AIR POLLUTION MEASUREMENT AND ANALYSIS

Major Project Report
submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology
in
Information Technology

By

Sumant Kumar	**(19UTIT0062)**
Sahil Sinha	**(19UTIT0056)**
Thakur Vedanshu Raj	**(19UTIT0065)**

DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D
INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

CHENNAI 600062, TAMILNADU, INDIA

APRIL 2023

Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Major Project Report is the bonafide work of "Sumant Kumar (19UTIT0062), Sahil Sinha (19UTIT0056) and Thakur Vedanshu Raj (19UTIT0065)" who carried out the project entitled "Air Pollution Measurement And Analysis" under our supervision from January 2023 to April 2023.

Internal Guide **Head of the Department**

Dr. K. JAYANTHI,M.Tech., Ph.D., **Dr. J. VISUMATHI, M.E., Ph.D.,**

Submitted for Viva Voce Examination held on ______________________________

Internal Examiner **External Examiner**

Submitted for the partial fulfillment for the award of the degree of Bachelor of Technology in Information technology from Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology (Deemed to be University, u/s 3 of UGC Act,1956).

DECLARATION

We Sumant Kumar(19UTIT0062),Sahil Sinha(19UTIT0056), Thakur Vedanshu Raj(19UTIT0065) hereby declare that the Major Project Report entitled " Air Pollution Measurement And Analysis " done by us under the guidance of Dr. Jayanthi Assistant Professor(IT), at Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology , Chennai is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: **SIGNATURE OF THE CANDIDATE**

PLACE: (Sumant Kumar)

(Sahil Sinha)

(Thakur Vedanshu Raj)

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO). DSc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We thankful to our esteemed **Dean SoC, Dr.V.SRINIVASA RAO, Ph.D.,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Head of the Department. Dr. J.Visumathi, Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

A special thanks to our **Project Coordinator Mr. P.N.Karthikayan , M.Tech.,** for his valuable guidance and support throughout the course of the project.

We also take this opportunity to express a deep sense of gratitude to our **Internal Guide Dr. K. JAYANTHI, M.Tech., Ph.D.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

ABSTRACT

Power BI and Regression analysis are critical tools used in statistical analysis. power BI, advanced via Microsoft, is a powerful information visualization device that allows customers to attach, analyze, and visualize facts from numerous assets. It allows groups to make records-driven selections by using growing interactive dashboards, reports, and records visualizations. On the other hand, Regression analysis is a statistical approach used to perceive the connection between a based variable and one or more impartial variables. It facilitates predicting the value of the dependent variable based on the values of the independent variables. Regression fashions, such as linear regression, logistic regression, Lasso regression, Ridge regression, decision tree regression, and random forest regression, are broadly used in diverse fields together with finance, economics, advertising and marketing, and social sciences. The overall performance of these models has evaluated the use of metrics including mean Absolute errors (MAE), Root mean Squared errors (RMSE), and Accuracy score. collectively, energy BI and Regression evaluation provide a powerful toolkit for facts evaluation and choice-making.

LIST OF FIGURES

4.1 **Architecture diagram** 11

4.2 **Dataflow diagram** 13

4.3 **Use case diagram** 15

4.4 **Class diagram** 17

4.5 **PowerBi Dashboard** 19

4.6 **Forecasting Analysis** 20

4.7 **AQI vs Pollutant Levels** 22

4.8 **Visualizing using Heatmap** 23

TABLE OF CONTENTS

Page.No

ABSTRACT i

LIST OF FIGURES ii

1 INTRODUCTION 1

1.1 Objective 2

1.2 Aim of the project 2

1.3 Scope of the Project 2

1.4 Problem statement 3

2 LITERATURE REVIEW 4

3 PROJECT DESCRIPTION 9

3.1 Existing System 9

3.2 Proposed System 9

3.3 System Specification 9

3.3.1 Hardware Specification 9

3.3.2 Software Specification 10

4 METHODOLOGY 11

4.1 General Architecture 11

4.2 Design phase 13

4.2.1 Data Flow Diagram 13

4.2.2 Use Case Diagram 15

4.2.3 Class Diagram Diagram 17

4.3 PowerBi Dashboard 19

4.4 Forecasting Analysis 20

4.5 AQI vs Pollutant Levels 22

4.6 Visualizing using Heatmap 23

4.7 Algorithm 24

4.7.1 Linear Regression 24

4.7.2 Logistic Regression 24

4.7.3 Lasso Regression 24

4.7.4 Ridge Regression 25

4.7.5 Decision Tree Regression 25

4.7.6 Random Forest Regression 25

5 IMPLEMENTATION 27

5.1 Code 27

6 RESULTS AND DISCUSSIONS 38

6.1 Efficiency of the proposed system 38

6.2 comparison of existing and proposed system 38

6.3 Result 39

7 CONCLUSION AND FUTURE ENHANCEMENTS 40

7.1 Conclusion 40

7.2 Future Enhancements 40

Reference 41

Chapter 1

INTRODUCTION

Air pollution is a first-rate international hassle, and India is no exception. With an unexpectedly developing economy and urban population, India faces significant challenges in maintaining air quality. The united states are one of the biggest members of its CO2 emissions in the world, and lots of its towns are frequently exposed to high stages of air pollution that pose huge health risks to its population. The effects of air pollution vary from respiratory problems to coronary heart sickness and might even result in the untimely loss of life in excessive instances. To deal with this important difficulty, the authorities of India have delivered Air satisfactory standards and indicators to reveal and adjust the extent of pollution in the air. An Air first-class Index gives a complete assessment of air pleasant in a specific vicinity and helps coverage makers make informed decisions to lessen pollutants stages. In the latest years, devices getting to know algorithms and regression models have emerged as effective gear for studying and predicting air high-quality indices. by using reading historical statistics on pollution levels, those fashions can offer a perception of air pollution developments and styles in specific regions. These records can be used to increase powerful techniques to control pollution levels and improve air best. This proposed painting objectives are to use machine mastering algorithms and regression fashions to analyze and are expecting the air best index of fundamental cities in India. by way of making use of those tools to historic pollutants facts, we hope to benefit the perception of the principal resources of air pollutants in those cities and develop powerful pollutants mitigation techniques. This painting can make contributions to the development of more powerful regulations and projects to improve air high-quality and protect public health in India.

1.1 Objective

- To analyze and measure the quality of the air as well as components present in the top most cities of India.
- To determine the most usual supervised machine learning algorithms that were used to predict the AQI by performing a literature review
- To evaluate the performance and accuracy of created models and determine the most accurate machine learning algorithm in the prediction of AQI.

1.2 Aim of the project

The primary aim of this study is to determine the most accurate machine learning algorithm to build a model that can predict the AQI. To build and analyze Air Quality Index of the major cities

1.3 Scope of the Project

- Air quality monitoring instruments can be used to analyze and find out harmful air indexes present in the environment.
- Parameters like carbon monoxide (CO), sulfur dioxide (SO2), nitrogen oxides (NOx), particulate matter (PM2.5), and other harmful gases are measured by these devices.
- CO is produced by incomplete combustion of fossil fuels and can cause headaches, dizziness, nausea, and even death in high concentrations.
- SO2 is released by the burning of coal and oil and can cause respiratory problems, particularly in people with asthma or other lung conditions.
- NOx includes nitrogen dioxide (NO2) and nitric oxide (NO) and can cause respiratory problems, particularly in people with asthma or other lung conditions.
- PM2.5 refers to tiny particles that can be inhaled deep into the lungs and are released by vehicle exhaust, industrial processes, and burning fossil fuels.
- Air quality monitoring is important to assess pollution levels and identify sources of pollution.
- Governments, industries, and individuals can use air quality monitoring data to make informed decisions about policies, practices, and behaviors that impact air quality.

1.4 Problem statement

• To reduce air-polluting gases like SO2, CO, NO2, etc. and mitigate their harmful effects on human health, it is essential to monitor the air quality index (AQI) of different cities and take appropriate measures to reduce the levels of these harmful gases. The AQI is a measure of how polluted the air is, and it is calculated based on the concentrations of different pollutants in the air.

• To analyze the AQI of different cities, we can identify areas with high levels of pollution and take steps to reduce them. For example, we can reduce the emissions from factories and power plants by implementing stricter emission standards, using cleaner fuels, and investing in renewable energy sources. We can also reduce emissions from vehicles by promoting public transportation, encouraging the use of electric vehicles, and promoting carpooling.

• To reduce the harmful effects of air pollution on human health, we can also take measures such as planting more trees and creating green spaces, which can help absorb pollutants from the air. We can also encourage people to use masks and air purifiers to reduce exposure to harmful pollutants.

• To reduce the levels of harmful gases in the air, it is also essential to take measures to protect vulnerable populations, such as children, the elderly, and people with respiratory problems, from the harmful effects of air pollution. This may involve providing better access to healthcare and encouraging people to adopt healthy lifestyles, such as exercising regularly and eating a healthy diet.

Chapter 2

LITERATURE REVIEW

1. Wang et al., 2022 The CNN-ILSTM version supplied via Wang et al. in 2022 extends previous studies on air high-quality prediction by combining the strengths of convolutional neural networks (CNNs) and stepping forward lengthy quick-time period reminiscence (ILSTM) networks. The version uses CNNs to extract applicable features from enter statistics, that's then fed into an ILSTM network that utilizes a progressed enter gate and forgets about the gate, in addition to a Conversion Information Module (CIM) to save supersaturation at some point of the gaining knowledge of technique. The version became tested on air first-class information from Shijiazhuang metropolis, Hebei Province, China, and became in comparison to eight different prediction fashions which include SVR, RFR, MLP, LSTM, GRU, ILSTM, CNN-LSTM, and CNN-GRU. The CNN-ILSTM version finished an advanced overall performance in phrases of mean absolute mistakes (MAE), suggest squared errors (MSE), and R2, demonstrating the effectiveness of the proposed approach. Universal, the CNN-ILSTM version contributes to the continuing research on air excellent prediction with the aid of providing a greater accurate and efficient approach that incorporates both CNNs and ILSTMs. This has crucial implications for mitigating the terrible effects of air pollutants on human health and the surroundings.

2. Bekkar et al., 2021 In their paper, Bekkar et al. proposed a deep mastering method for predicting the awareness of particulate matter with a diameter of much less than 2.5um (2.5PM). They used a convolutional neural community (CNN) to extract applicable features from the air nice facts and a protracted short-term reminiscence (LSTM) network to version the temporal dependencies in the information. The version become skilled and tested using facts accrued from six air pleasant monitoring stations in Casablanca, Morocco. The authors compared their approach to several different devices getting to know fashions, such as linear regression, choice tree, and random forest. They determined that their proposed CNN-LSTM model outperformed the other fashions, accomplishing a mean absolute error (MAE) of 7. fifty-two µg/m³ and a coefficient of willpower (R^2) of 0.83. The consequences of this have a look at advocate that deep gaining knowledge of approaches, such as CNN-LSTM, may be effective for predicting 2.5PM concentrations in urban environments. Such models can assist policymakers and public health officials in making choices about air excellent control and reducing exposure to harmful pollution.

3. Xu et al., 2022 Chuanqi et al. (2022) explored the evolution of air pollution and its consequences on human health in North China. They performed a study in 10 towns in North China, such as Beijing, Tianjin, and Hebei, and gathered air pollutant statistics from 2015 to 2020. The effects confirmed that the common concentrations of PM2.5, PM10, SO2, NO2, and O3 had been higher than the country-wide air nice standards. the highest awareness of PM2.five turned into determined in wintry weather, and the highest attention of O3 turned into discovered in the summer time. They have a look at also located that exposure to air pollution was related to respiration sicknesses, cardiovascular sicknesses, and unfavourable being pregnant effects. The authors endorsed that measures need to be taken to reduce air pollutant emissions, which include lowering coal intake and selling smooth energy, to improve air quality and defend human fitness in North China. This examination presents treasured insights into the impact of air pollution on human health and highlights the want for powerful rules and strategies to mitigate the harmful results of air pollution.

4. Sun and Liu (2022) The study by Sun and Liu (2022) proposes a novel approach for Air Quality Index (AQI) prediction based on a combination of the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), Autoregressive Moving Average (ARMA), and Long Short-Term Memory (LSTM) models. The authors collected AQI data from three cities in China (Beijing, Shanghai, and Guangzhou) from 2015 to 2020, and used the proposed model to predict AQI levels up to 24 hours in advance.The CEEMDAN model was used to decompose the original AQI time series into a set of Intrinsic Mode Functions (IMFs), which capture different temporal scales of variability in the data. The ARMA model was used to forecast the short-term trend of each IMF, and the LSTM model was used to model the long-term dependencies and predict the overall AQI level. The proposed approach was compared with several other machine learning models, including Support Vector Regression (SVR), Random Forest (RF), and Gradient Boosting Decision Tree (GBDT), and was found to outperform them in terms of prediction accuracy. The results showed that the proposed approach achieved high accuracy in AQI prediction, with mean absolute error (MAE) and Root Mean Squared Error (RMSE) values ranging from 5 to 17 μg/m³ and 7 to 23 μg/m³, respectively. The authors suggest that this approach can be used to provide accurate and timely AQI predictions, which can help policymakers and the public take proactive measures to mitigate the impact of air pollution on public health and the environment. Overall, the study provides a promising new method for AQI prediction that combines multiple models and can be used to improve air quality management strategies in urban areas.

5. Raja et al. (2022) The study by Raja et al. (2022) aims to predict crop yield based on the characteristics of the agricultural environment using various feature selection techniques and classifiers. The authors collected data on environmental factors such as temperature, rainfall, and soil moisture from six different agricultural regions in India and used machine learning algorithms to predict crop yield. The authors employed various feature selection techniques, including principal component analysis (PCA), correlation-based feature selection (CFS), and genetic algorithm (GA), to identify the most important features for predicting crop yield. They then compared the performance of different classifiers, including decision tree, random forest, k-nearest neighbor (KNN), support vector machine (SVM), and artificial neural network (ANN), in predicting crop yield.The results showed that SVM and ANN had the highest accuracy in predicting crop yield, with SVM achieving an accuracy of 90.2 and ANN achieving an accuracy of 92.8. The feature selection techniques also had a significant impact on the performance of the classifiers, with GA and PCA yielding the highest accuracies. The authors suggest that their approach can help farmers and policymakers make informed decisions about crop management and resource allocation. The findings of this study can also inform the development of precision agriculture systems that optimize crop yield based on environmental factors. Overall, this study highlights the potential of machine learning algorithms in predicting crop yield and improving agricultural productivity.

