**Program to develop a linear regression model for forecasting time series data.**

**1. Import Libraries**

import pandas as pd

from sklearn.linear\_model import LinearRegression

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

import matplotlib.pyplot as plt

pandas: For reading and handling the dataset.

LinearRegression: The linear regression model from scikit-learn.

mean\_squared\_error, r2\_score: For evaluating the performance of the model.

matplotlib.pyplot: For plotting the results.

**2. Load the Dataset**

data = pd.read\_csv('OzoneHole\_Data.csv')

**3. Prepare the Data**

X = data['Year'].values.reshape(-1, 1)

y\_area = data['Hole Area'].values

y\_ozone = data['Minimum Ozone'].values

X is the feature (independent variable), which is the Year. It's reshaped into a 2D array as required by scikit-learn.

y\_area and y\_ozone are the target variables (dependent variables) for prediction.

### **4. Initialize the Linear Regression Models**

model\_area = LinearRegression()

model\_ozone = LinearRegression()

**5. Train the Models:**

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

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

**6. Make Predictions**

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

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

**7. Evaluate the Models**

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

r2\_area = r2\_score(y\_area, y\_area\_pred)

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

r2\_ozone = r2\_score(y\_ozone, y\_ozone\_pred)

**Mean Squared Error (MSE)** measures the average squared difference between actual and predicted values (lower is better).

**R² Score** measures how well the model explains the variance in the target variable (closer to 1 is better).

### **8. Visualize the Results**

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)

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

plt.plot(X, y\_area\_pred, color='red', label='Predicted Area')

plt.title('Ozone Hole Area Over Time')

plt.xlabel('Year')

plt.ylabel('Hole Area')

plt.legend()

plt.subplot(1, 2, 2)

plt.scatter(X, y\_ozone, color='green', label='Actual Ozone')

plt.plot(X, y\_ozone\_pred, color='orange', label='Predicted Ozone')

plt.title('Minimum Ozone Over Time')

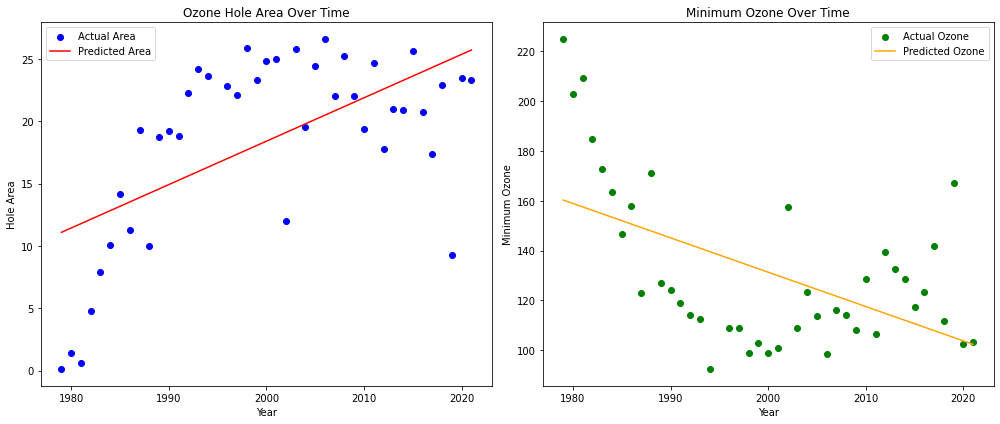
plt.xlabel('Year')

plt.ylabel('Minimum Ozone')

plt.legend()

plt.tight\_layout()

plt.show()



**Result**

Thus successfully implemented a program related to ozone depletion using machine learning