AIR QUALITY INDEX ANALYZER USING ML

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted by

ERRABOINA SHESHIKUMAR	21UK1A6614
BODHIREDDY NIDHI REDDY	21UK1A6658
SAKINALA SAIPRIYA	21UK1A6629
ADABOINA SHIVA	21UK1A6646

Under the guidance of

Ms. R. SWATHI

Assistant professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI AND ML)

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL (T. S) – 506005 2021-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

VAAGDEVI ENGINEERING COLLEGE

WARANGAL

CERTIFICATE OF COMPLETION

MINI PROJECT

This is to certify that the mini project report entitled by "AIR QUALITY INDEX ANALYZER USING MACHINE LEARNING" is being submitted by E.SHESHIKUMAR(21UK1A6614) , B.NIDHI(21UK1A6658) , S.SAIPRIYA(21UK1A6629) , E.SHIVA(21UK1A6646) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in computer science and engineering(AI&ML) to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025

Project guide	Head of the Department
Ms. R. SWATHI	Dr SHARMILA REDDY
Assistant professor	Professor

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal,Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.K.SHARMILA**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide,Ms. R. SWATHI Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

ERRABOINA SHESHIKUMAR	21UK1A6614
ASAKINALA SAIPRIYA	21UK1A6629
BODHIREDDY NIDHIREDDY	21UK1A6658
EDABOINA SHIVA	21UK1A6646

ABSTRACT

The Air Quality Index (AQI) is a crucial tool for assessing air quality based on pollutant concentrations. It categorizes air quality into six buckets: Good, Satisfactory, Moderate, Poor, Very Poor, and Severe, with specific AQI value ranges, associated symptoms, diseases, and precautions. Good signifies excellent air quality with no notable symptoms,while Severe indicates life-threatening pollution with severe health risks.Understanding the AQI allows individuals to make informed decisions about outdoor activities and take appropriate precautions to protect their health. It serves as a valuable resource in promoting awareness and mitigating the adverse effects of air pollution on human well-being.

TABLE OF CONTENTS:-

1. INTRODUCTION

1.1. OVERVIEW

Clean and healthy air is a fundamental necessity for human well-being, and the Air Quality Index (AQI) serves as a critical tool for assessing and understanding the quality of the air we breathe. The AQI system categorizes air quality into six distinct buckets, each defined by specific AQI value ranges. These categories, ranging from "Good" to "Severe," provide essential information about the levels of pollutants in the atmosphere and their potential impacts on our health. This introduction explores the AQI classification, the associated symptoms and diseases, as well as the recommended precautions at each level, emphasizing the importance of AQI awareness in safeguarding public health. Understanding the AQI is vital in making informed decisions about outdoor activities and taking proactive measures to minimize the risks associated with varying air quality conditions. In this overview, we delve into the AQI classification, the associated symptoms and diseases at each level, and the recommended precautions to protect public health. Recognizing the significance of AQI awareness is essential in making informed decisions about outdoor activities and implementing preventive measures to mitigate health risks associated with fluctuating air quality conditions. It is a fundamental aspect of safeguarding the well-being of individuals and communities in an increasingly polluted world.

1.2. PURPOSE

The Air Quality Index (AQI) serves several critical purposes in assessing and communicating air quality, with a primary focus on safeguarding public health and the environment. In detail, its purposes include:

1. **Informing the Public**: The AQI is designed to provide the general public with easily understandable information about air quality. It informs individuals about the safety of outdoor activities and helps them make informed decisions to protect their health.

2. **Health Protection**: One of its primary purposes is to protect public health. The AQI categorizes air quality into different levels, each associated with specific health risks. This enables individuals, particularly those with respiratory conditions, to take precautions based on the current air quality conditions.

3. **Government Regulation**: The AQI is used by regulatory agencies and governments to set air quality standards and policies. It helps in identifying areas with poor air quality and implementing measures to improve it, such as emission controls and pollution reduction strategies.

