EX:No.5 DATE:7/02/25	Develop a linear regression model for forecasting time series data.

AIM:

Develop a linear regression model for forecasting time series data.

OBJECTIVE:

To develop a linear regression model to forecast future air pollution levels based on historical data.

BACKGROUND:

- Linear regression models the relationship between dependent and independent variables.
- In time-series forecasting, time (e.g., year, month) can be an independent variable for predicting pollution levels.
- Linear regression can help predict future pollution trends based on historical data.
- The model is simple but effective for linear relationships and can be used for short-term forecasts.

SCOPE OF THE PROGRAM:

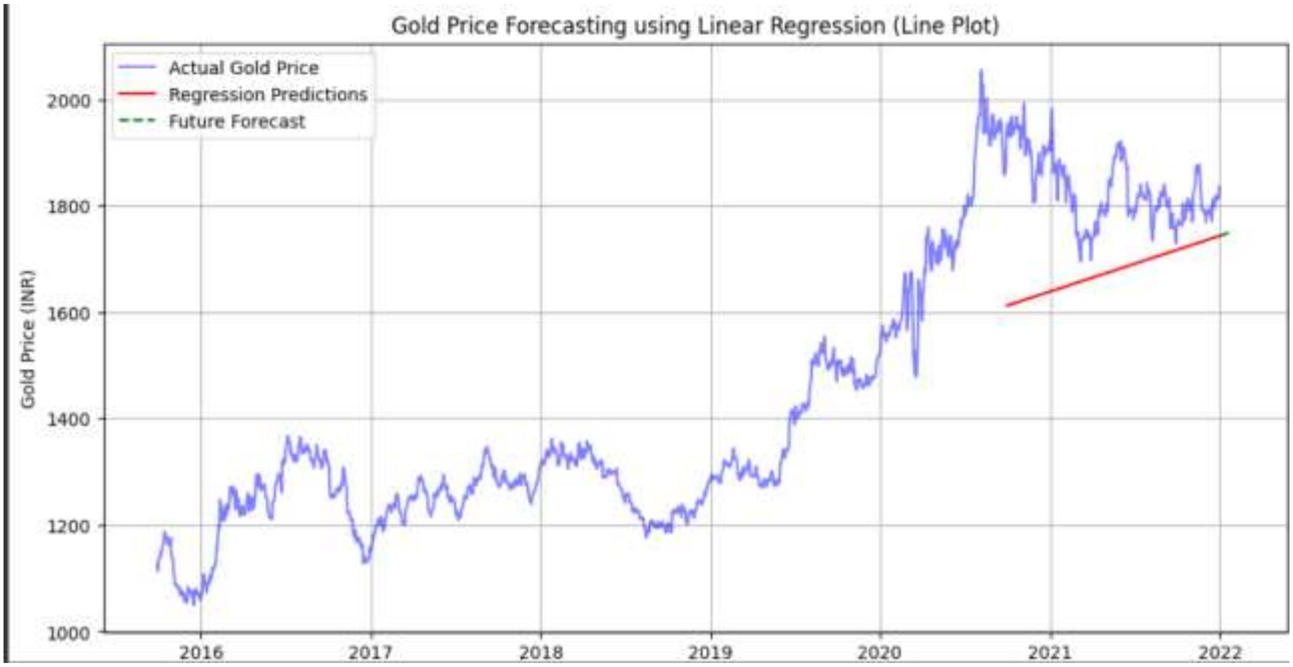
- Load and clean air pollution data (2012-2021).
- Use time (month/year) as a feature for regression.
- Build a linear regression model for predicting future pollution levels.
- Evaluate the model performance with metrics like mean squared error (MSE). CODE:

CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

df = pd.read_csv("/content/gold_data.csv")

df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

gold_price_col = "Price"

df = df[(df['Date'].dt.year >= 2012) & (df['Date'].dt.year <= 2021)]

df.set_index('Date', inplace=True)

df['Date_Ordinal'] = df.index.map(lambda x: x.toordinal())

X = df[['Date_Ordinal']]
y = df[gold_price_col]
```

```
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)

mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print(f"📊 Model Evaluation:")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R² Score: {r2:.4f}")

future_dates = pd.date_range(start=df.index[-1], periods=30, freq='D')  # Predict next 30 days
future_dates_ordinal = future_dates.map(lambda x: x.toordinal()).values.reshape(-1, 1)
future_predictions = model.predict(future_dates_ordinal)

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

plt.plot(df.index, df[gold_price_col], label="Actual Gold Price", color='blue', alpha=0.5)

plt.plot(df.index[-len(y_test):], y_pred, label="Regression Predictions", color='red')

plt.plot(future_dates, future_predictions, label="Future Forecast", color='green', linestyle='dashed')

plt.xlabel("Date")
plt.ylabel("Gold Price (INR)")
plt.title("Gold Price Forecasting using Linear Regression (Line Plot)")
plt.legend()
plt.grid()
plt.show()
```

OUTPUT:



RESULT:
Thus, the program to implement the linear regression model for furcating is executed successfully.