**Code Implementation in Air Qulaity Monitoring.**

**Program:**

import numpy as np

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

# Generate synthetic air quality data

np.random.seed(0)

num\_samples = 1000

temperature = np.random.uniform(10, 30, num\_samples)

humidity = np.random.uniform(30, 70, num\_samples)

time\_of\_day = np.random.uniform(0, 24, num\_samples)

previous\_pm25 = np.random.uniform(0, 50, num\_samples)

previous\_pm10 = np.random.uniform(0, 60, num\_samples)

pm25 = 20 + 1.5 \* temperature - 0.5 \* humidity + 0.1 \* time\_of\_day + 0.2 \* previous\_pm25

pm10 = 15 + 1.0 \* temperature - 0.3 \* humidity + 0.05 \* time\_of\_day + 0.15 \* previous\_pm10

# Step 3: Data Splitting

X = np.column\_stack((temperature, humidity, time\_of\_day, previous\_pm25, previous\_pm10))

y = pm25 # Predicting PM2.5, you can change this to predict other metrics

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.15, random\_state=42)

# Step 5: Model Selection

model = LinearRegression()

# Step 6: Model Training

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

# Step 7: Model Evaluation

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

mae = mean\_absolute\_error(y\_test, y\_pred)

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

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

print("Model Evaluation:")

print(f"Mean Absolute Error: {mae}")

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

print(f"R-squared (R²): {r2}")