## Air Quality Analysis and Prediction Process

### 1. Data Collection:

Collect historical air quality data for Tamil Nadu. This data should include parameters like PM2.5, PM10, NO2, SO2, CO, O3, and meteorological data like temperature, humidity, and wind speed. Data sources can include government agencies, research institutions, or online repositories.

### 2. Data Preprocessing:

Clean and preprocess the data:

* Handle missing values: Use interpolation or imputation methods to fill missing data.
* Outlier detection: Identify and handle outliers in the data.
* Feature engineering: Create new features or transform existing ones. For example, you can calculate daily averages or rolling averages of air quality parameters.

### 3. Data Exploration:

Visualize and explore the data to understand patterns and correlations. Libraries like Matplotlib and Seaborn can help with this.

### 4. Feature Selection:

Select the most relevant features for air quality prediction. You may use domain knowledge or feature selection techniques like feature importance.

### 5. Model Selection:

Choose an appropriate model for air quality prediction. Time series forecasting models like ARIMA or machine learning models like Random Forest, XGBoost, or neural networks can be effective.

### 6. Model Training and Testing:

Split the data into training and testing sets. Train your selected model on the training data and evaluate its performance on the testing data using metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), or R-squared.

### 7. Hyperparameter Tuning:

Optimize model hyperparameters to improve prediction accuracy.

### 8. Prediction:

Use the trained model to make air quality predictions for future time periods.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.arima\_model import ARIMA

# Load your air quality data

data = pd.read\_csv('air\_quality\_data.csv')

# Preprocess the data (fill missing values, remove outliers, etc.)

# Select the relevant features and time series

time\_series = data['Date'] # Assuming you have a date column

air\_quality = data['PM2.5'] # Replace with the appropriate air quality parameter

# Split the data into training and testing sets

train\_size = int(0.8 \* len(air\_quality))

train, test = air\_quality[:train\_size], air\_quality[train\_size:]

# Train an ARIMA model

model = ARIMA(train, order=(5, 1, 0))

model\_fit = model.fit(disp=0)

# Make predictions

predictions = model\_fit.forecast(steps=len(test))

# Evaluate the model

from sklearn.metrics import mean\_absolute\_error

mae = mean\_absolute\_error(test, predictions)

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

# Visualize the results

plt.plot(test.index, test.values, label='Actual')

plt.plot(test.index, predictions, color='red', label='Predicted')

plt.legend()

plt.show()