6. Raja et al. (2022) The study by Zaib et al. (2022) aims to analyze the spatio-temporal characteristics of the Air Quality Index (AQI) in Northwest China using various statistical methods. The authors collected AQI data from 14 cities in the region from 2015 to 2020 and used methods such as spatial autocorrelation analysis, cluster analysis, and principal component analysis to identify patterns and trends in AQI levels.The authors found that the AQI levels in the region were generally high, with more than 70 of the days exceeding the national air quality standards. The highest AQI levels were observed in the winter months, with a peak in January. The spatial analysis revealed that AQI levels were higher in the southern and eastern parts of the region, which are more urbanized and have higher levels of industrial activity. The cluster analysis identified three distinct clusters of cities based on their AQI levels, with the cities in the eastern part of the region forming a separate cluster with the highest AQI levels. The principal component analysis revealed that the main factors contributing to AQI variability in the region were PM2.5, PM10, SO2, and NO2. The authors suggest that reducing emissions from industrial and transportation sources could help improve air quality in the region. The findings of this study provide valuable insights into the spatio-temporal patterns of AQI in Northwest China and can inform policies and interventions aimed at reducing air pollution levels in the region.

7. Yang and Zhong (2022) The study by Yang and Zhong (2022) aimed to investigate the spatio-temporal evolution and influencing factors of the Air Quality Index (AQI) in China. The authors collected AQI data from 366 cities across China from 2015 to 2020 and used statistical methods such as trend analysis, cluster analysis, and regression analysis to identify patterns and factors influencing AQI levels.The authors found that the AQI levels in China had gradually improved over the years, with a decrease in the number of days with poor air quality. However, there were still significant spatial variations in AQI levels across the country, with higher levels observed in the eastern and central regions. The cluster analysis identified four distinct clusters of cities based on their AQI levels, with the cities in the eastern and central regions forming separate clusters with higher AQI levels.The regression analysis revealed that the main factors contributing to AQI variability in China were industrial activity, transportation, and meteorological factors such as wind speed and temperature. The authors suggested that reducing emissions from industrial and transportation sources and improving meteorological monitoring and forecasting could help further improve air quality in China.Overall, this study provides a comprehensive analysis of the spatio-temporal evolution and influencing factors of AQI levels in China. The findings can inform policies and interventions aimed at reducing air pollution levels and improving public health in the country.

8. Li et al. (2022) Li et al. (2022) present a new approach for predicting the Air Quality Index (AQI) using an evolutionary deep learning model based on an improved grey wolf optimization algorithm and Deep Belief Network-Extreme Learning Machine (DBN-ELM). The authors collected AQI data from a monitoring station in Zhengzhou, China, for the period from January 2016 to December 2020. The data were preprocessed and feature-selected using the correlation-based feature selection (CFS) method.The authors then proposed an improved grey wolf optimization algorithm (IGWO) to optimize the parameters of the DBN-ELM model. The proposed IGWO algorithm improves the original grey wolf optimization algorithm by introducing adaptive parameter control and randomization of the search space to increase the exploration and exploitation abilities of the algorithm.The results of the study showed that the proposed approach achieved high accuracy in predicting the AQI compared to other existing methods. The authors also performed a sensitivity analysis to evaluate the contribution of different input variables to the AQI prediction. The analysis showed that the PM2.5, PM10, NO2, and SO2 were the most important variables for predicting the AQI.The proposed approach offers a novel and effective method for predicting the AQI, which can be used to support decision-making for air quality management and control. The results demonstrate the potential of using advanced machine learning techniques for improving air quality monitoring and prediction.

9. Benchrif et al. (2021) The study by Benchrif et al. (2021) aimed to assess the impact of COVID-19 lockdowns on air quality in cities with more than one million inhabitants in Morocco. The authors collected air quality data, specifically Air Quality Index (AQI), PM2.5, and NO2, from five cities during three different lockdown phases in 2020. They compared the air quality data during the lockdown periods with the data from the same periods in the previous year.The authors found a significant reduction in AQI, PM2.5, and NO2 levels during the lockdown periods compared to the same periods in the previous year. The largest reductions were observed during the strictest lockdown period, with AQI levels decreasing by up to 65, PM2.5 levels decreasing by up to 53, and NO2 levels decreasing by up to 77. The authors also found that the reductions in air pollutants were more significant in the cities with higher levels of industrial activity.The study highlights the potential for significant improvements in air quality during lockdown periods due to reduced human activities, such as transportation and industrial activities. The authors suggest that this study can inform policies and interventions aimed at reducing air pollution levels in cities, including promoting remote work, reducing industrial activities, and increasing the use of clean energy sources. The findings of this study can also help in understanding the relationship between human activities and air quality and inform future research on this topic.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The current method used for prediction of Air Pollutant is Air Pollution Prediction using deep learning approach in this approach Air pollution index is find out on the basis of particulate matter size with a diameter of less than 2.5um(2.5PM) but is unable to determine the amount of pollutant gases that is available in our environment and which we are inhaling daily. along with that it doesn't have much accuracy. it is having accuracy near to 82After knowing about pollutant gases people can get aware and can reduce harmful diseases like asthma, viral fever, infections etc.

3.2 Proposed System

We'll analyse and measure the air quality index using Machine learning algorithm. Instead of pm(particle matter) size we'll find out components of harmful gases in particular region. Along with that in this proposed system various Regression models are going to be used to find out the best accuracy level. For Visualization of data set we'll use PowerBi dashboard and will predict the future prediction of Air quality. It will help in tackling with pollution and preventing the life of human being from various diseases.

3.3 System Specification

The system uses Jupiter Notebook as a software for implementation, analysis and measurement. PowerBi is used for visualization of data.

3.3.1 Hardware Specification

- Processor - Intel or high
- RAM – 1024mb
- Space on disk– 100mb
- Device- Any device that can access internet

- Min Space-20mb

3.3.2 Software Specification

- Operating system- Any OS
- Network- wifi or internet
- Jupiter Notebook
- PowerBi

Chapter 4

METHODOLOGY

4.1 General Architecture

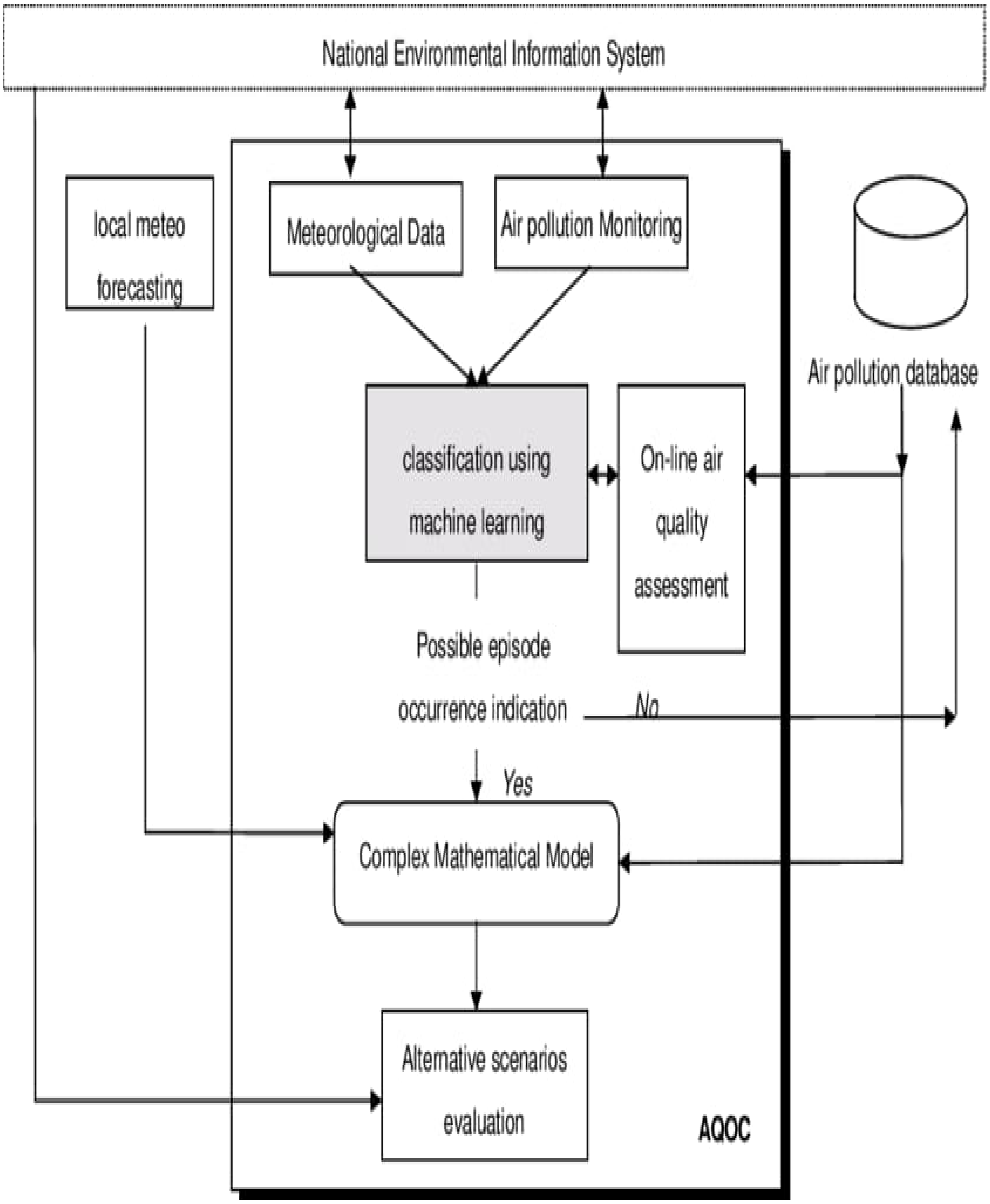

National Environmental Information System
local meteo forecasting
Meteorological Data
Air pollution Monitoring
Air pollution database
classification using machine learning
On-line air quality assessment
Possible episode occurrence indication
No
Yes
Complex Mathematical Model
Alternative scenarios evaluation
AQOC

Figure 4.1: **Architecture diagram**

In the Figure 4.1 The architecture diagram of the Air quality index measurement and analysis is shown. In this Architectural diagram the data set is taken from kaggel or National Environmental Information System and the monitoring of air pollution is being measured based on machine learning algorithm such as regression models. Later on the performance matrix need to be examined to predict the best suitable accuracy of the model. IN performance matrix Mean Square Error(MSE) or Root Mean Square Error(RMSE) is to be calculated.

4.2 Design phase

4.2.1 Data Flow Diagram

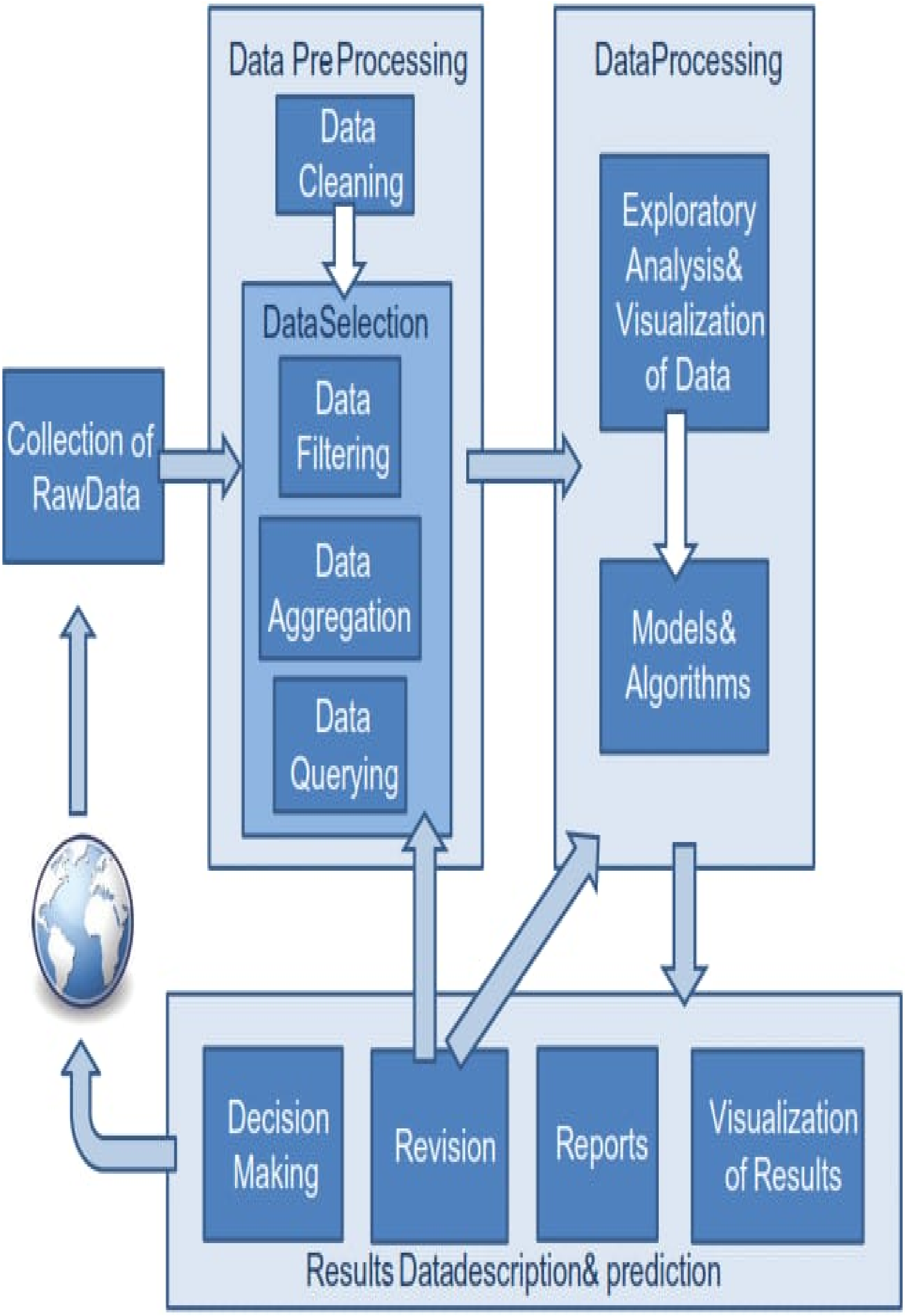


Figure 4.2: **Dataflow diagram**

In this Figure 4.2 A data flow diagram (DFD) is a visual representation of the flow of data and information within a system or process. The system starts with the collection of historical data on air quality in major cities in India. This data is then fed into the machine learning algorithms and regression models for analysis and prediction of the air quality index. The output of the analysis is then used to identify the main sources of air pollution in the cities and develop effective strategies to mitigate pollution levels. These strategies are then implemented, and the resulting changes in air quality are measured and recorded. The data flow diagram for this process would show the inputs of historical air quality data, the processing and analysis done by the machine learning algorithms and regression models, and the output of recommended pollution mitigation strategies. It would also show the implementation of these strategies and the resulting changes in air quality, creating a feedback loop to inform further analysis and strategy development. Overall, the Data flow diagram helps to illustrate the flow of data and information in the proposed project, from data collection to policy implementation and evaluation.

