- b. Side-by-side model comparison and flexible user input prediction.
- c. A simple yet effective dashboard that is easy to use and expand.

## **Dataset Link:**

File link: <https://www.kaggle.com/datasets/altruistdelhite04/gold-price-data->

### **Features:**

1. SPX - S&P 500 Index
2. USO- US Oil Fund
3. SLV - Silver Price
4. EUR/USD - Exchange Rate
5. GLD - Target: Gold ETF price

## **Techniques Used**

### ***1. Data Preprocessing***

Import Libraries

Exploratory Data Analysis (EDA)

Feature Engineering

### ***2. Feature Selection\_1***

Model selection\_1 : Random Forest Regressor

### ***3. Feature Selection\_2***

Model selection\_2 : Gradient Boosting Algorithm

### ***4. Model Comparison.***

### ***5. Model Evaluation.***

### ***6. Web Deployment.***

## **Models Used**

Random Forest Regressor

Gradient Boosting Method

## **Evaluation**

- a. View model evaluation metrics ( $R^2$ , RMSE, MSE, MAE)
- b. Compare predictions vs actual values based on  $R^2$  especially.
- c. User-input sliders to predict gold price
- d. Choose model dynamically (Random Forest or Gradient Boosting)

## **How to Run**

1. Clone the repository:

[https://github.com/SatadalS99/Gold\\_price\\_prediction](https://github.com/SatadalS99/Gold_price_prediction)

2. Install dependencies:  
Read: Requirement.txt

3. Run the Streamlit app:  
streamlit run gold\_price\_app.py

## **How to Use the Project**

Once the app is running in your browser:

- i. Navigate between sections using the radio buttons.
- ii. View model analysis and compare prediction plots.
- iii. Go to the “Predict Using Input” section.
- iv. Adjust the sliders for SPX, USO, SLV, and EUR/USD.
- v. Choose a model and click “Predict”.
- vi. See your predicted gold price in real-time!

**Requirements:**

VSCode Version: 1.100.2  
Jupyter Notebook Version: 7.2.2  
Python Version: 3.12.7  
Pandas Version: 2.2.2  
Numpy Version: 1.26.4  
scikit-learn Version: 1.5.1  
Matplotlib Version: 3.9.2  
Streamlit Version: 1.37.1

**Future Improvements**

1. Can be fine tuned in both cases.

**How to Contribute**

- i. Fork the repo
- ii. Create your feature branch
- iii. Commit your changes
- iv. Push to the branch
- v. Open a Pull Request

Suggestions and improvements are welcome!