**4. Develop a linear regression model for forecasting time series data.**

| **EX.N0 : 4** | **DEVELOP A LINEAR REGRESSION MODEL FOR FORECASTING TIME SERIES DATA** |
| --- | --- |
| **DATE : 29/03/2025** |

**AIM:**

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

**PROGRAM:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

df = pd.read\_csv(r"C:\Users\heman\OneDrive\Desktop\abdul\TSA\EX 6\Crude Oil Prices Daily.xlsx", parse\_dates=["Date"]) # Replace with actual file

df = df.sort\_values(by="Date")

df["Time\_Index"] = np.arange(len(df))

X = df[["Time\_Index"]] # Features

y = df["Close"] # Target (gold price)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, shuffle=False)

model = LinearRegression()

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

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

future\_X = np.arange(len(df), len(df) + 30).reshape(-1, 1) # Forecast next 30 days future\_y = model.predict(future\_X)

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

print(f"Mean Squared Error: {mse:.2f}")

plt.figure(figsize=(12, 6))

plt.plot(df["Date"], df["Close"], label="Actual Gold Price", color="blue")

plt.plot(df.iloc[len(X\_train):]["Date"], y\_pred, label="Predicted Gold Price", color="red") plt.plot(pd.date\_range(df["Date"].max(), periods=30, freq='D'), future\_y,

label="Forecasted Price", color="green", linestyle="dashed")

plt.xlabel("Date")

plt.ylabel("Gold Price")

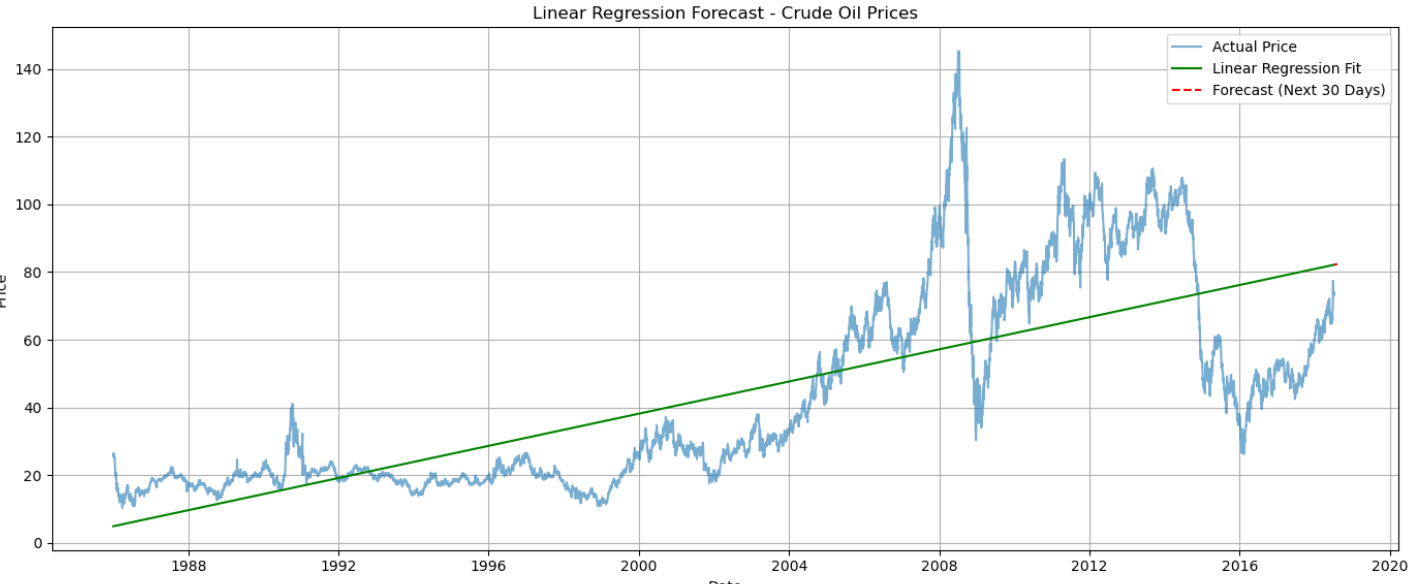
plt.legend()

plt.title("Gold Price Forecasting using Linear Regression")

plt.xticks(rotation=45)

plt.show()

**OUTPUT:**

****

**RESULT:**

Thus, the program for Develop a linear regression model for forecasting time series data is executed successfully.