4.2.2 Use Case Diagram

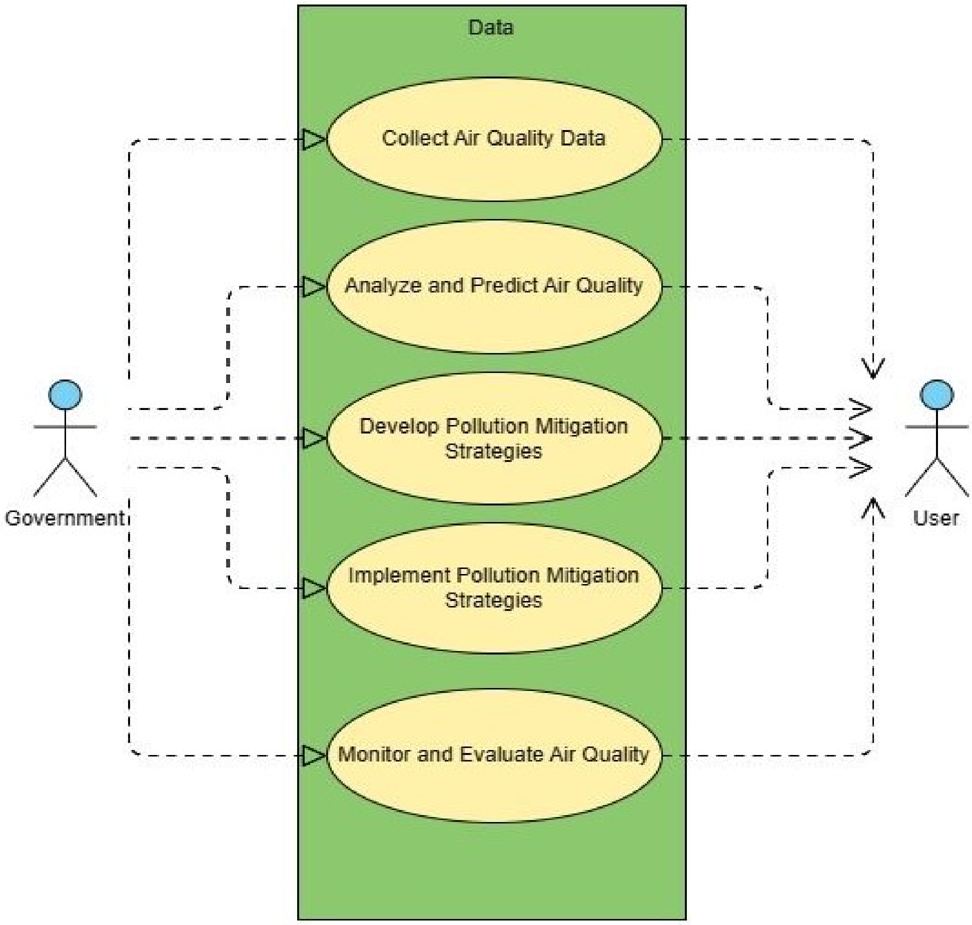

Data
Collect Air Quality Data
Analyze and Predict Air Quality
Develop Pollution Mitigation Strategies
Implement Pollution Mitigation Strategies
Monitor and Evaluate Air Quality
Government
User

Figure 4.3: **Use case diagram**

In this fig 4.3 user case diagram of proposed system is shown. A use case diagram is a type of UML diagram that represents the interactions between a system and its external actors or users. In the context of the proposed project, the following use case diagram explanation can be provided: The primary actor in the system would be the ”Government” or ”Environmental Agency,” which would interact with the system to monitor and improve air quality in major cities in India. The use cases for the system would include: ”Collect Air Quality Data”: This use case involves collecting air quality data from various sensors and sources in each city, including industrial emissions, vehicular pollution, and other sources of air pollution. ”Analyze and Predict Air Quality”: This use case involves using machine learning algorithms and regression models to analyze historical air quality data and predict future air quality levels. ”Develop Pollution Mitigation Strategies”: This use case involves using the output of the analysis and prediction to identify the main sources of air pollution in each city and develop effective strategies to mitigate pollution levels. ”Implement Pollution Mitigation Strategies”: This use case involves implementing the recommended pollution mitigation strategies in each city, including regulatory policies, technological solutions, and public awareness campaigns. ”Monitor and Evaluate Air Quality”: This use case involves monitoring and evaluating the effectiveness of the pollution mitigation strategies, measuring changes in air quality, and providing feedback to inform further analysis and strategy development. The use cases would be represented by ovals on the use case diagram, with lines connecting them to the primary actor or other actors who interact with the system. The use case diagram would also show the relationships between the use cases, indicating which use cases are dependent on others and which can be performed independently. Overall, the use case diagram helps to illustrate the interactions between the system and its external actors, providing a clear visual representation of the system’s purpose and functionality.

4.2.3 Class Diagram Diagram

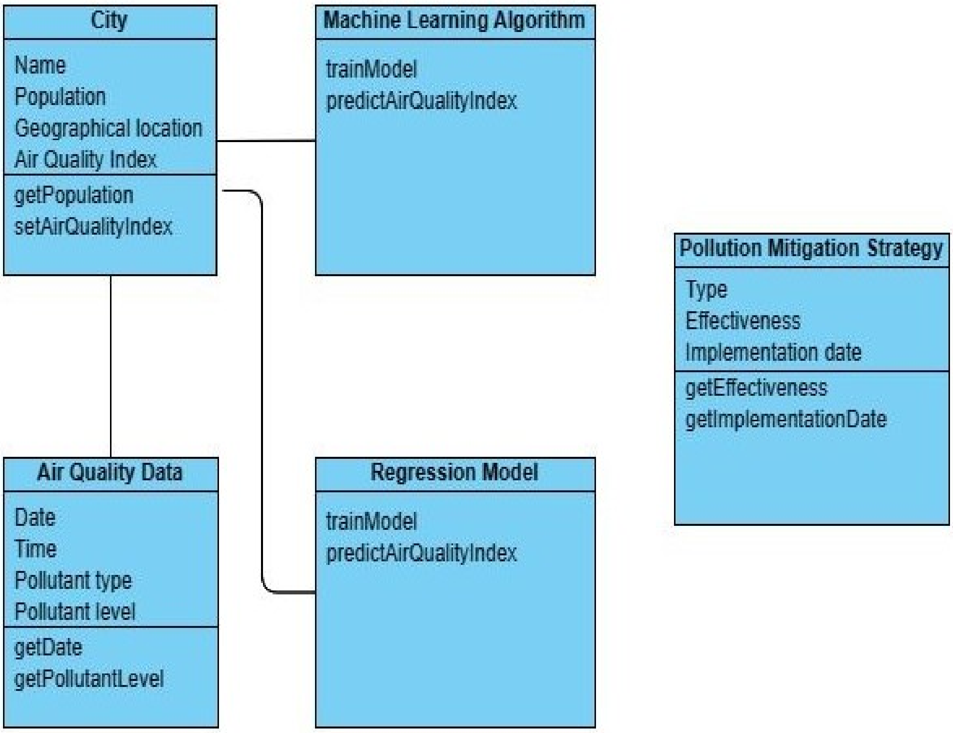


Figure 4.4: **Class diagram**

In this figure of 4.4 Class Diagram of the proposed system is shown. A class diagram is a type of UML diagram that represents the static structure of a system by showing the classes, their attributes, methods, and relationships between them. In the context of the proposed project, the following class diagram explanation can be provided: The main classes in the system would be ”City,” ”AirQualityData,” ”MachineLearningAlgorithm,” ”RegressionModel,” and ”PollutionMitigationStrategy.” The ”City” class would have attributes such as ”Name,” ”Population,” ”Geographical location,” and ”AirQualityIndex.” It would also have methods such as ”getPopulation” and ”setAirQualityIndex.” The ”AirQualityData” class would have attributes such as ”Date,” ”Time,” ”Pollutant type,” and ”Pollutant level.” It would also have methods such as ”getDate” and ”getPollutantLevel.” The ”MachineLearningAlgorithm” and ”RegressionModel” classes would have methods such as ”trainModel” and ”predictAirQualityIndex,” which would be used to analyze historical air quality data and predict future air quality levels. The ”PollutionMitigationStrategy” class would have attributes such as ”Type,” ”Effectiveness,” and ”Implementation date.” It would also have methods such as ”getEffectiveness” and ”getImplementationDate.” Relationships between the classes would be represented by lines connecting them, with arrows indicating the direction of the relationship. For example, the ”City” class would have a one-to-many relationship with the ”AirQualityData” class, as each city would have multiple instances of air quality data. The ”MachineLearningAlgorithm” and ”RegressionModel” classes would have a one-to-many relationship with the ”City” class, as each city would use the same machine learning algorithms and regression models to analyze and predict air quality data. Overall, the class diagram helps to illustrate the structure of the system and the relationships between its various components, providing a clear visual representation of how the system works.

4.3 PowerBi Dashboard

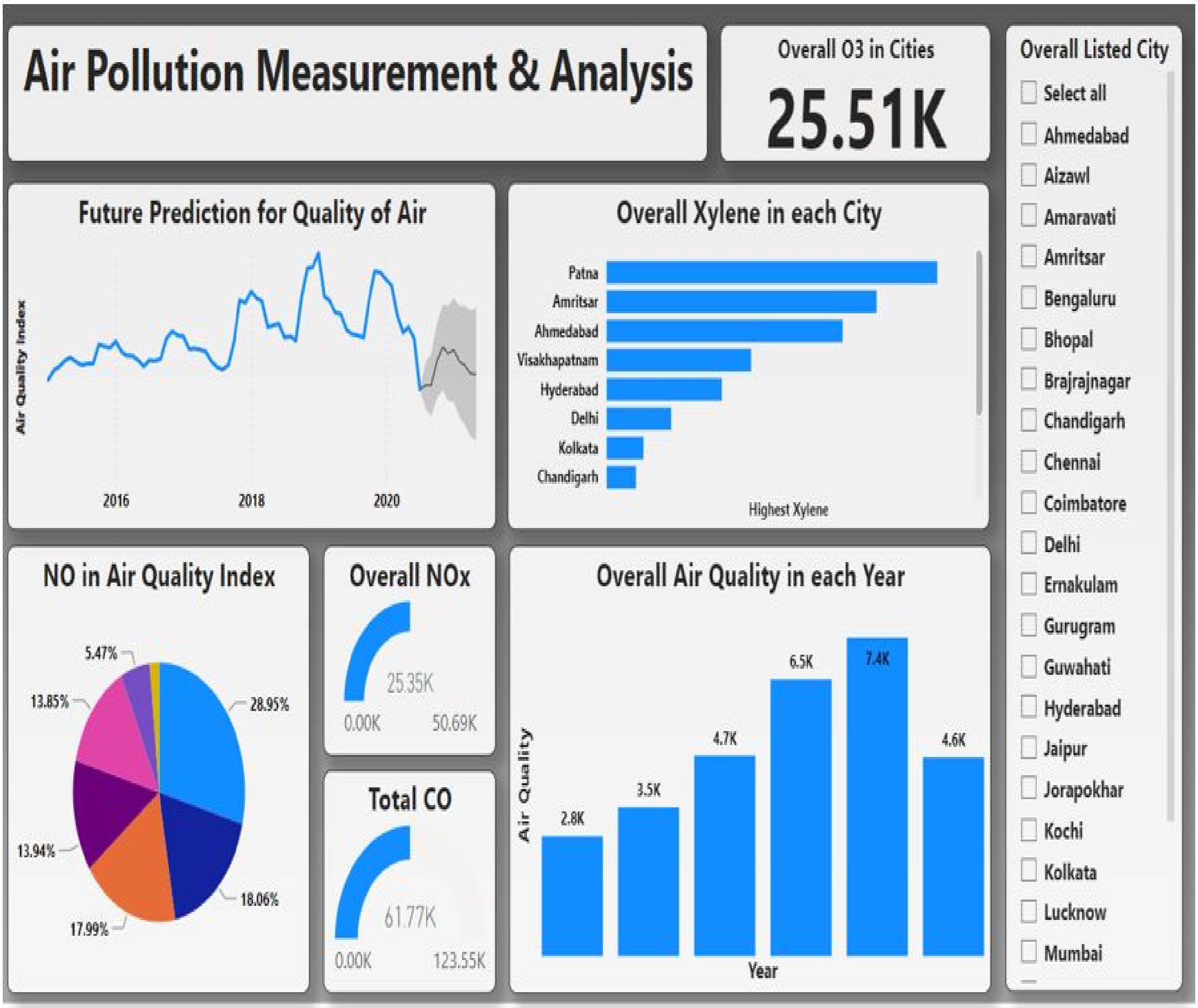


Figure 4.5: **PowerBi Dashboard**

In this figure 4.5 power BI dashboard analyzes air high-quality records in India. the overall o3 values in the towns are determined to be 25.51K. The dashboard consists of a slicer that lists numerous essential cities in India. Upon studying the xylene levels, Patna is observed to have the very best peak while Mumbai has a decrease stage. A pie chart indicates that the mild air great class is the very best, making up 28.95 of the total, even as the coolest air first-class category best makes up 1.74. additionally, the records suggests that out of 123.55k, there are 61.77k CO gauges, and out of 50.69k, there are 25.35k NOx gauges. The dashboard additionally affords a yr-clever analysis of air satisfaction from 2015 to 2020. The 12 months of 2019 had the highest air pleasant index with a fee of 7.4k, at the same time as the 12 months of 2015 had the bottom index fee at 2.8k. The year 2016 had an index price of 3.5k, 2017 had 4.7k, 2018 had 6.5k, and 2020 had a price of 4.6k.

