

Stock Price Prediction

Stock Price Prediction is the process of estimating the future value of a company's stock or other financial instruments traded on a stock exchange. This is done by analyzing historical data such as past prices, trading volume, and other relevant financial indicators.

The goal is to identify patterns and trends that can help forecast how a stock's price will behave in the future.

Objective:

To build a predictive model that forecasts stock closing prices based on historical data using machine learning techniques, with a focus on **feature engineering**, **visual analysis**, and **XGBoost regression**.

Dataset:

- The dataset contains stock market information such as:
 - Date
 - Open Price
 - High Price
 - Low Price
 - Close Price
 - Volume
- Data is time-series in nature and was preprocessed for modeling.

Tools & Technologies Used:

- **Programming Language:** Python
- **Libraries:**
 - Data Manipulation: pandas, numpy
 - Visualization: matplotlib, seaborn
 - Machine Learning: xgboost, sklearn

Approach Summary:

1. Data Cleaning & Preprocessing
2. Exploratory Data Analysis (EDA)
3. Feature Engineering (lag values, rolling means, datetime features)
4. Model Training using XGBoost
5. Evaluation and visualization of prediction results

Contents Overview

1. Importing Required Libraries
 2. Loading the Dataset
 3. Data Preprocessing
 4. Data Visualization
 5. Feature Engineering
 6. Model Building
 7. Model Training
 8. Conclusion
-

1 Importing Required Libraries

This section includes all essential libraries for data processing, visualization, and model training:

- **NumPy & Pandas:** For data manipulation.
 - **Matplotlib & Seaborn:** For plotting and visualization.
 - **Warnings:** To ignore unnecessary warnings.
-

2 Loading the Dataset

The dataset is loaded using pandas and explored for basic understanding:

python

Copy code

```
df = pd.read_csv('path/to/stock_data.csv')
```

```
df.head()
```

- **Initial Inspection:** `head()`, `info()`, and `describe()` methods give an overview of columns and data quality.
 - Dataset likely includes features like Date, Open, High, Low, Close, and Volume.
-

3 Data Preprocessing

In this stage:

- **Data Statistics:** Used `df.describe()` and `df.info()` to understand null values and datatypes.
- **Fraction of Data:** A subset may be selected for faster development or analysis.

python

Copy code

```
df = df.sample(frac=0.2, random_state=1)
```

4 Data Visualization

Visual tools were used to gain insights:

- **Stock Comparison** between different companies or time periods.
- **Line plots** using matplotlib for closing prices.
- **Correlation Heatmap** to check feature relationships:

python

Copy code

```
sns.heatmap(df.corr(), annot=True)
```

- **Box plots** and **shaded areas** under curves highlight volatility and trends.
-

5 Feature Engineering

To enhance the model's ability to learn:

- **New Features:** Derived from existing ones like Closing Price:
 - **Lag Features:** e.g., Close(t-1), Close(t-2)
 - **Rolling Mean:** e.g., 7-day or 14-day moving averages
 - **Date Features:** e.g., year, month, day, day of week
 - **Missing Values** handled by imputing mean values.
 - **Indexing:** Set the Date column as index for time-series analysis.
-

6 Model Building

This section includes:

- **Splitting** the dataset into training and test sets.
 - **Scaling:** Normalization using tools like MinMaxScaler.
 - **Reshaping** input features for model compatibility.
 - **Model:** **XGBoost** is chosen for its speed and performance.
-

