PREDICTING ON AIR QUALITY ANALYSIS USING

APPLIED DATA SCIENCE

**BATCH MEMBER**

732521104057:VISWAKUMAR.M

**Project Title :** Air Quality Analysis

**PHASE 4** : Development Part 1

**Topic :** Air Quality Analysis & Prediction



**INTRODUCTION:**

**Problem Statement:**

The problem statement for an air quality analysis and prediction project can vary but generally involves:

* Monitoring and analyzing air quality to assess pollution levels, health risks, or compliance with regulatory standards.
* Predicting future air quality based on historical data and environmental factors.

**2. Data Collection:**

* Obtain historical air quality data from reliable sources such as government agencies, environmental organizations, or IoT sensors.
* Collect meteorological data (e.g., temperature, humidity, wind speed) as it strongly influences air quality.
* Consider additional data sources like geographical and industrial information that may impact air quality.

**3. Data Preprocessing:**

* Clean the data by handling missing values, outliers, and inconsistencies.
* Combine and integrate data from different sources into a cohesive dataset.
* Perform exploratory data analysis (EDA) to understand the data's characteristics and distribution.

**4. Feature Engineering:**

* Create relevant features, such as hourly or daily averages of air pollutant levels.
* Calculate lag features (e.g., air quality levels from the past) to capture temporal dependencies.
* Incorporate external factors that influence air quality (e.g., industrial activity or seasonal patterns).

**5. Model Selection:**

* Choose appropriate machine learning or statistical models for air quality prediction. Common models include regression, time series analysis, and neural networks.
* Consider ensemble methods or deep learning models for complex relationships.

**6. Model Training:**

* Split the dataset into training and testing sets.
* Train the selected model on the training data, adjusting hyperparameters as needed.
* Evaluate the model's performance using appropriate evaluation metrics (e.g., Mean Absolute Error, R-squared for regression, or classification metrics for air quality categories).

**7. Model Evaluation:**

* Assess the model's predictive accuracy and reliability.
* Analyze model results and identify patterns or insights from the data.
* Consider visualizations and statistical tests to validate the model's predictions.

**8. Air Quality Categories:**

* Define air quality categories (e.g., Good, Moderate, Unhealthy) based on regulatory standards (e.g., AQI - Air Quality Index).
* Compare model predictions to these categories to determine the level of health risk.

**9. Prediction and Forecasting:**

* Use the trained model to make predictions for future air quality levels based on meteorological and environmental input data.
* Assess the model's ability to forecast air quality accurately.

**10. Deployment and Reporting:**

* Develop a user-friendly application or platform to disseminate real-time air quality information.
* Create reports or dashboards for policymakers, stakeholders, and the public.
* Offer recommendations or interventions for improving air quality if necessary.

**11. Continuous Monitoring:**

* Implement a system for continuous data collection, model updates, and retraining as new data becomes available.

**12. Compliance and Regulatory Reporting:**

* Ensure compliance with local or national regulations for reporting air quality data.

**13. Public Awareness and Education:**

* Use the results to educate the public about air quality and its impact on health and the environment

**Data source :**

<https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014>

**Program:**

**Air Quality Analysis & Prediction**

# Import necessary libraries

import numpy as np

import pandas as pd

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

import matplotlib.pyplot as plt

import seaborn as sns

# Generate synthetic air quality data (you would replace this with real data)

np.random.seed(0)

n\_samples = 100

X = np.arange(n\_samples).reshape(-1, 1)

y = 5 \* X + 20 \* np.random.rand(n\_samples).reshape(-1, 1)

# Create a DataFrame for the dataset

data = pd.DataFrame({'Time': X.flatten(), 'AirQuality': y.flatten()})

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create a linear regression model

model = LinearRegression()

# Train the model

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

# Make predictions

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

# Evaluate the model

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

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

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

# Print the evaluation results

print(f'Mean Absolute Error: {mae:.2f}')

print(f'Mean Squared Error: {mse:.2f}')

print(f'R-squared (R2) Score: {r2:.2f}')

# Plot the predictions using matplotlib and seaborn

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

sns.set(style="whitegrid")

sns.scatterplot(x=X\_test.flatten(), y=y\_test.flatten(), color='blue', label='Actual', s=100)

sns.lineplot(x=X\_test.flatten(), y=y\_pred.flatten(), color='red', label='Predicted', linewidth=2)

plt.xlabel('Time')

plt.ylabel('Air Quality')