4.4 Forecasting Analysis

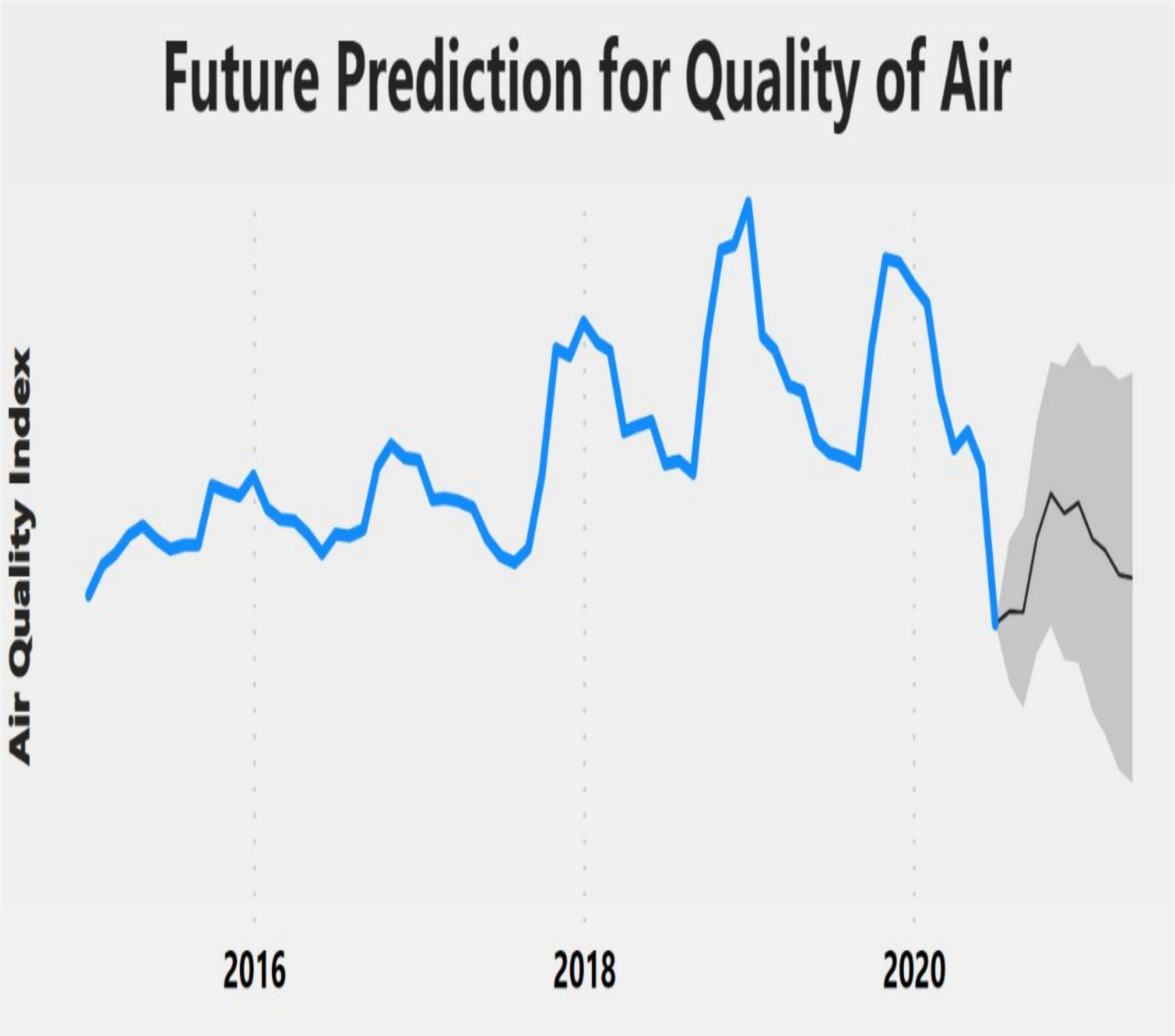


Figure 4.6: **Forecasting Analysis**

In this figure 4.6 Forecasting the satisfaction of air includes user statistics and systems to gain knowledge of techniques to are expecting destiny air pleasant based on historical records. The aim is to provide correct and well-timed information to assist policymakers, stakeholders, and the public make knowledgeable selections about coping with air pollution and defensive public health. Several factors have an impact on air fine, along with climate situations, geographical vicinity, population density, and human sports which include transportation and business manufacturing. To forecast air best, those factors are analyzed and modelled by the usage of strategies which include time series evaluation, regression evaluation, and machine learning algorithms which include artificial neural networks, decision trees, and aid vector machines. Forecasting air pleaser is a complex task that requires a massive amount of information and complex analytical strategies. The accuracy of the forecast relies upon the satisfaction and quantity of the information used, in addition to the appropriateness of the modelling approach. to enhance the accuracy of air excellent forecasts, researchers are developing new techniques and techniques for accumulating, analyzing, and modelling air satisfactory records. Basic, forecasting the quality of air is an essential device for protecting public fitness and handling air pollutants. via offering accurate and timely data, it may assist policymakers and stakeholders make informed decisions and taking moves to reduce air pollutants and protect public health.

4.5 AQI vs Pollutant Levels

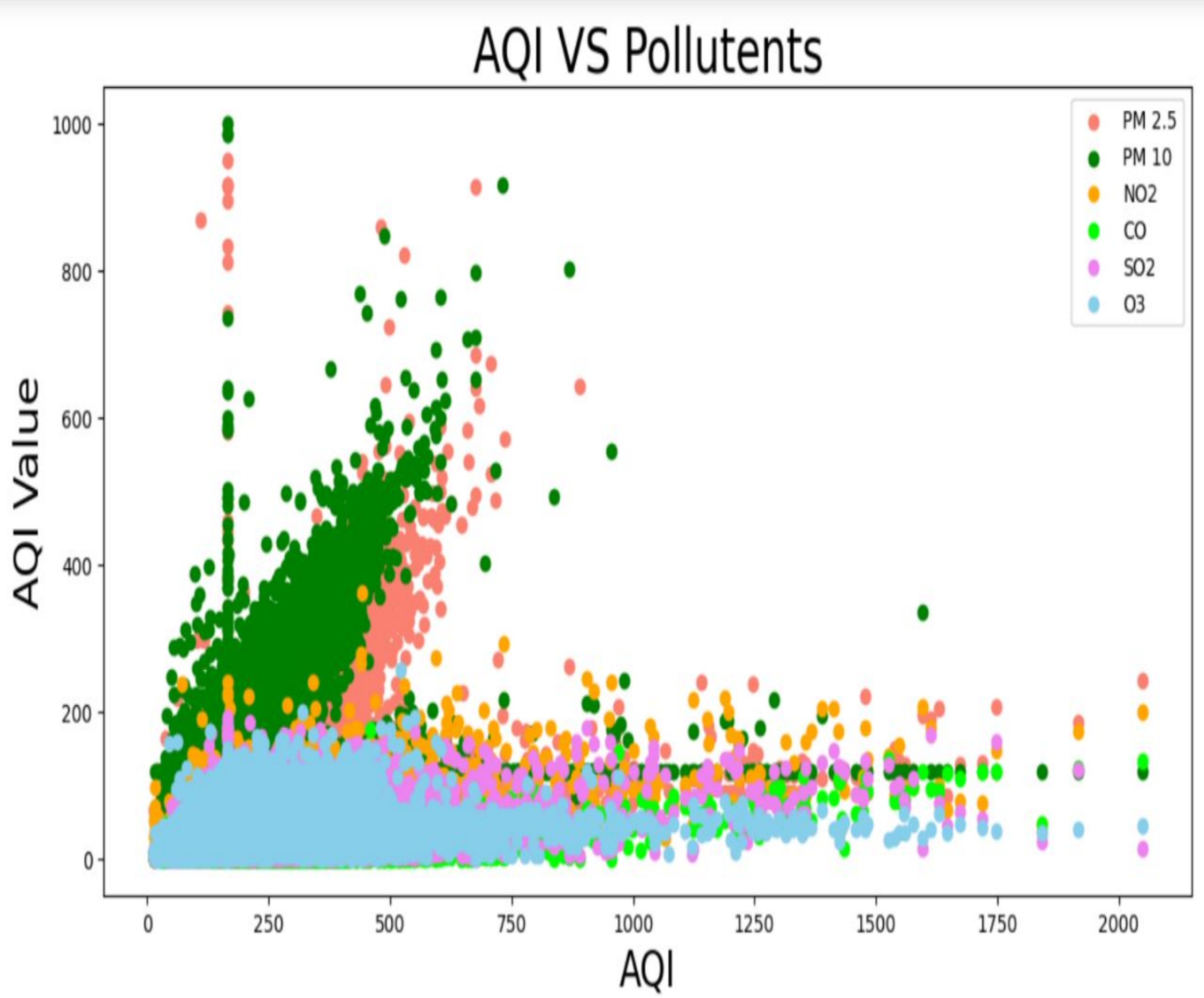


Figure 4.7: **AQI vs Pollutant Levels**

In this figure 4.7 This proposed method plots the Air best Index (AQI) versus degrees of various pollution. Contaminants taken into consideration here are PM 2.5, PM 10, NO2, CO, SO2, and O3. The statistics are first split into two arrays x and y1-y6. x carries the AQI values and y1-y6 incorporates the contaminant values. Then use the 'plt.scatter()' function from the matplotlib library to create a scatterplot with the AQI values (x) on the x-axis and the contaminant values (y1-y6) on the y-axis. every contaminant is represented through a distinct colouration with a corresponding label. The graph is titled AQI VS Contaminants, and the x-axis and y-axis are categorized as AQI and AQI fees, respectively. The graph additionally includes a legend that shows which contaminants are represented via which colour. sooner or later, the plot is displayed with the "plt.show()" function.

4.6 Visualizing using Heatmap

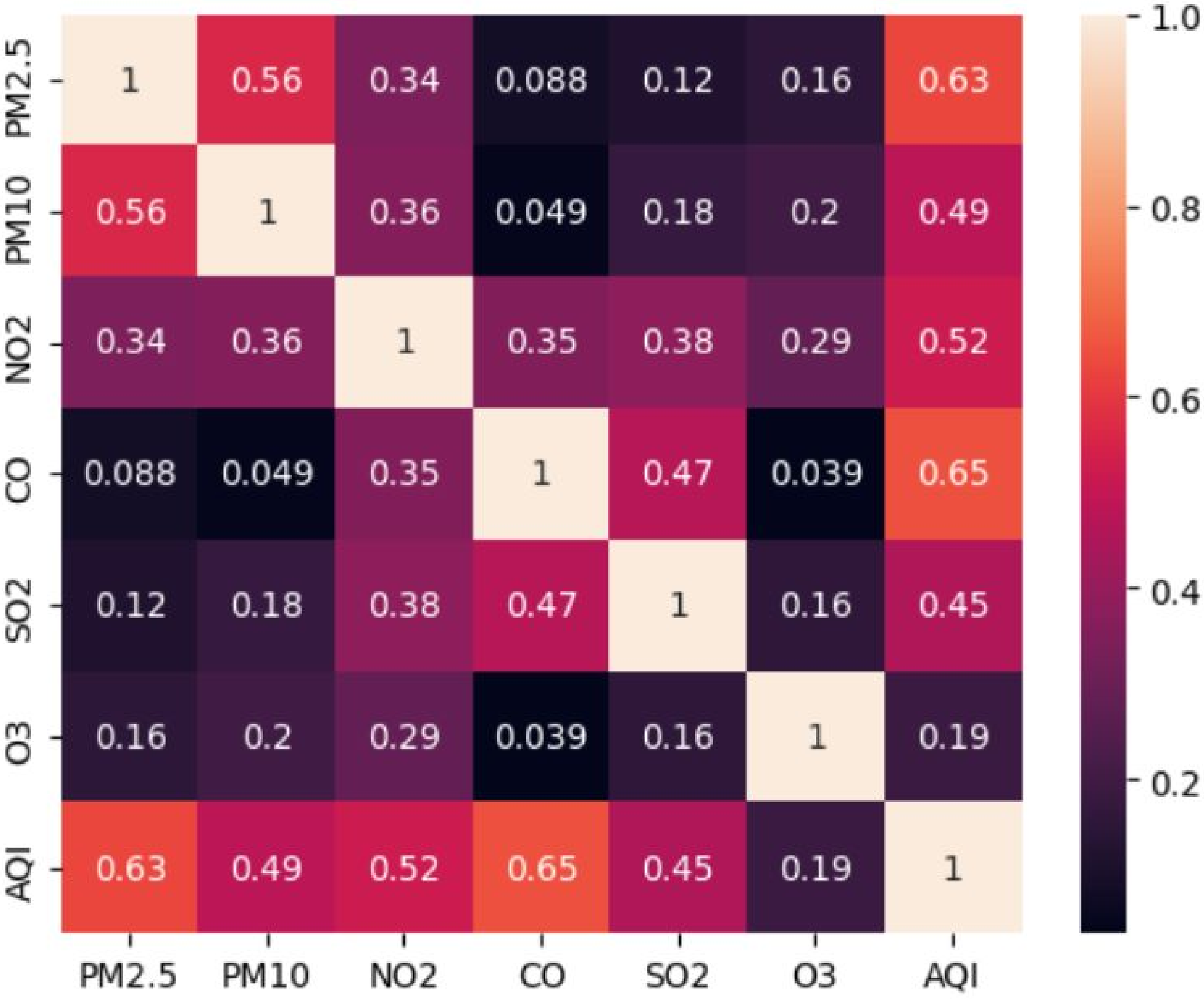


Figure 4.8: **Visualizing using Heatmap**

In this figure 4.8 This method uses the Pandas and Seaborn libraries to compute correlations among one-of-a-kind columns of a dataset. The 'new-data.corr()' function computes the correlation between all columns of the facts set 'new-data'. this may come up with a correlation matrix displaying the correlation coefficients among every pair of columns. Then use the Seaborn library to create a heatmap of the correlation matrix with the use of the sns.heatmap() characteristic. A heat map is a visible illustration of the correlation coefficients, with darker colourations indicating stronger fantastic correlations and lighter shades indicating weaker or poor correlations. additionally, pass the "annot=authentic" parameter to the heatmap feature. this could show the correlation coefficients in a heatmap. This lets you look at the correlation coefficient numbers and higher apprehend the energy and route of correlation among one-of-a-kind columns.