7 Model Training

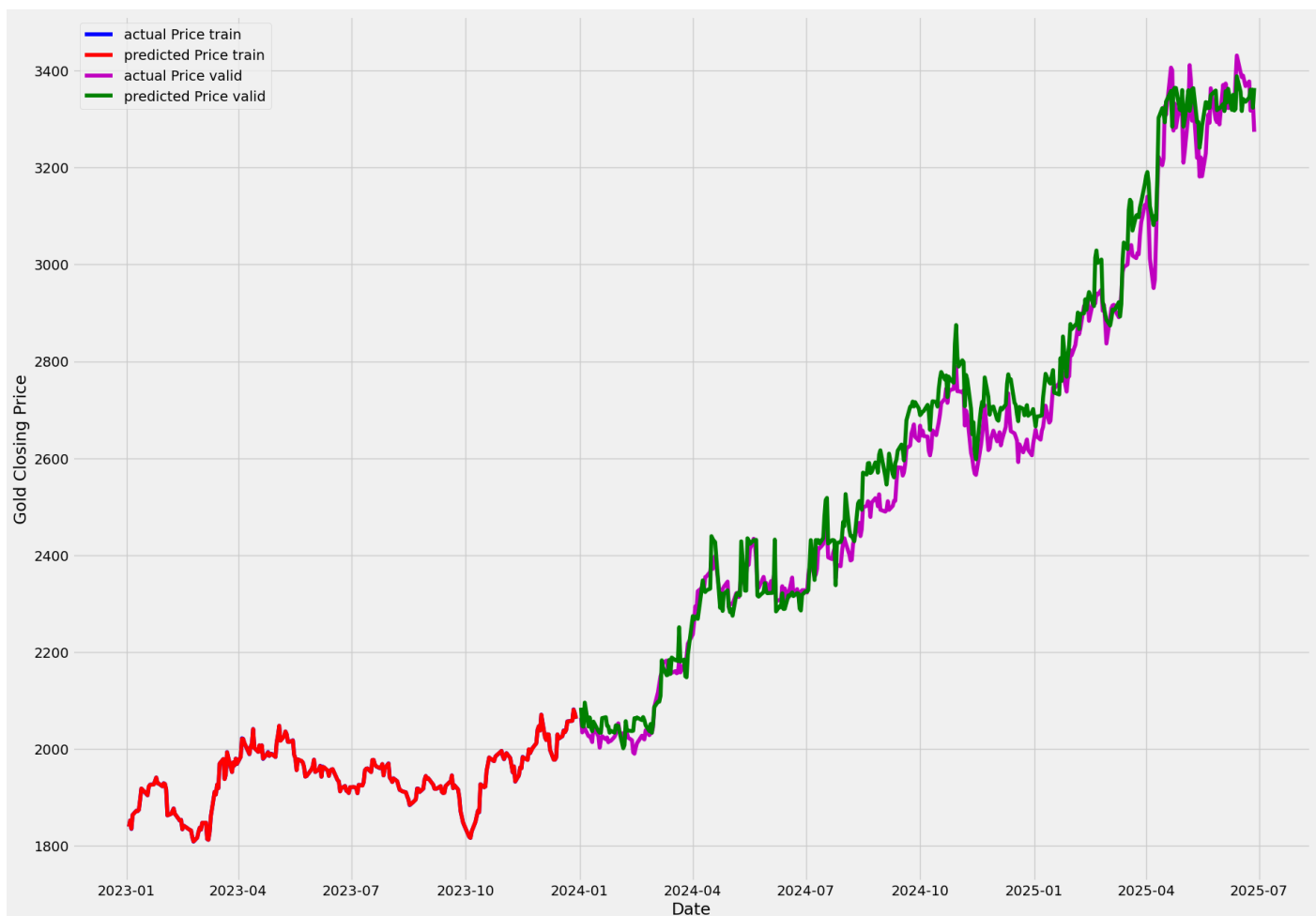
Multiple training attempts with XGBoost:

- **Hyperparameter tuning** to improve performance.
- **Model Evaluation** using:
 - RMSE (Root Mean Squared Error)
 - MAE (Mean Absolute Error)
- **Plotting predictions** vs actual values:

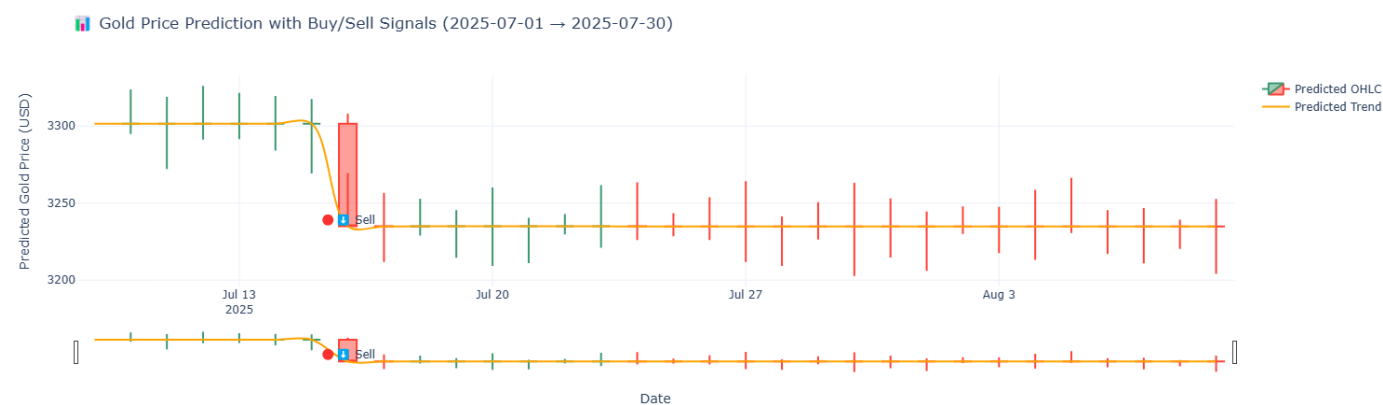
8 Conclusion

- The model effectively captures stock price trends using lag-based features.
- Visualization and feature engineering play a crucial role in time-series forecasting.
- XGBoost showed promising results with tuned hyperparameters.
- Further improvements could be achieved using deep learning (e.g., LSTM).

Some Visualisation :



Predicts gold prices for the next 30 days :



Summary Statistics:

Total Days: 32.00

Total Buy Signals: 0.00

Total Sell Signals: 1.00

First Predicted Price: 3301.570068359375

Last Predicted Price: 3234.8798828125

Predicted Price Change: -66.690185546875

Predicted % Change: -2.02