EX.NO:03 221501017

07 /02/2025

Develop a linear regression model for forecasting time series data

AIM:

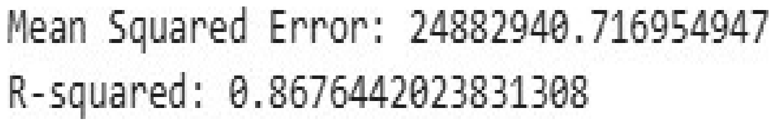
To develop a linear regression model for forecasting time series data.

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)
```

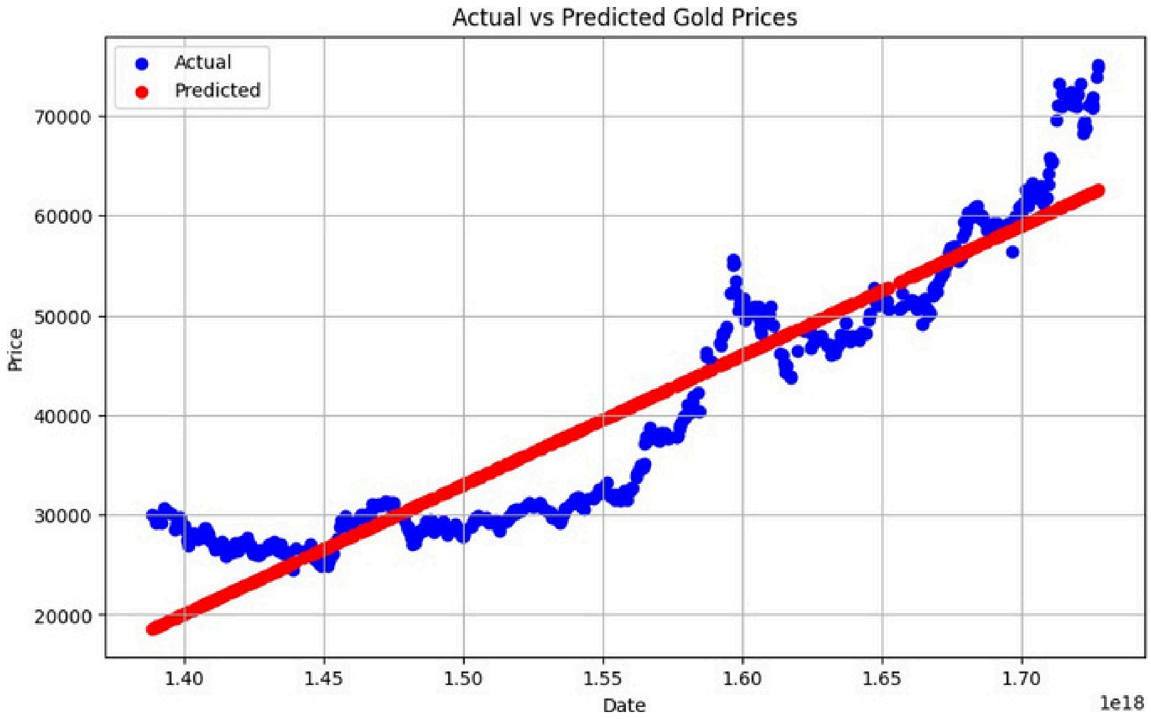
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

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

OUTPUT:



```
Mean Squared Error: 24882940.716954947
R-squared: 0.8676442023831308
```

```
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 .