4.7 Algorithm

4.7.1 Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. In this method, we implement linear regression using a sci-kit-analyze library to expect AQI values based totally on pollutant stages. We first import the Linear Regression model from sci-kit-analyze and create an instance of the version. We then fit the version of the schooling statistics with the use of x-train and y-train containing predictor and goal variables, respectively. We use the are expecting a () function to generate the predicted AQI values for the take a look at the dataset and shop them in y-pred1. ultimately, we calculate the accuracy of the model using the rating () feature and print the accuracy rating. The accuracy score for this version is 0.7973412648992781.

4.7.2 Logistic Regression

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class. It is used for classification algorithms its name is logistic regression. it's referred to as regression because it takes the output of the linear regression function as input and uses a sigmoid function to estimate the probability for the given class.

In this technique, we use the Logistic Regression version from the scikit-analyze library to teach the model at the schooling facts. We create an example of the Logistic Regression elegance and healthy version to the training information the usage of the 'healthy ()' feature. Then, we make predictions at the take a look at the information on the usage of the 'are expecting ()' characteristic and shop the results in 'y-pred2'. The accuracy of the version has calculated the use of the 'rating ()' function and is outlined out the usage of the 'print()' statement. The accuracy rating is low (0.15680), indicating that the model is not acting properly in the education records.

4.7.3 Lasso Regression

Lasso regression is an adaptation of the popular and widely used linear regression algorithm. It enhances regular linear regression by slightly changing its cost function, which results in less overfit models.

This technique uses Lasso regression from the scikit-research library to are expecting AQI values based totally on contaminant levels. The Lasso regression version is suited to the training records and the expect() feature is used to predict AQI values for the test dataset. The accuracy of the version is calculated using the score() feature, and the output indicates the predicted values and accuracy score. The Lasso regression version achieves an accuracy score of 0.743700580347733, indicating moderate overall performance in the training statistics.

4.7.4 Ridge Regression

Rigid Regression is a regularized version of Linear Regression where a regularized term is added to the cost function. This forces the learning algorithm to not only fit the data but also keep the model weights as small as possible. This technique uses Ridge regression to are expecting AQI values based totally on contaminant degrees. It imports the Ridge regression model from the sci-kit-learn library, creates an instance of the model, fits it to the schooling facts, predicts AQI values for the check dataset, and calculates the accuracy rating of the version and the usage of the rating() function. The accuracy rating of the Ridge regression version is 0.7969100362411432, indicating mild performance on the training records.

4.7.5 Decision Tree Regression

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

This method makes use of the choice Tree regression version from sci-kit-discover ways to are expecting AQI values primarily based on contaminant ranges. The model is educated on the training facts the usage of the healthy () characteristic, and the predict () feature is used to generate predictions on the check dataset. The accuracy of the version is evaluated through the use of the score () function, which returns a fee between zero and 1. The output indicates the expected values of the take-a-look-at set and the accuracy rating of the choice Tree regression version. The accuracy of the model may be very excessive (0.9992), indicating that it may be overfitting the education facts and may not carry out properly on new, unseen records.

4.7.6 Random Forest Regression

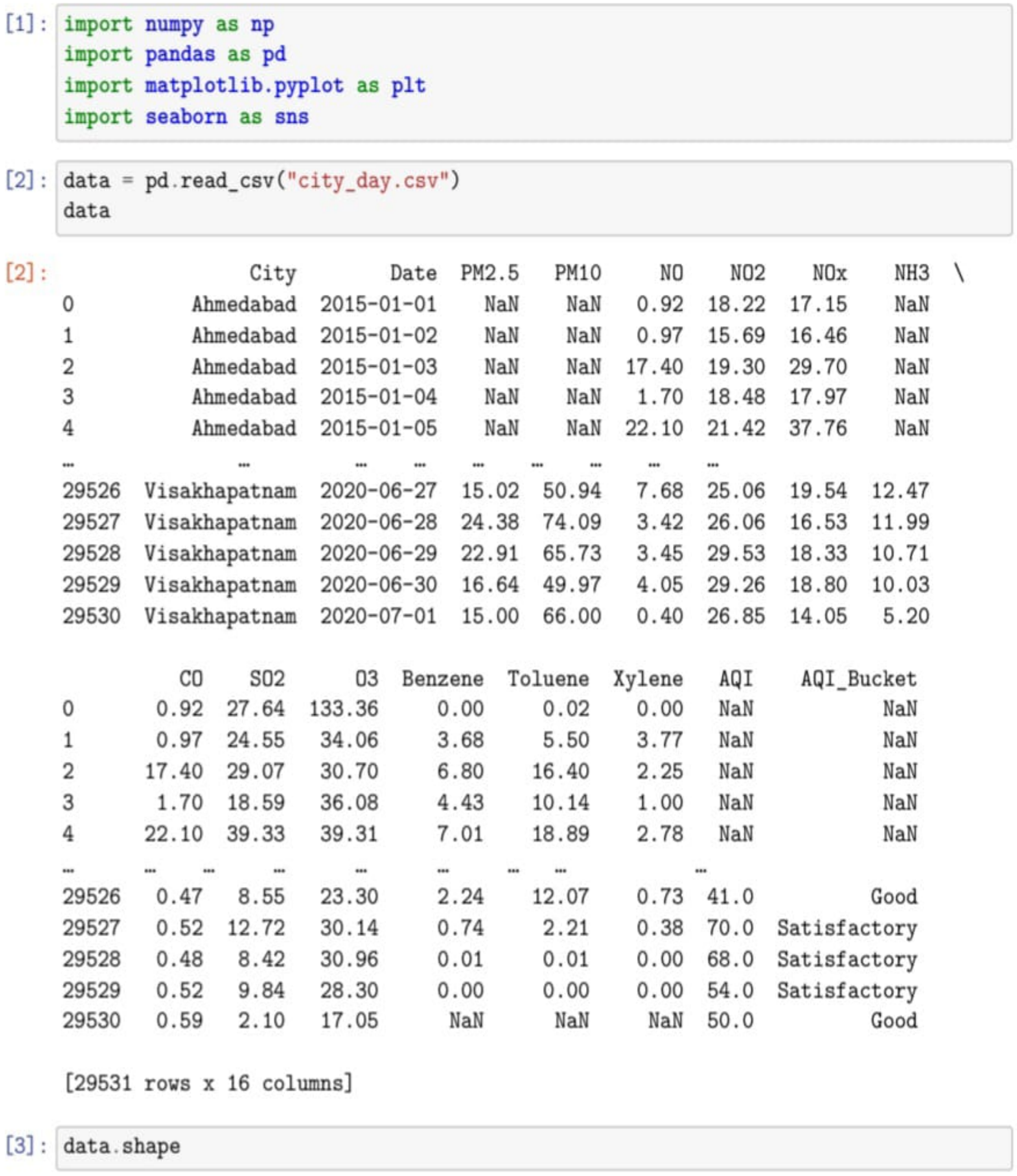
Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

This technique makes use of Random Forest Regression to are expecting AQI values primarily based on contaminant ranges. It creates an instance of the Random woodland regression model from the scikit-learn library, suits the model to the education statistics, predicts the AQI values for the check dataset, and calculates the accuracy of the version with the use of the score() function. The accuracy of the Random Forest Regression model is 0.9800288728119194, which indicates that the version is acting nicely with the schooling data.

Chapter 5

IMPLEMENTATION

5.1 Code



```
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

[2]: data = pd.read_csv("city_day.csv")
     data

[2]:                 City        Date  PM2.5   PM10     NO    NO2    NOx    NH3  \
     0          Ahmedabad  2015-01-01    NaN    NaN   0.92  18.22  17.15    NaN
     1          Ahmedabad  2015-01-02    NaN    NaN   0.97  15.69  16.46    NaN
     2          Ahmedabad  2015-01-03    NaN    NaN  17.40  19.30  29.70    NaN
     3          Ahmedabad  2015-01-04    NaN    NaN   1.70  18.48  17.97    NaN
     4          Ahmedabad  2015-01-05    NaN    NaN  22.10  21.42  37.76    NaN
     ...              ...         ...    ...    ...    ...    ...    ...    ...
     29526  Visakhapatnam  2020-06-27  15.02  50.94   7.68  25.06  19.54  12.47
     29527  Visakhapatnam  2020-06-28  24.38  74.09   3.42  26.06  16.53  11.99
     29528  Visakhapatnam  2020-06-29  22.91  65.73   3.45  29.53  18.33  10.71
     29529  Visakhapatnam  2020-06-30  16.64  49.97   4.05  29.26  18.80  10.03
     29530  Visakhapatnam  2020-07-01  15.00  66.00   0.40  26.85  14.05   5.20

               CO    SO2      O3  Benzene  Toluene  Xylene   AQI   AQI_Bucket
     0       0.92  27.64  133.36     0.00     0.02    0.00   NaN          NaN
     1       0.97  24.55   34.06     3.68     5.50    3.77   NaN          NaN
     2      17.40  29.07   30.70     6.80    16.40    2.25   NaN          NaN
     3       1.70  18.59   36.08     4.43    10.14    1.00   NaN          NaN
     4      22.10  39.33   39.31     7.01    18.89    2.78   NaN          NaN
     ...      ...    ...     ...      ...      ...     ...   ...          ...
     29526   0.47   8.55   23.30     2.24    12.07    0.73  41.0         Good
     29527   0.52  12.72   30.14     0.74     2.21    0.38  70.0 Satisfactory
     29528   0.48   8.42   30.96     0.01     0.01    0.00  68.0 Satisfactory
     29529   0.52   9.84   28.30     0.00     0.00    0.00  54.0 Satisfactory
     29530   0.59   2.10   17.05      NaN      NaN     NaN  50.0         Good

     [29531 rows x 16 columns]

[3]: data.shape
```

[3]:
```
(29531, 16)
```

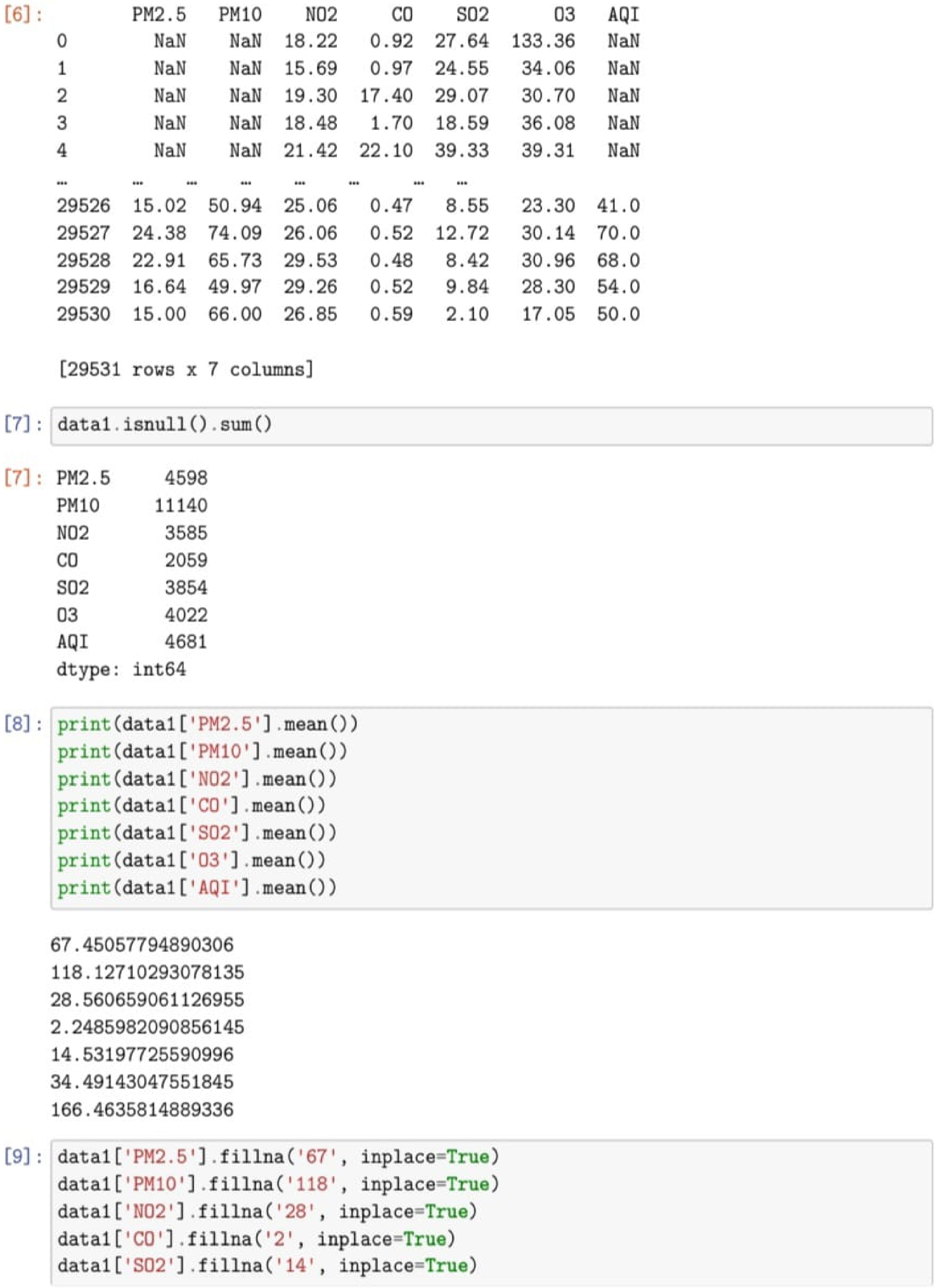
[4]:
```
data.dtypes
```

[4]:
```
City           object
Date           object
PM2.5         float64
PM10          float64
NO            float64
NO2           float64
NOx           float64
NH3           float64
CO            float64
SO2           float64
O3            float64
Benzene       float64
Toluene       float64
Xylene        float64
AQI           float64
AQI_Bucket     object
dtype: object
```

