**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 pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Sample population dataset (replace with your actual data or read from CSV)

data = {

'Year': np.arange(2000, 2021),

'Population': [1.2, 1.25, 1.3, 1.35, 1.42, 1.5, 1.55, 1.62, 1.7, 1.75,

1.8, 1.85, 1.9, 1.96, 2.0, 2.05, 2.1, 2.15, 2.18, 2.2, 2.25]

}

df = pd.DataFrame(data)

# Define features (Year) and target (Population)

X = df[['Year']]

y = df['Population']

# Create and train the Linear Regression model

model = LinearRegression()

model.fit(X, y)

# Predict population for the known years

df['Predicted'] = model.predict(X)

# Forecast future population (e.g., for 2021–2030)

future\_years = pd.DataFrame({'Year': np.arange(2021, 2031)})

future\_years['Forecasted\_Population'] = model.predict(future\_years)

# Plotting

plt.figure(figsize=(10, 6))

plt.plot(df['Year'], df['Population'], label='Actual Population', marker='o')

plt.plot(df['Year'], df['Predicted'], label='Predicted Population', linestyle='--')

plt.plot(future\_years['Year'], future\_years['Forecasted\_Population'], label='Forecasted Population', linestyle='-.')

plt.xlabel('Year')

plt.ylabel('Population (in billions)')

plt.title('Population Forecast using Linear Regression')

plt.legend()

plt.grid(True)

plt.tight\_layout()

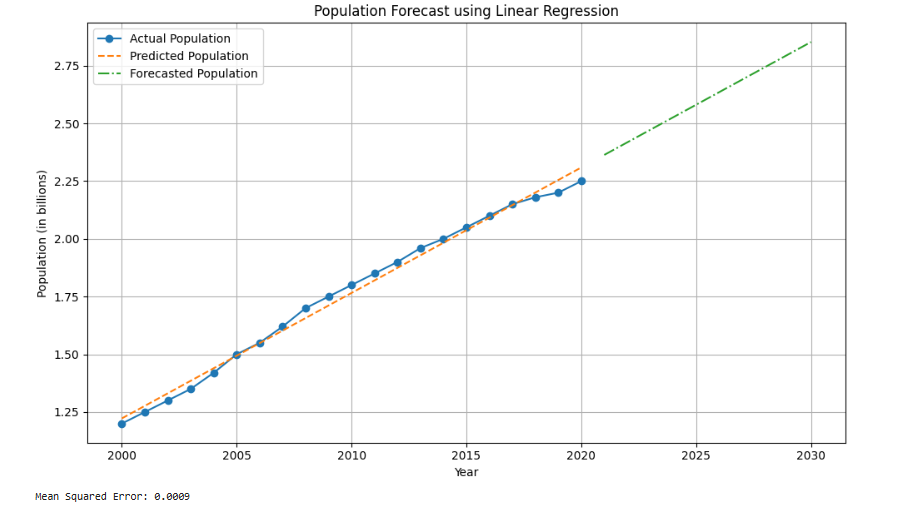
plt.show()

# Optional: Evaluate model

mse = mean\_squared\_error(y, df['Predicted'])

print(f"Mean Squared Error: {mse:.4f}")

**OUTPUT:**

****

**RESULT:**

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