# EX.NO:03 07 /02/2025

AIM:

Develop a linear regression model for forecasting time series data

221501019

To develop a linear regression model for forecasting time series data.

# CODE:

import pandas as pd

data = pd.read\_csv("Gold\_Price\_DataSet.csv")

data.head()

data.drop( 'Volume' ,axis=1 ,inplace = True) data.drop( 'Chg%' ,axis=1 ,inplace = True)

data['Date']=pd.to\_datetime(data['Date']) data = data.set\_index(data['Date'])

import numpy as np

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

data['Date'] = pd.to\_numeric(pd.to\_datetime(data['Date']))

X = data[['Date']] y = data['Price']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression() model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred) r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}") print(f"R-squared: {r2}")

## OUTPUT:

******

import matplotlib.pyplot as plt

# Create the scatter plot plt.figure(figsize=(10, 6))

plt.scatter(X\_test, y\_test, color='blue', label='Actual')

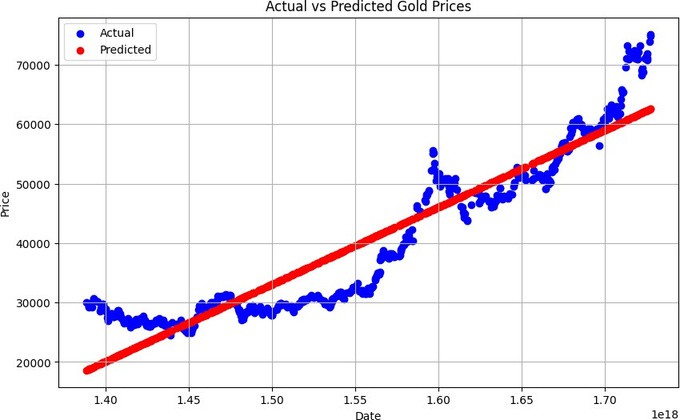
plt.scatter(X\_test, y\_pred, color='red', label='Predicted')

# Customize the plot plt.xlabel("Date") plt.ylabel("Price")

plt.title("Actual vs Predicted Gold Prices")

plt.legend() plt.grid(True)

# Show the plot plt.show()



give = pd.DataFrame([pd.to\_numeric(pd.to\_datetime(['2023-01-01']))]) give.columns = ['Date']

g = model.predict(give) print(g)

# OUTPUT:

[55484.20951941]

## RESULT:

A linear regression model for forecasting time series data has been developed .