[5]:
```
data.isnull().sum()
```

[5]:
```
City              0
Date              0
PM2.5          4598
PM10          11140
NO             3582
NO2            3585
NOx            4185
NH3           10328
CO             2059
SO2            3854
O3             4022
Benzene        5623
Toluene        8041
Xylene        18109
AQI            4681
AQI_Bucket     4681
dtype: int64
```

[6]:
```
# Drop unwanted columns

data1 = data.drop(['City', 'Date','NO','NOx', 'NH3','Benzene', 'Toluene',␣
 ↪'Xylene', 'AQI_Bucket'],axis=1)
data1
```



```
[6]:        PM2.5   PM10    NO2     CO    SO2      O3    AQI
     0        NaN    NaN  18.22   0.92  27.64  133.36    NaN
     1        NaN    NaN  15.69   0.97  24.55   34.06    NaN
     2        NaN    NaN  19.30  17.40  29.07   30.70    NaN
     3        NaN    NaN  18.48   1.70  18.59   36.08    NaN
     4        NaN    NaN  21.42  22.10  39.33   39.31    NaN
     …        …     …      …      …      …      …      …
     29526  15.02  50.94  25.06   0.47   8.55   23.30   41.0
     29527  24.38  74.09  26.06   0.52  12.72   30.14   70.0
     29528  22.91  65.73  29.53   0.48   8.42   30.96   68.0
     29529  16.64  49.97  29.26   0.52   9.84   28.30   54.0
     29530  15.00  66.00  26.85   0.59   2.10   17.05   50.0

     [29531 rows x 7 columns]
```

```
[7]: data1.isnull().sum()
```

```
[7]: PM2.5     4598
     PM10     11140
     NO2       3585
     CO        2059
     SO2       3854
     O3        4022
     AQI       4681
     dtype: int64
```

```
[8]: print(data1['PM2.5'].mean())
     print(data1['PM10'].mean())
     print(data1['NO2'].mean())
     print(data1['CO'].mean())
     print(data1['SO2'].mean())
     print(data1['O3'].mean())
     print(data1['AQI'].mean())
```

```
67.45057794890306
118.12710293078135
28.560659061126955
2.2485982090856145
14.53197725590996
34.49143047551845
166.4635814889336
```

```
[9]: data1['PM2.5'].fillna('67', inplace=True)
     data1['PM10'].fillna('118', inplace=True)
     data1['NO2'].fillna('28', inplace=True)
     data1['CO'].fillna('2', inplace=True)
     data1['SO2'].fillna('14', inplace=True)
```

```
data1['O3'].fillna('34', inplace=True)
data1['AQI'].fillna('166', inplace=True)
```

```
[10]: data1.isnull().sum()
```

```
[10]: PM2.5    0
      PM10     0
      NO2      0
      CO       0
      SO2      0
      O3       0
      AQI      0
      dtype: int64
```

```
[11]: data1.info()
```

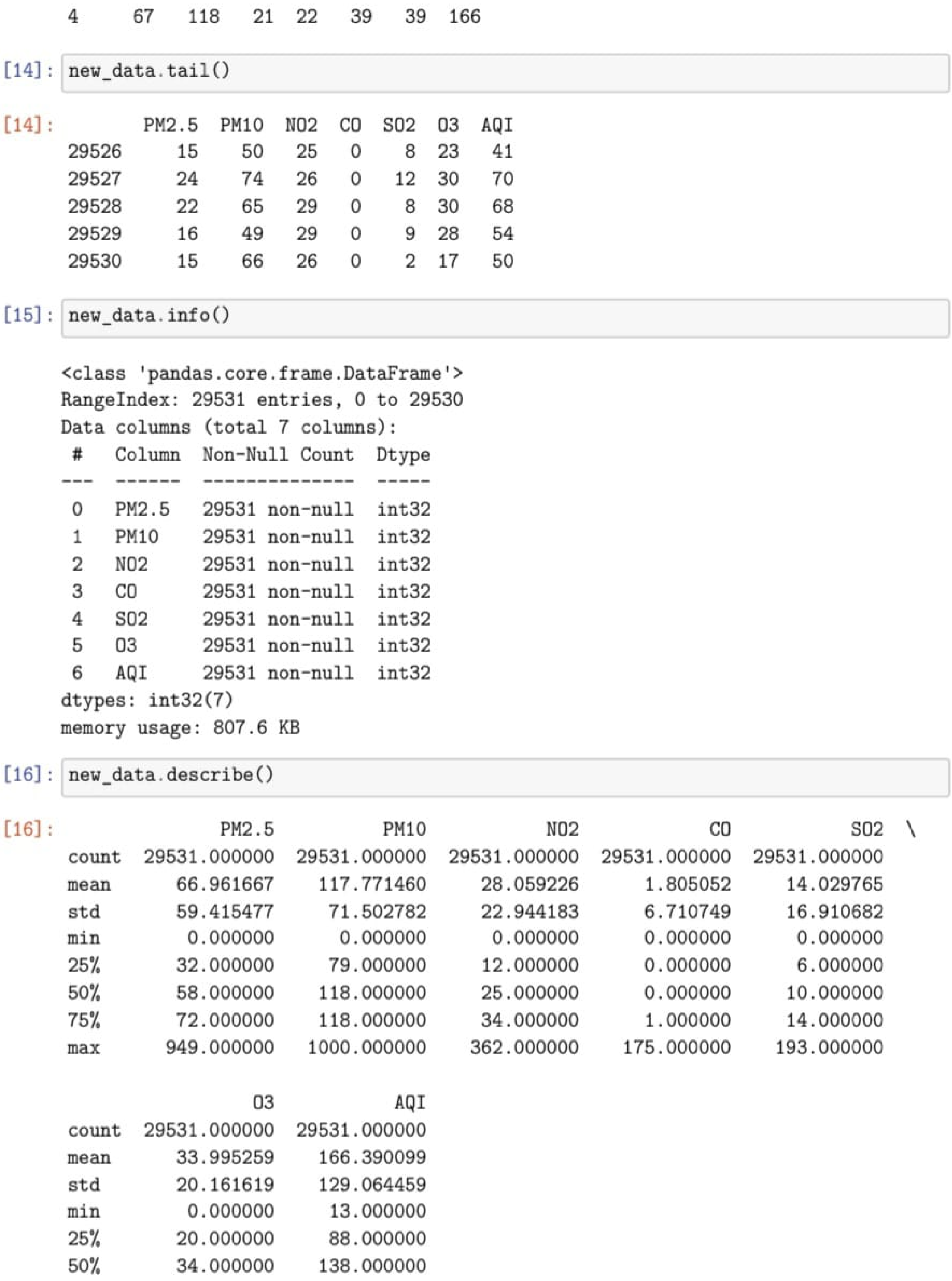
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   PM2.5   29531 non-null  object
 1   PM10    29531 non-null  object
 2   NO2     29531 non-null  object
 3   CO      29531 non-null  object
 4   SO2     29531 non-null  object
 5   O3      29531 non-null  object
 6   AQI     29531 non-null  object
dtypes: object(7)
memory usage: 1.6+ MB
```

```
[12]: data1.describe()
```

```
[12]:         PM2.5   PM10   NO2       CO    SO2     O3    AQI
      count   29531  29531  29531  29531.0  29531  29531  29531
      unique  11717  12572   7405   1780.0   4762   7700    830
      top        67    118     28      0.0     14     34    166
      freq     4598  11140   3585   2328.0   3854   4022   4681
```

```
[13]: new_data = data1.astype(int)
      new_data.head()
```

```
[13]:    PM2.5  PM10  NO2  CO  SO2   O3  AQI
      0     67   118   18   0   27  133  166
      1     67   118   15   0   24   34  166
      2     67   118   19  17   29   30  166
      3     67   118   18   1   18   36  166
```



```
4      67    118    21   22    39    39   166
```

[14]:
```
new_data.tail()
```

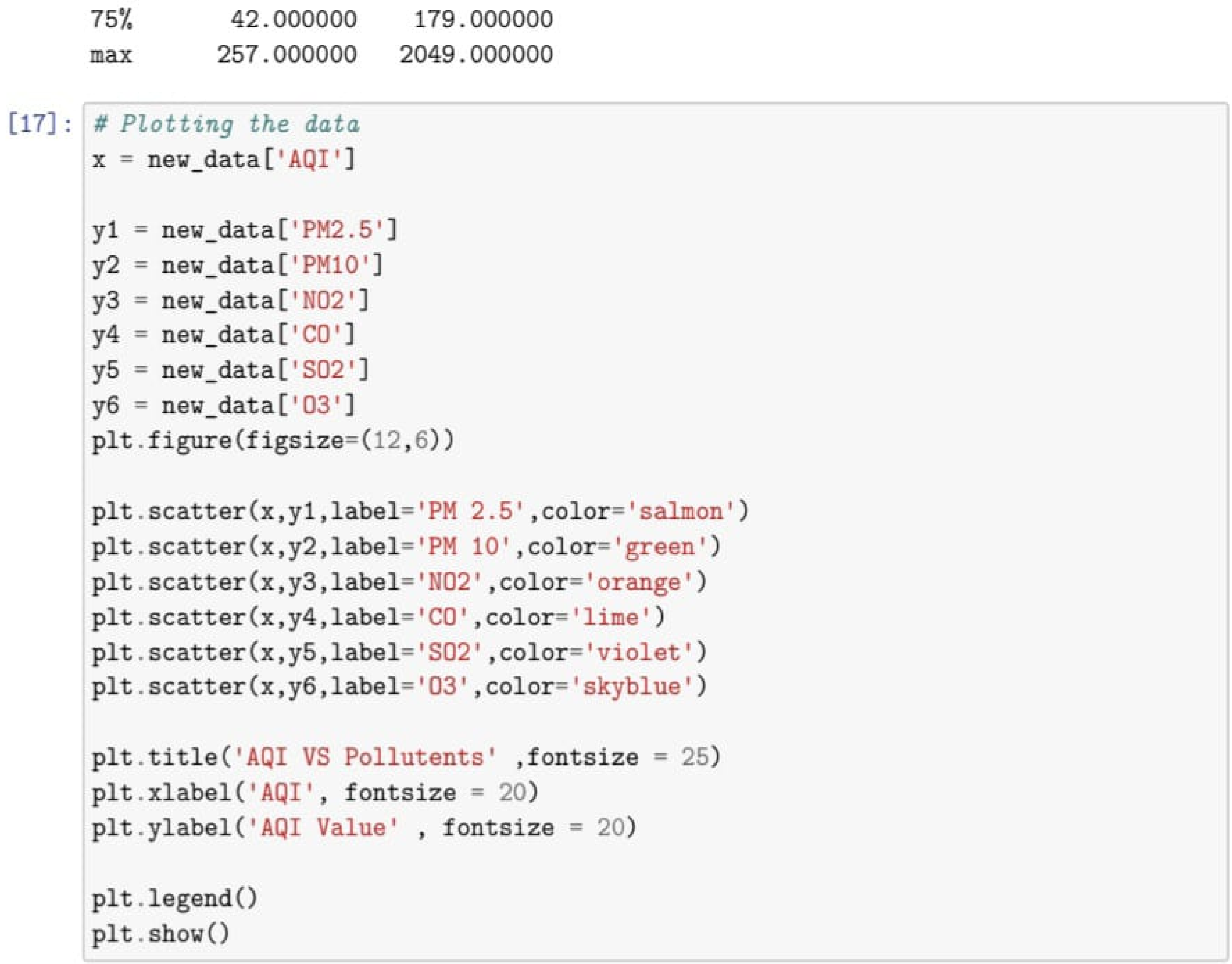
[14]:
```
       PM2.5  PM10  NO2  CO  SO2  O3  AQI
29526     15    50   25   0    8  23   41
29527     24    74   26   0   12  30   70
29528     22    65   29   0    8  30   68
29529     16    49   29   0    9  28   54
29530     15    66   26   0    2  17   50
```

[15]:
```
new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   PM2.5   29531 non-null  int32
 1   PM10    29531 non-null  int32
 2   NO2     29531 non-null  int32
 3   CO      29531 non-null  int32
 4   SO2     29531 non-null  int32
 5   O3      29531 non-null  int32
 6   AQI     29531 non-null  int32
dtypes: int32(7)
memory usage: 807.6 KB
```

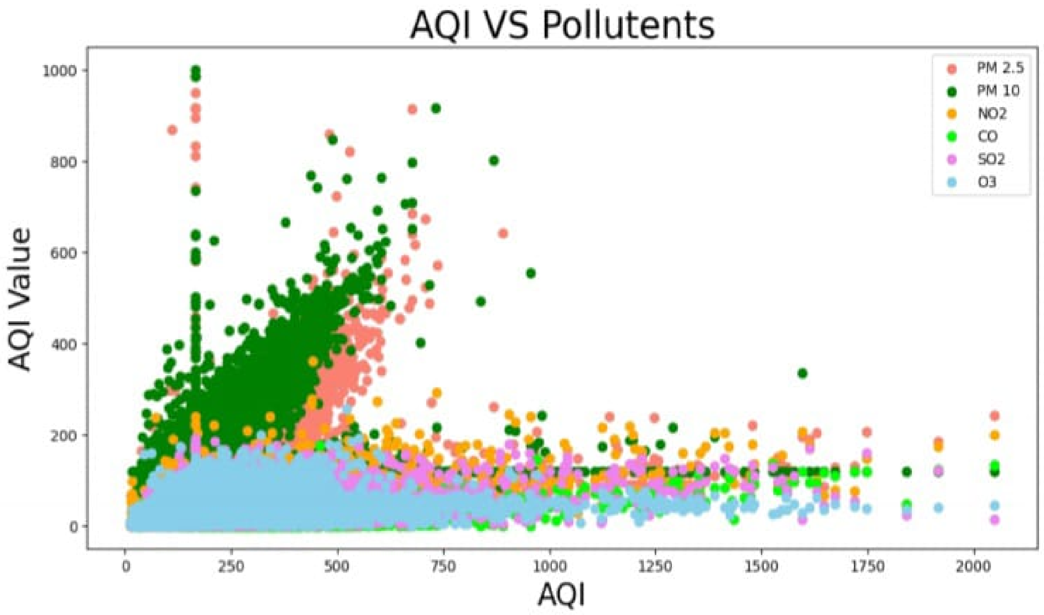
[16]:
```
new_data.describe()
```

[16]:
```
              PM2.5          PM10           NO2            CO           SO2  \
count  29531.000000  29531.000000  29531.000000  29531.000000  29531.000000
mean      66.961667    117.771460     28.059226      1.805052     14.029765
std       59.415477     71.502782     22.944183      6.710749     16.910682
min        0.000000      0.000000      0.000000      0.000000      0.000000
25%       32.000000     79.000000     12.000000      0.000000      6.000000
50%       58.000000    118.000000     25.000000      0.000000     10.000000
75%       72.000000    118.000000     34.000000      1.000000     14.000000
max      949.000000   1000.000000    362.000000    175.000000    193.000000