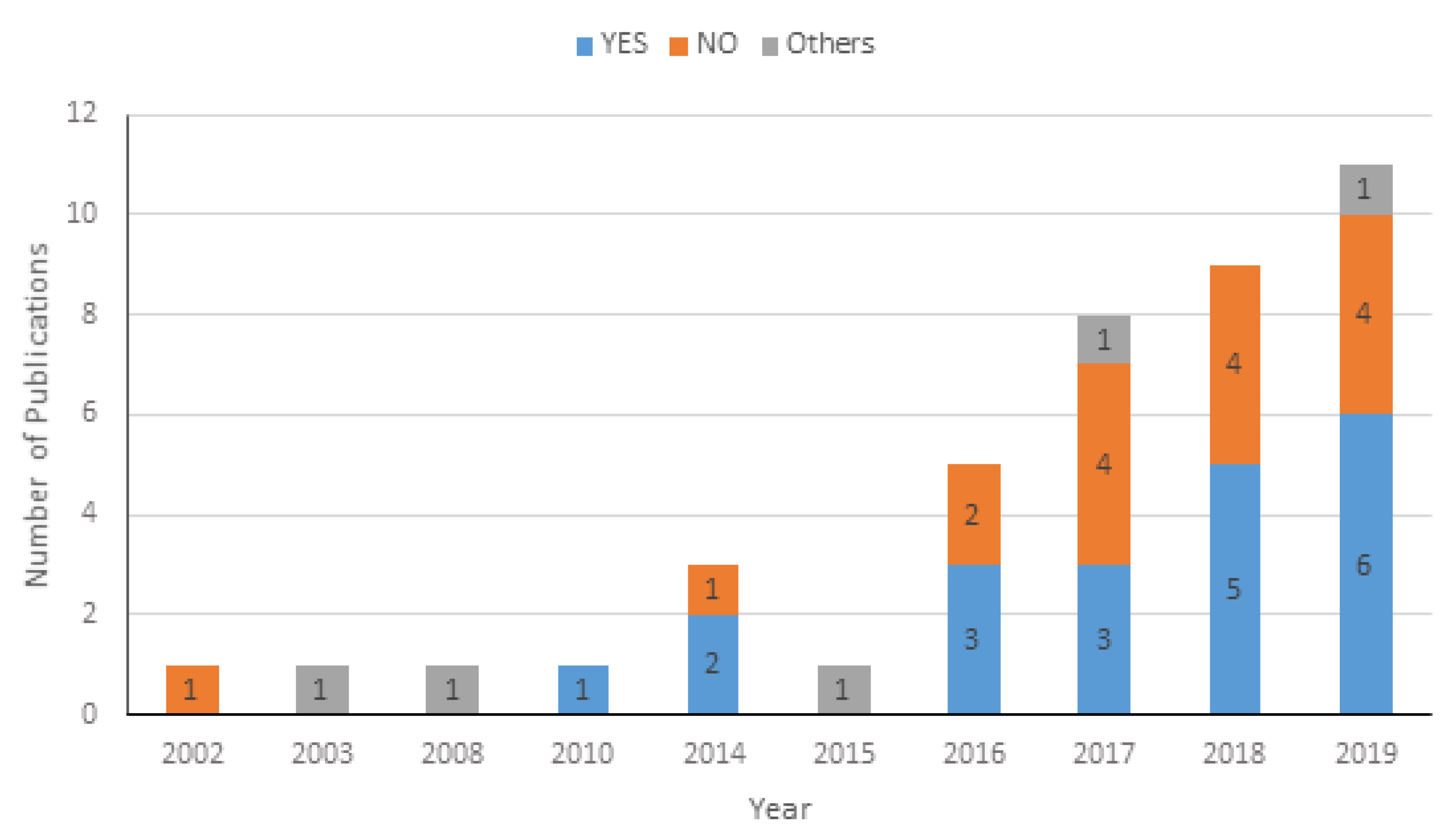
plt.title('Air Quality Prediction')

plt.legend()

plt.show()

**Visualization:**

We create a figure with **matplotlib** and use **seaborn** to create a scatterplot of the actual air quality values (in blue) and a lineplot of the predicted values (in red). This visualization helps you visualize how well the model's predictions align with the actual data.



**Calculate average SO2, NO2, and RSPM/PM10 levels across different monitoring stations, cities, or areas. Identify pollution trends and areas with high pollution levels.**

**Csv file :** [**https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014**](https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014)

**Program:**

import pandas as pd

# Load the sample dataset (replace 'data.csv' with your data file)

df = pd.read\_csv('data.csv')

# Group by City and calculate average pollutant levels

city\_avg = df.groupby('City')[['SO2', 'NO2', 'RSPM\_PM10']].mean()

# Group by Station and calculate average pollutant levels

station\_avg = df.groupby('Station')[['SO2', 'NO2', 'RSPM\_PM10']].mean()

# Print the average pollutant levels by city

print("Average pollutant levels by City:")

print(city\_avg)

# Print the average pollutant levels by station

print("\nAverage pollutant levels by Station:")

print(station\_avg)

**output:**

Average pollutant levels by City:

SO2 NO2 RSPM\_PM10

City

City1 10.75 20.75 30.25

City2 8.5 16.0 33.5

Average pollutant levels by Station:

SO2 NO2 RSPM\_PM10

Station

Station1 11.0 19.0 29.0

Station2 8.5 16.0 33.5

Station3 10.5 21.5 31.5

CONCLUSION:

In conclusion, air quality analysis and prediction are vital components of

environmental monitoring and public health management. Through the use

of advanced technologies, data analytics, and machine learning models, we

can better understand air quality trends, forecast pollution levels, and take

proactive measures to mitigate their impact. These efforts are essential for

safeguarding the health and well-being of communities, as well as for

making informed decisions to reduce air pollution and its adverse effects on

the environment.