4. **Environmental Monitoring**: The AQI also plays a role in monitoring the impact of air pollution on the environment. It helps track changes in air quality over time and assess the effectiveness of pollution control measures.

5. **Economic Impact**: Understanding air quality is essential for evaluating the economic impact of pollution on industries, healthcare costs, and productivity. It can guide policy decisions that affect economic development.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

 The existing problem of air quality degradation is a multifaceted global challenge that spans health, environmental, and economic domains. Poor air quality, characterized by elevated levels of pollutants, takes a devastating toll on public health, contributing to a wide array of respiratory and cardiovascular diseases while being a leading cause of premature mortality.

 Moreover, air pollution wreaks havoc on ecosystems, causing harm to both flora and fauna, and contributes to environmental issues like acid rain and smog. Economically, it translates into substantial costs related to healthcare expenses, lost workdays, and decreased labor productivity. Additionally, air pollution plays a significant role in climate change, with greenhouse gas emissions intensifying global warming. Vulnerable communities, often located near pollution sources, bear a

disproportionate burden.

 Inadequate regulations, limited public awareness, and the variability of air quality further compound the issue. Addressing these problems necessitates stringent regulations, the transition to cleaner energy sources, robust public education, and international cooperation to combat the health, environmental, and economic consequences associated with poor air quality.

2.2 PROPOSED SOLLUTION

 Our innovative proposed solution leverages advanced predictive modeling to anticipate and communicate Air Quality Index (AQI) bucket categories accurately and efficiently. By forecasting AQI buckets, we enable proactive dissemination of information regarding air quality, accompanying health risks, and recommended precautions.

This solution integrates technology, data analytic s, and public health awareness to create a comprehensive system for managing air quality issues.

1. Advanced Predictive Modeling: We employ sophisticated predictive models that take into account various environmental factors, historical data, and real time monitoring to forecast AQI buckets. This ensures timely and reliable information for the public.

2. Customized Health Information: For each AQI category, our system

provides a tailored list of potential diseases and recommended precautions. This empowers individuals to take proactive steps to protect their health based on the fore castes air quality conditions.

3. User-Friendly Interfaces: Our solution offers user-friendly interfaces, making it easy for the public to access and understand AQI predictions and associated health information.

4. Community Engagement: We actively engage with communities to raise awareness about the AQI forecasting system, ensuring that the public is well informed and prepared to take necessary precautions.

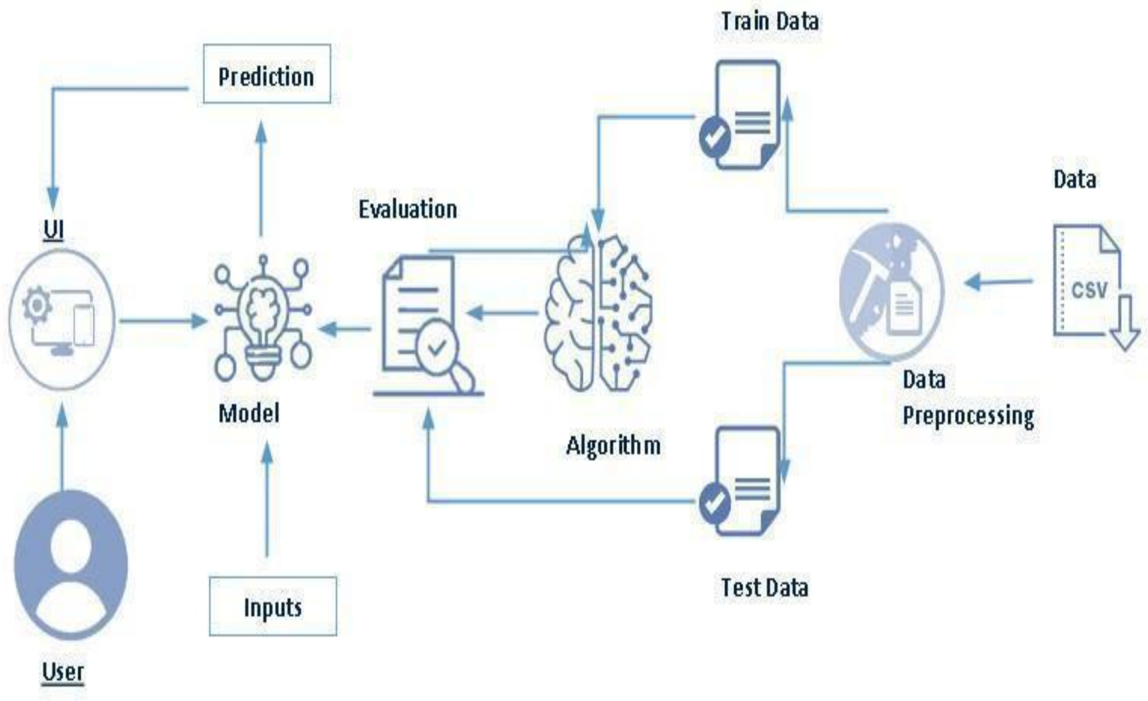
5. Educational Initiatives: Our solution includes educational initiatives to raise awareness about the importance of air quality and the impact of air pollution on public health, empowering individuals to make informed choices