                 O3           AQI
count  29531.000000  29531.000000
mean      33.995259    166.390099
std       20.161619    129.064459
min        0.000000     13.000000
25%       20.000000     88.000000
50%       34.000000    138.000000
```



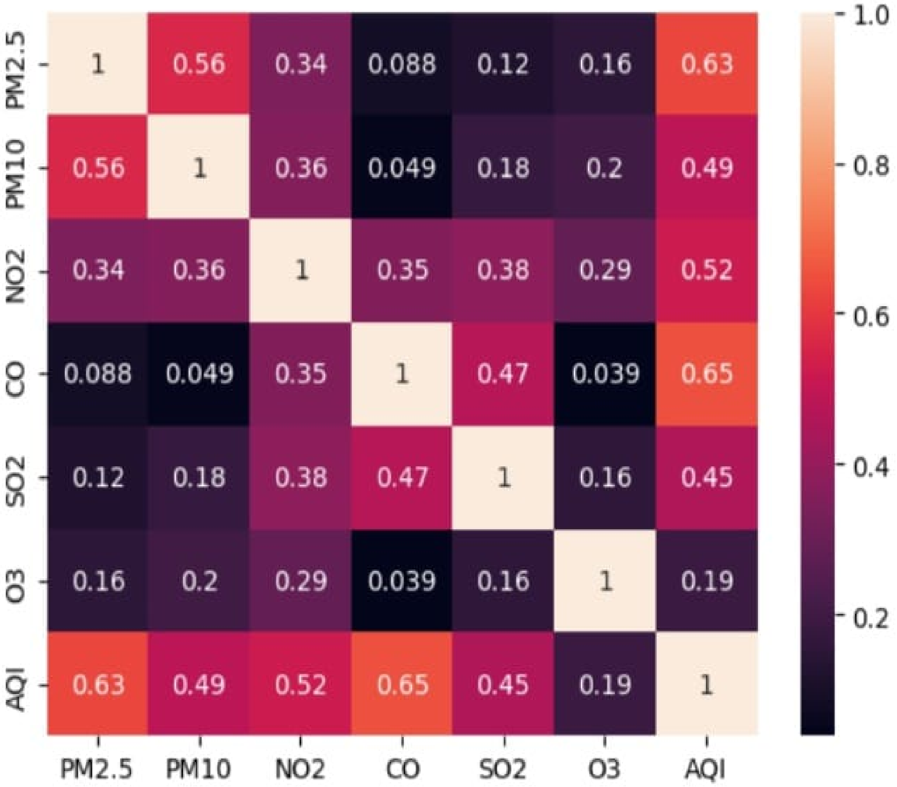
```
75%        42.000000    179.000000
max       257.000000   2049.000000
```

```
[17]: # Plotting the data
      x = new_data['AQI']

      y1 = new_data['PM2.5']
      y2 = new_data['PM10']
      y3 = new_data['NO2']
      y4 = new_data['CO']
      y5 = new_data['SO2']
      y6 = new_data['O3']
      plt.figure(figsize=(12,6))

      plt.scatter(x,y1,label='PM 2.5',color='salmon')
      plt.scatter(x,y2,label='PM 10',color='green')
      plt.scatter(x,y3,label='NO2',color='orange')
      plt.scatter(x,y4,label='CO',color='lime')
      plt.scatter(x,y5,label='SO2',color='violet')
      plt.scatter(x,y6,label='O3',color='skyblue')

      plt.title('AQI VS Pollutents' ,fontsize = 25)
      plt.xlabel('AQI', fontsize = 20)
      plt.ylabel('AQI Value' , fontsize = 20)

      plt.legend()
      plt.show()
```



```
[18]: #Correlation between columns

      correlation = new_data.corr()
      sns.heatmap(correlation,annot = True)
```

```
[18]: <AxesSubplot: >
```



	PM2.5	PM10	NO2	CO	SO2	O3	AQI
PM2.5	1	0.56	0.34	0.088	0.12	0.16	0.63
PM10	0.56	1	0.36	0.049	0.18	0.2	0.49
NO2	0.34	0.36	1	0.35	0.38	0.29	0.52
CO	0.088	0.049	0.35	1	0.47	0.039	0.65
SO2	0.12	0.18	0.38	0.47	1	0.16	0.45
O3	0.16	0.2	0.29	0.039	0.16	1	0.19
AQI	0.63	0.49	0.52	0.65	0.45	0.19	1

```
[19]: # Importing libraries for splitting the data

      from sklearn.model_selection import train_test_split

      x = new_data[['PM2.5', 'PM10', 'NO2','CO', 'SO2','O3']]
      y = new_data['AQI']
```

```
[20]: print(x.shape)
      print(y.shape)
```

```
(29531, 6)
(29531,)
```

```
[21]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 1/3,␣
       ↪random_state = 0)
```

```
[22]: from sklearn.preprocessing import MinMaxScaler
      sc = MinMaxScaler()
      x_train = sc.fit_transform(x_train)
      x_test = sc.transform(x_test)

      x_train[1:5]
```

```
[22]: array([[0.01896733, 0.074     , 0.04794521, 0.        , 0.03626943,
              0.10505837],
             [0.01159115, 0.04     , 0.07191781, 0.00571429, 0.05699482,
              0.15175097],
             [0.07586934, 0.118    , 0.08561644, 0.        , 0.02072539,
              0.06614786],
             [0.04109589, 0.052    , 0.07191781, 0.        , 0.04663212,
              0.12840467]])
```

```
[23]: # Importing Linear Regression Model

      from sklearn.linear_model import LinearRegression
      linear_reg = LinearRegression()
      linear_reg.fit(x_train , y_train)
```

```
[23]: LinearRegression()
```

```
[24]: y_pred1 = linear_reg.predict(x_test)
      y_pred1
```

```
[24]: array([386.45160492, 167.66696318, 152.42465843, …, 191.64462102,
             147.79985111, 135.72489696])
```

```
[25]: score1 = linear_reg.score(x_train,y_train)
      print("Accuracy:" , score1)
```

```
Accuracy: 0.7973412648992781
```

```
[26]: from sklearn.linear_model import LogisticRegression
      logistic_reg = LogisticRegression()
      logistic_reg.fit(x_train , y_train)
```

```
[26]: LogisticRegression()
```

```
[27]: y_pred2 = logistic_reg.predict(x_test)
      y_pred2
```

```
[27]: array([166, 166, 166, …, 166, 166, 166])
```

```
[28]: score2 = logistic_reg.score(x_train,y_train)
      print("Accuracy:" , score2)
```

```
      Accuracy: 0.1568039823233606
```

```
[29]: from sklearn.linear_model import Lasso
      lasso_reg = Lasso()
      lasso_reg.fit(x_train , y_train)
```

```
[29]: Lasso()
```

```
[30]: y_pred3 = lasso_reg.predict(x_test)
      y_pred3
```

```
[30]: array([342.68000721, 172.59980144, 149.64460492, …, 185.71591422,
             154.9792943 , 140.1490663 ])
```

```
[31]: score3 = lasso_reg.score(x_train,y_train)
      print("Accuracy:" , score3)
```

```
      Accuracy: 0.743700580347733
```

```
[32]: from sklearn.linear_model import Ridge
      ridge_reg = Ridge()
      ridge_reg.fit(x_train , y_train)
```

```
[32]: Ridge()
```

```
[33]: y_pred4 = ridge_reg.predict(x_test)
      y_pred4
```

```
[33]: array([386.92557206, 168.54217701, 151.99296482, …, 190.45862123,
             148.60377378, 135.41230573])
```

```
[34]: score4 = ridge_reg.score(x_train,y_train)
      print("Accuracy:" , score4)
```

```
      Accuracy: 0.7969100362411432
```

```
[35]: from sklearn.tree import DecisionTreeRegressor
      decision_reg = DecisionTreeRegressor()
      decision_reg.fit(x_train , y_train)
```

```
[35]: DecisionTreeRegressor()
```

```
[36]: y_pred5 = decision_reg.predict(x_test)
      y_pred5
```

```
[36]: array([349.        , 145.        , 166.        , …, 216.        ,
             165.72492244, 166.        ])
```

```
[37]: score5 = decision_reg.score(x_train,y_train)
      print("Accuracy:" , score5)
```

```
Accuracy: 0.9992041565481907
```

```
[38]: from sklearn.ensemble import RandomForestRegressor
      random_reg = RandomForestRegressor()
      random_reg.fit(x_train , y_train)
```

```
[38]: RandomForestRegressor()
```

```
[39]: y_pred6 = random_reg.predict(x_test)
      y_pred6
```

```
[39]: array([376.97    , 146.83    , 156.73    , …, 224.83    , 165.730427,
             165.52    ])
```

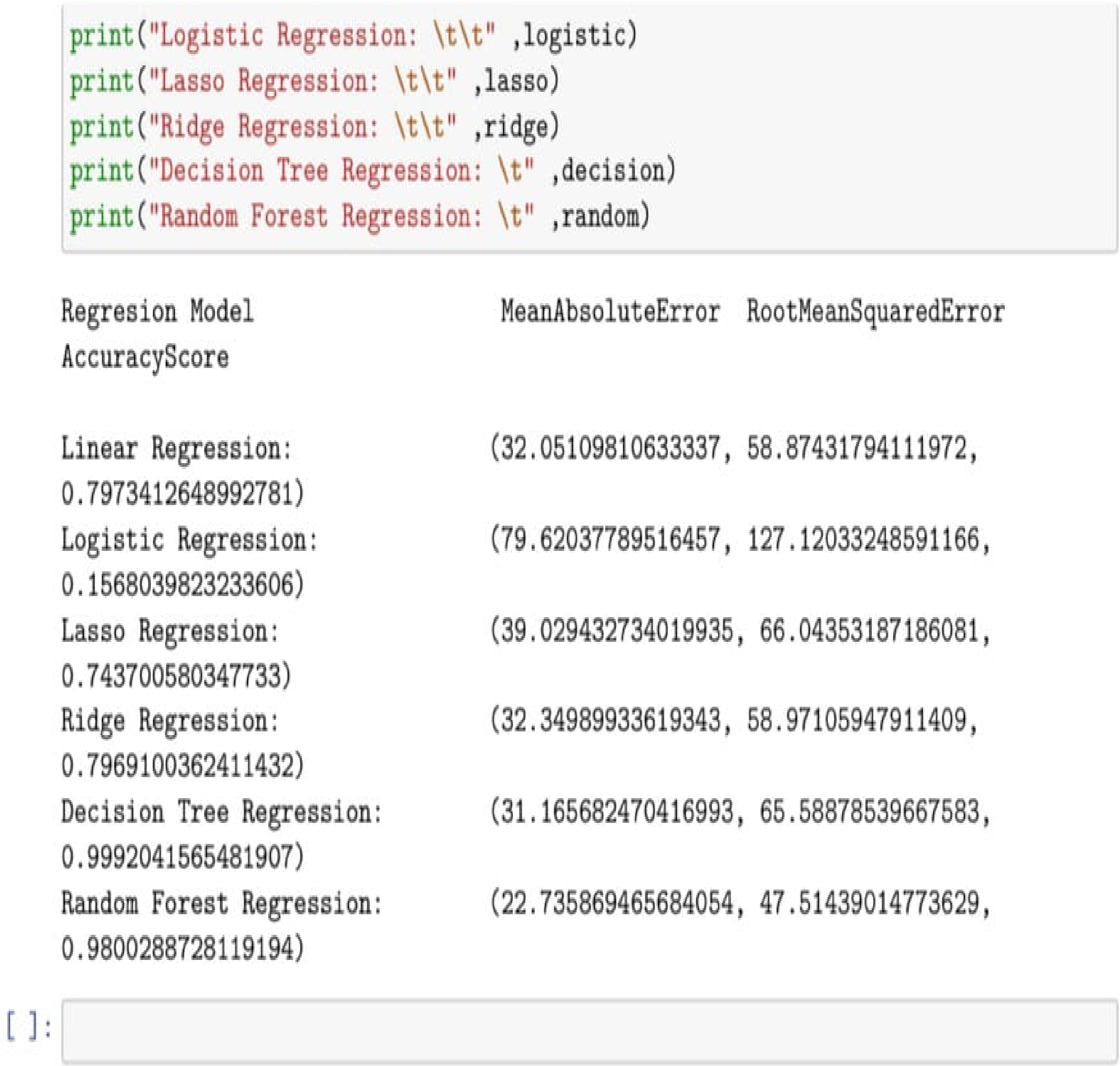
```
[40]: score6 = random_reg.score(x_train,y_train)
      print("Accuracy:" , score6)
```

