KEY FINDINGS FROM THE AIR QUALITY ANALYSIS AND VISUALIZATIONS

Data Preprocessing:

* The code handles missing values in columns 'SO2,' 'NO2,' 'RSPM/PM10,' and 'PM 2.5' by filling them with zeros and converting them to float data types.
* Unnecessary columns 'Stn Code' and 'Agency' are dropped, and the 'Sampling Date' column is renamed to 'year.'

Descriptive Statistics:

* Descriptive statistics, including the mean, median, and standard deviation, are calculated for the 'SO2' levels.
* This provides a summary of the central tendency and spread of the data.

Box Plot:

* A box plot is generated to visualize the distribution of 'RSPM/PM10' levels across different cities or areas.
* This plot helps identify the spread of particulate matter levels, including potential outliers.

Heatmap:

* A heatmap is created to display the average 'RSPM/PM10' levels by city and year.
* This visualization allows for the observation of trends and variations in air quality over time and across different locations.

Time Series Plots:

* Time series plots are presented for 'SO2,' 'NO2,' and 'RSPM/PM10' levels over time.
* These plots reveal the temporal patterns and fluctuations in air pollutant levels, which can be useful for understanding long-term trends.

Overall, the code facilitates data preparation, statistical analysis, and visual exploration of air quality data.The code enables users to assess air pollution levels, detect potential outliers, and identify patterns in air quality measurements over time and across different geographical areas.

CODE:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import folium

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

df = pd.read\_csv('C://air.csv')

print(df.head())

df['SO2']=df['SO2'].fillna(0).astype('str').astype('float')

df['NO2']=df['NO2'].fillna(0).astype('str').astype('float')

df['RSPM/PM10']=df['RSPM/PM10'].fillna(0).astype('str').astype('float')

df['PM 2.5']=df['PM 2.5'].fillna(0).astype('str').astype('float')

df.drop(['Stn Code','Agency'],axis=1,inplace=True)

df=df.rename(index=str,columns={'Sampling Date':'year'})

print(df.info())

average\_so2 = df.groupby('City/Town/Village/Area')['SO2'].mean()

print (average\_so2)

average\_no2 = df.groupby('City/Town/Village/Area')['NO2'].mean()

print (average\_no2)

average\_rspm\_pm10 = df.groupby('City/Town/Village/Area')['RSPM/PM10'].mean()

print (average\_rspm\_pm10)

df['year'] = pd.to\_datetime(df['year'], format='%d-%m-%Y')

df.set\_index('year', inplace=True)

#Descriptive statistics

mean\_so2 = df['SO2'].mean()

median\_so2 = df['SO2'].median()

std\_dev\_so2 = df['SO2'].std()

print(f"Mean SO2 Level: {mean\_so2}")

print(f"Median SO2 Level: {median\_so2}")

print(f"Standard Deviation SO2 Level: {std\_dev\_so2}")

# Box Plot of rspm

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

sns.boxplot(x='City/Town/Village/Area', y='RSPM/PM10', data=df)

plt.xlabel('City/Town/Village/Area')

plt.ylabel('RSPM/PM10 Levels')

plt.title('RSPM/PM10 Levels Across Cities (Box Plot)')

plt.xticks(rotation=45)

plt.show

# Heatmap for rspm

pivot\_table = df.pivot\_table(index='City/Town/Village/Area', columns='year', values='RSPM/PM10', aggfunc='mean')

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

sns.heatmap(pivot\_table, cmap='YlGnBu', annot=True)

plt.xlabel('year')

plt.ylabel('City/Town/Village/Area')

plt.title('Average RSPM/PM10 Levels by City and year')

#plot time series data for so2

plt.figure(figsize=(12, 6))

plt.subplot(3, 1, 1)

plt.plot(df['SO2'], label='SO2 Levels', color='blue')

plt.title('SO2 Levels Over Time')

plt.legend()

# Plot time series data for NO2

plt.subplot(3, 1, 2)

plt.plot(df['NO2'], label='NO2 Levels', color='green')

plt.title('NO2 Levels Over Time')

plt.legend()

# Plot time series data for RSPM/PM10

plt.subplot(3, 1, 3)

plt.plot(df['RSPM/PM10'], label='RSPM/PM10 Levels', color='red')

plt.title('RSPM/PM10 Levels Over Time')

plt.legend()

plt.tight\_layout()

plt.show()