 By accurately predicting AQI bucket categories and providing associated health information, our solution empowers individuals and communities to make informed decisions, protect their health, and reduce the adverse health effects of air pollution. It is a comprehensive approach to proactively address air quality issues and create healthier environments for all.

 By encompassing these elements, our proposed solution creates a comprehensive and adaptable framework for AQI prediction and management, with a strong emphasis on accuracy, user-friendliness, and community engagement.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

 Jupyter Notebook: Jupyter Notebook will serve as the development and execution environment for your predictive modeling, data preprocessing, and model training tasks. It provides an interactive, web-based environment with access to Python libraries and hardware acceleration.

 Dataset (CSV File): The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.

 Data Preprocessing Tools: Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.

 Feature Selection/Drop: Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.

 Model Training Tools: Machine learning libraries such as Scikit-learn,

TensorFlow, or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the AQI prediction task.

 Model Performance Evaluation: After model training, performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict AQI categories based on historical data using the R^2 score.

 UI Based on Flask Environment: Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view AQI predictions, health information, and recommended precautions.

 Jupyter Notebook will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selection will optimize the model. Finally, model performance evaluation will confirm the system's predictive capabilities, allowing users to rely on the AQI predictions and associated health information.

4.EXPERIMENTAL INVESTIGATION

In this project, we have used Air Quality Dataset. This dataset is a csv file consisting of labelled data and having the following columns-

1. **City**: Location or urban area for which air quality data is recorded.

2. **Date**: The specific date on which air quality measurements were taken.

3. **PM2.5**: Particulate Matter with a diameter of 2.5 micrometers, a key air pollutant.

4. **PM10**: Particulate Matter with a diameter of 10 micrometers, another significant air pollutant.

5. **NO**: Nitric Oxide, a gaseous air pollutant.

6. **NO2**: Nitrogen Dioxide, a toxic gas often related to combustion processes.

7. **NOx**: Nitrogen Oxides, a group of nitrogen-containing air pollutants.

8. **NH3**: Ammonia, a compound that can contribute to air pollution.

9. **CO**: Carbon Monoxide, a colorless, odorless gas produced by incomplete combustion.

10. **SO2**: Sulfur Dioxide, a toxic gas often linked to industrial processes.

11. **O3**: Ozone, a secondary pollutant with varying health impacts.

12. **Benzene**: A volatile organic compound that can be harmful when inhaled.

13. **Toluene**: Another volatile organic compound commonly found in urban air.

14. **Xylene**: A group of volatile organic compounds, often associated with vehicle emissions.

15. **AQI**: Air Quality Index, a numerical value indicating overall air quality.

16. **AQI_Bucket**: The qualitative classification of air quality based on the AQI value.