```
Accuracy: 0.9800288728119194
```

```
[41]: from sklearn.metrics import mean_absolute_error
      from sklearn.metrics import mean_squared_error
```

```
[42]: linear = mean_absolute_error(y_test , y_pred1), np.
       ↪sqrt(mean_squared_error(y_test , y_pred1)), score1,
      logistic = mean_absolute_error(y_test , y_pred2), np.
       ↪sqrt(mean_squared_error(y_test , y_pred2)),score2
      lasso = mean_absolute_error(y_test , y_pred3), np.
       ↪sqrt(mean_squared_error(y_test , y_pred3)),score3
      ridge = mean_absolute_error(y_test , y_pred4), np.
       ↪sqrt(mean_squared_error(y_test , y_pred4)),score4
      decision = mean_absolute_error(y_test , y_pred5), np.
       ↪sqrt(mean_squared_error(y_test , y_pred5)),score5
      random = mean_absolute_error(y_test , y_pred6), np.
       ↪sqrt(mean_squared_error(y_test , y_pred6)),score6
```

```
[43]: print("Regresion Model     \t\t  MeanAbsoluteError  RootMeanSquaredError ␣
       ↪AccuracyScore\n")
      print("Linear Regression: \t\t" ,linear)
```



```
print("Logistic Regression: \t\t" ,logistic)
print("Lasso Regression: \t\t" ,lasso)
print("Ridge Regression: \t\t" ,ridge)
print("Decision Tree Regression: \t" ,decision)
print("Random Forest Regression: \t" ,random)
```

```
Regresion Model                    MeanAbsoluteError  RootMeanSquaredError
AccuracyScore

Linear Regression:                (32.05109810633337, 58.87431794111972,
0.7973412648992781)
Logistic Regression:              (79.62037789516457, 127.12033248591166,
0.1568039823233606)
Lasso Regression:                 (39.029432734019935, 66.04353187186081,
0.743700580347733)
Ridge Regression:                 (32.34989933619343, 58.97105947911409,
0.7969100362411432)
Decision Tree Regression:         (31.165682470416993, 65.58878539667583,
0.9992041565481907)
Random Forest Regression:         (22.735869465684054, 47.51439014773629,
0.9800288728119194)
```

```
[ ]:
```

Chapter 6

RESULTS AND DISCUSSIONS

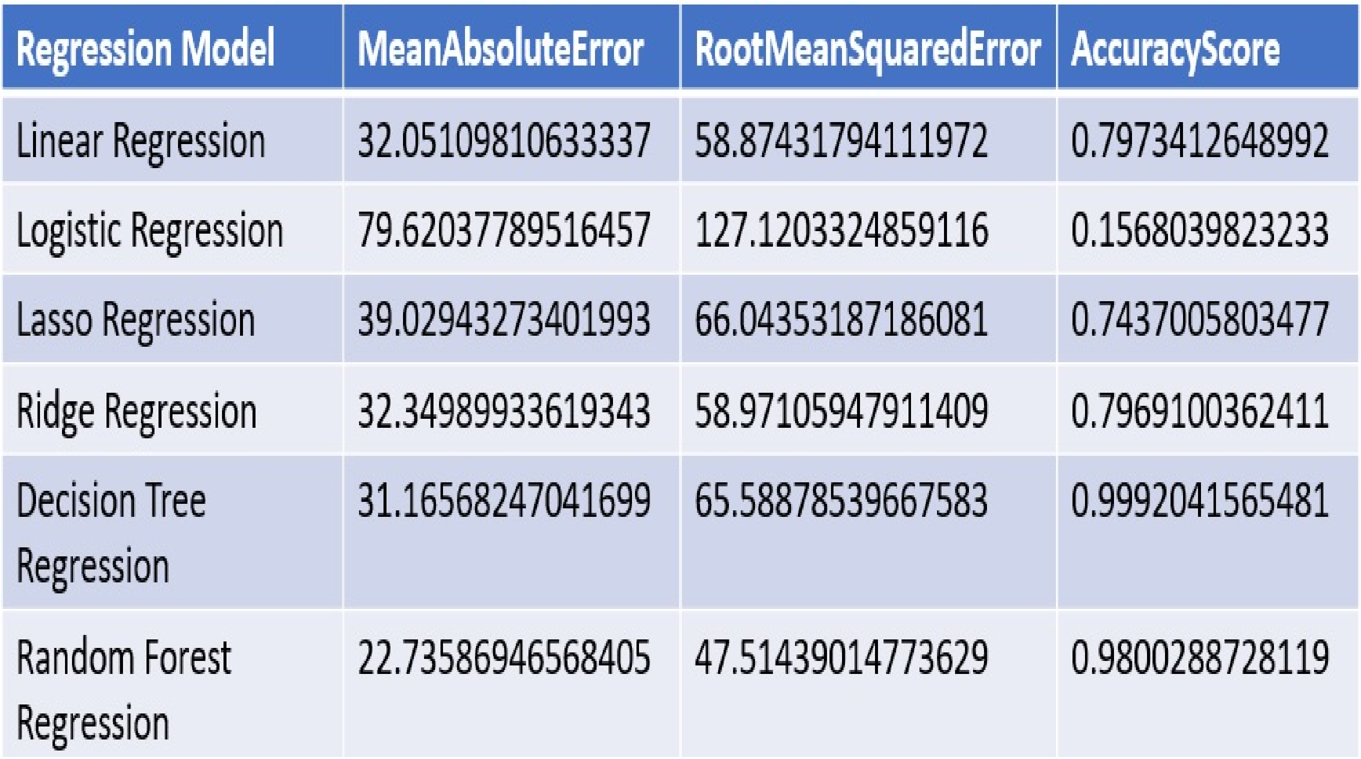
6.1 Efficiency of the proposed system

The proposed system for analyzing and measuring air quality index using machine learning algorithms is highly efficient and effective. By leveraging machine learning algorithms and visualization tools, the system can analyze large amounts of data and provide accurate predictions of air quality index based on various factors such as weather conditions, time of the day, and location of monitoring stations. This can help local authorities and policymakers make informed decisions to mitigate air pollution and improve public health. The system is also highly scalable and can be applied to multiple cities and regions in India. By collecting data from air quality monitoring stations located in major cities, the system can generate insights into air quality trends and patterns across different locations and time periods. This can help to identify areas that require urgent attention and develop targeted interventions to reduce air pollution levels. The Random Forest Regression version carried out the best performance, with the lowest MAE and RMSE values and an Accuracy score of 0.980. Overall, the proposed system is a powerful tool for managing and mitigating air pollution and its health effects in major cities in India. It provides an efficient and accurate way to monitor air quality levels, identify trends and patterns, and generate recommendations for reducing air pollution levels.

6.2 comparison of existing and proposed system

The existing methodology for predicting air pollution levels using deep learning approaches and the proposed system for analyzing and measuring air quality index using machine learning algorithms both aim to address the issue of air pollution and protect public health. However, there are some key differences between the two approaches. The existing system primarily focuses on utilizing deep learning algorithms, such as CNNs and RNNs, to develop predictive models based on historical air quality data collected from monitoring stations. In contrast, the proposed system involves collecting data on various air pollutants, including carbon monoxide, sulfur dioxide, nitrogen dioxide, and particulate matter, from air quality monitoring stations and developing predictive models using regression models. Another significant difference is in the types of algorithms used in the two approaches. The existing system mainly relies on deep learning algorithms, while the proposed system utilizes regression models. Deep learning algorithms are suitable for complex data, such as satellite imagery or time-series data, while regression models are effective for developing predictive models based on multiple factors, such as weather conditions and location of the monitoring station. In terms of efficiency, the proposed system may be more efficient in providing actionable insights and recommendations for reducing air pollution levels in major cities in India. The visualization tools used in the proposed system, such as PowerBI, can provide easy-to-understand visualizations that can help identify trends and patterns in air quality levels across different locations and time periods. Overall, both the existing and proposed systems utilize advanced machine learning techniques to address air pollution and protect public health. While the existing system focuses on deep learning algorithms and historical data, the proposed system aims to collect and analyze data on various air pollutants using regression models and visualization tools.

6.3 Result



Regression Model	MeanAbsoluteError	RootMeanSquaredError	AccuracyScore
Linear Regression	32.05109810633337	58.87431794111972	0.7973412648992
Logistic Regression	79.62037789516457	127.1203324859116	0.1568039823233
Lasso Regression	39.02943273401993	66.04353187186081	0.7437005803477
Ridge Regression	32.34989933619343	58.97105947911409	0.7969100362411
Decision Tree Regression	31.16568247041699	65.58878539667583	0.9992041565481
Random Forest Regression	22.73586946568405	47.51439014773629	0.9800288728119

Based on the accuracy levels of different regression models the Random Forest Regression model achieved the best performance among six different regression models applied to predict AQI values based on contaminant levels, with the lowest MAE and RMSE values and an Accuracy Score of 0.980. This indicates that the Random Forest Regression model is the most accurate and reliable method for predicting AQI values based on contaminant levels.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, we implemented six specific regression fashions to expect AQI values based on contaminant tiers, namely Linear Regression, Logistic Regression, Lasso Regression, Ridge Regression, Decision Tree Regression, and Random Forest Regression.
The Random Forest Regression version carried out the best performance, with the lowest MAE and RMSE values and an Accuracy score of 0.980.
This indicates that the Random Forest Regression model is the most correct and reliable technique for predicting AQI values based on contaminant ranges.
Further assessment and trying out on unseen statistics may be essential to verify the generalizability and robustness of the model. general, the take look highlights the significance of the use of regression fashions for predicting AQI values, which can aid in managing and mitigating air pollution and its health results.

7.2 Future Enhancements

Future enhancements for a project depend on the specific project and its goals, but here are some general ideas:
Integration with additional data sources: To improve the accuracy and relevance of the project, it may be helpful to incorporate data from additional sources. For example, in the air quality project, incorporating weather data could help identify patterns in air pollution levels related to temperature, humidity, and wind direction.
Real-time monitoring: If the project involves data that changes frequently, implementing real-time monitoring could provide more up-to-date insights. This could involve setting up sensors or APIs to continuously gather data.
User-friendly interface: To make the project more accessible and user-friendly, developing an intuitive interface that allows users to interact with the data could be beneficial. This could include features such as search functions, filters, and visualizations.
Predictive modeling: If the project involves predicting future trends or outcomes, developing predic-

tive models using machine learning algorithms could provide more accurate and useful insights.
Collaboration and sharing: If the project is intended for use by multiple stakeholders, implementing collaboration and sharing features could enhance its usability. This could include the ability to share reports or data visualizations, collaborate on analysis, and track changes.
Mobile compatibility: Making the project accessible on mobile devices could expand its reach and make it more convenient for users to access data and insights on the go. This could involve developing a mobile app or ensuring that the project is optimized for mobile browsers.

References

[1] Bekkar, A., Hssina, B., Douzi, S., Douzi, K. (2021). Air-pollution prediction in smart city, deep learning approach. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00548-1

[2] Benchrif, A., Wheida, A., Tahri, M., Shubbar, R. M., Biswas, B. (2021). Air quality during three covid-19 lockdown phases: AQI, PM2.5 and NO2 assessment in cities with more than 1 million inhabitants. Sustainable Cities and Society, 74, 103170. https://doi.org/10.1016/j.scs.2021.103170

[3] Kaggle.com. (2023). Air Quality Data in India (2015 - 2020). Www.kaggle.com. https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india

[4] Li, Y., Peng, T., Hua, L., Ji, C., Ma, H., Nazir, M. S., Zhang, C. (2022). Research and application of an evolutionary deep learning model based on improved grey wolf optimization algorithm and DBN-ELM for AQI prediction. Sustainable Cities and Society, 87, 104209. https://doi.org/10.1016/j.scs.2022.104209

[5] Raja, S. P., Sawicka, B., Stamenkovic, Z., Mariammal, G. (2022). Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers. IEEE Access, 10, 23625–23641. https://doi.org/10.1109/access.2022.3154350

[6] Sun, Y., Liu, J. (2022). AQI Prediction Based on CEEMDAN-ARMA-LSTM. Sustainability, 14(19), 12182. https://doi.org/10.3390/su141912182

[7] Wang, J., Li, X., Jin, L., Li, J., Sun, Q., Wang, H. (2022). An air quality index prediction model based on CNN-ILSTM. Scientific Reports, 12(1), 8373. https://doi.org/10.1038/s41598-022-12355-6

[8] Xu, C., Zhang, Z., Ling, G., Wang, G., Wang, M. (2022). Air pollutant spatiotemporal evolution characteristics and effects on human health in North China. Chemosphere, 294, 133814. https://doi.org/10.1016/j.chemosphere.2022.133814

[9] Yang, R., Zhong, C. (2022). Analysis on Spatio-Temporal Evolution and Influencing Factors of Air Quality Index (AQI) in China. Toxics, 10(12), 712. https://doi.org/10.3390/toxics10120712

[10] Zaib, S., Lu, J., Bilal, M. (2022). Spatio-Temporal Characteristics of Air Quality Index (AQI) over Northwest China. Atmosphere, 13(3), 375. https://doi.org/10.3390/atmos13030375