Gold Price Prediction — Full Documentation + Complete Code

Below is the complete documentation along with ALL the code extracted exactly from your notebook, without adding or modifying anything.

1. Overview

This project trains a regression model (RandomForestRegressor) to predict gold prices.

The notebook contains data loading, preprocessing, EDA, model training, and evaluation.

2. Libraries Used

• pandas

• numpy

• sklearn.model_selection (train_test_split)

• sklearn.ensemble (RandomForestRegressor)

• sklearn.metrics (evaluation metrics)

3. Workflow Steps

---

1. Load the dataset using pandas

2. Perform exploratory data analysis

3. Check correlations

4. Prepare features (x) & target (y)

5. Split into train/test sets

6. Train RandomForestRegressor

7. Predict test values

8. Evaluate using R^2 score and other metrics

4. Important Variables

• gold_data

• x

• y

• regressor

• test_data_prediction

• correlation

• error_score

5. Notes

• No model saving (joblib/pickle) included in notebook

• No missing imports detected

• No undefined variables detected

6. Complete Notebook Code

All code cells from the notebook are included exactly as they appear.

Code Cell 1

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
```

Code Cell 2

```
#loading the csv data
gold_data = pd.read_csv('/content/drive/MyDrive/Data Science/Projects/08 Gold Price Prediciton/gld_p
gold_data.head()
```

Code Cell 3

```
#print last 5 rows of the dataframe
gold_data.tail()
```

Code Cell 4

```
#number of rows and cols
gold_data.shape
```

Code Cell 5

```
#getting some basic information about the data
gold_data.info()
```

Code Cell 6

```
#We will change the date and make it as index , as it is an object
gold_data = gold_data.drop(['Date'], axis=1)
```

Code Cell 7

```
#checking no. of missing values
gold_data.isnull().sum()
```

Code Cell 8

```
#getting some statistical measures of the data
gold_data.describe()
```

Code Cell 9

```
correlation = gold_data.corr()
```

Code Cell 10

```
#constructing heatmap to understand the correlation
plt.figure(figsize=(8,8))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':8}, cmap='
```

Code Cell 11

```
#correlation values of GLD
correlation['GLD']
```

Code Cell 12

```
#check the distribution of the gold price
sns.distplot(gold_data['GLD'], color="yellow")
```

Code Cell 13

```
x = gold_data.drop(['GLD'], axis=1)
y = gold_data['GLD']
```

Code Cell 14

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)
x.shape, x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Code Cell 15

```
regressor = RandomForestRegressor(n_estimators=100)
```

Code Cell 16

```
#Train the model
regressor.fit(x_train, y_train)
```

Code Cell 17

```
#prediction on Test Data
test_data_prediction = regressor.predict(x_test)
test_data_prediction
```

Code Cell 18

```
#R square error
error_score = metrics.r2_score(y_test, test_data_prediction)
print("R squared error : ", error_score*100)
```

Code Cell 19

```
y_test = list(y_test)
plt.plot(y_test, color='blue', label='Actual Value')
plt.plot(test_data_prediction, color='green', label='Predicted Value')
plt.xlabel('Number of values')
plt.ylabel('Gold Price')
plt.legend()
plt.show()
```

Code Cell 20

```
# (This cell is empty)
```