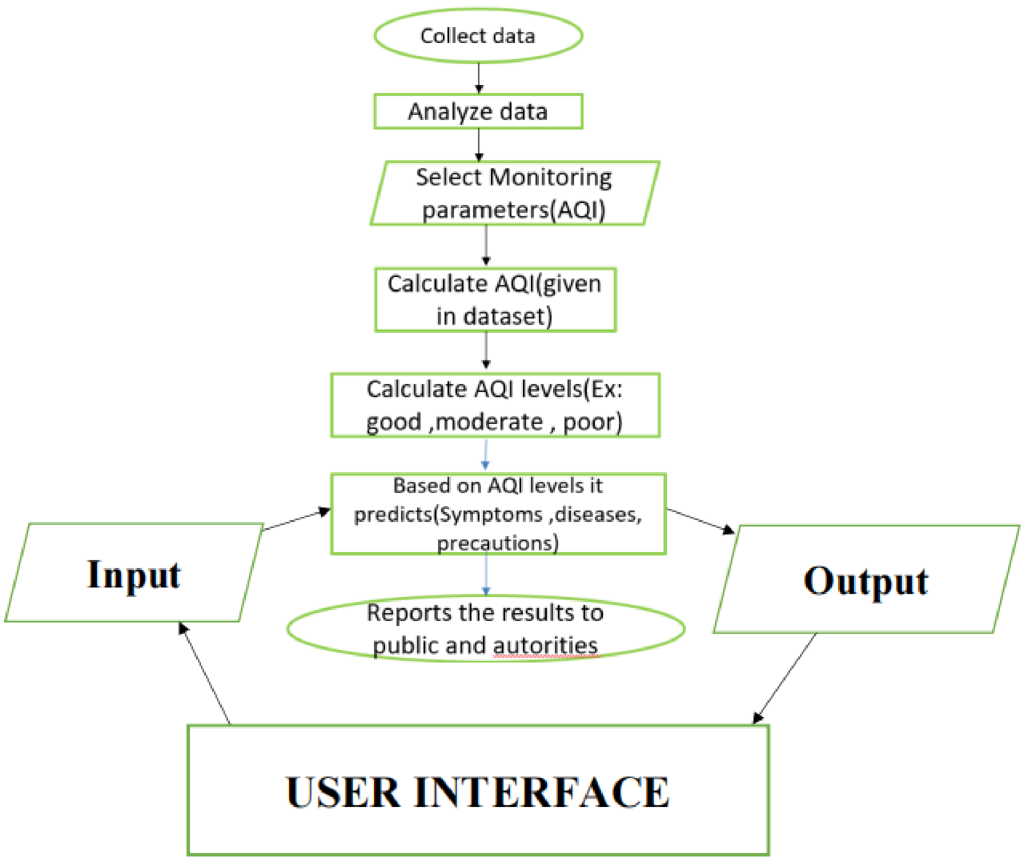
For the dataset we selected, it consists of more than the columns we want to predict it .

So, we have chosen the feature drop it contains the columns that we are going to predict

the AQI value.

 Feature drop means it drops the columns that we don’t want in our dataset.

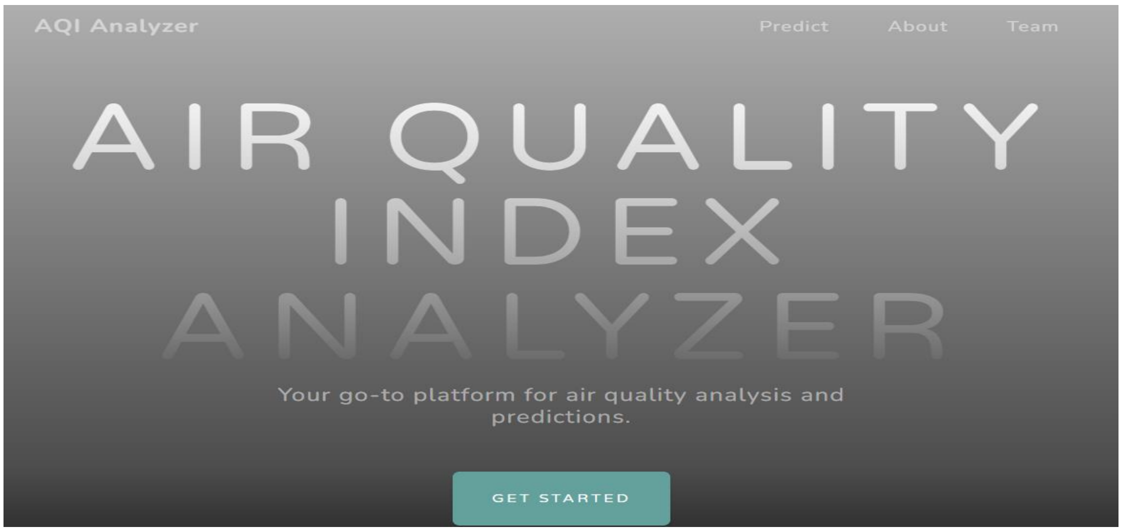
5. FLOWCHART



Collect data
Analyze data
Select Monitoring parameters(AQI)
Calculate AQI(given in dataset)
Calculate AQI levels(Ex: good ,moderate , poor)
Based on AQI levels it predicts(Symptoms ,diseases, precautions)
Input
Output
Reports the results to public and autorities
USER INTERFACE

6. RESULT

ABOUT PAGE



AQI Analyzer

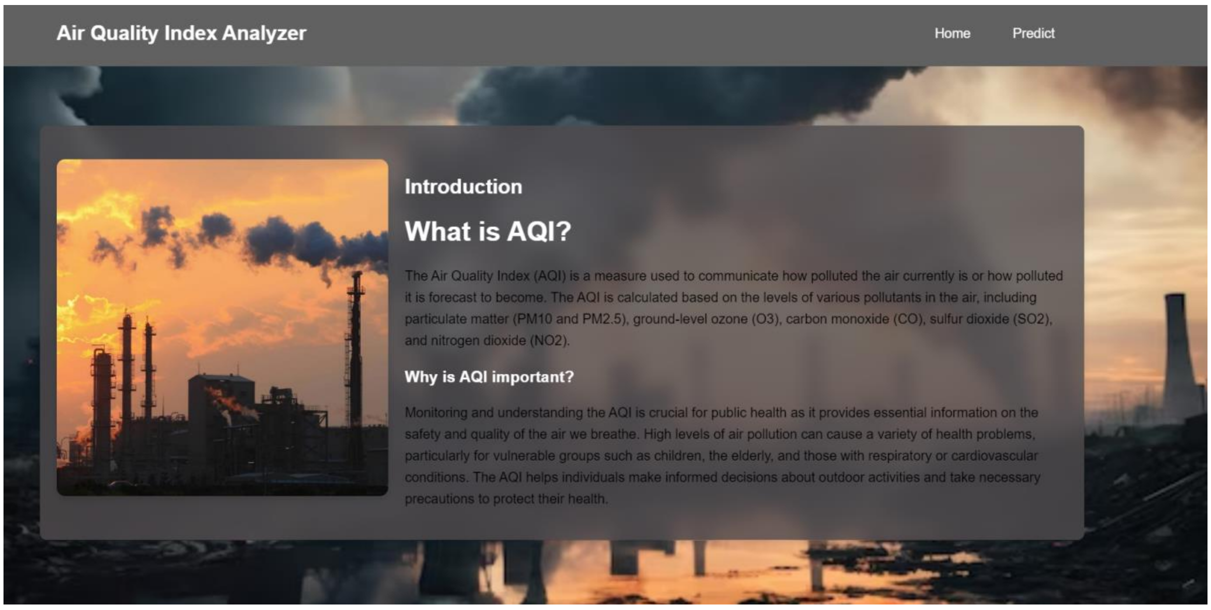
Predict About Team

AIR QUALITY INDEX ANALYZER

Your go-to platform for air quality analysis and predictions.

GET STARTED

HOME PAGE



Air Quality Index Analyzer

Home Predict

Introduction

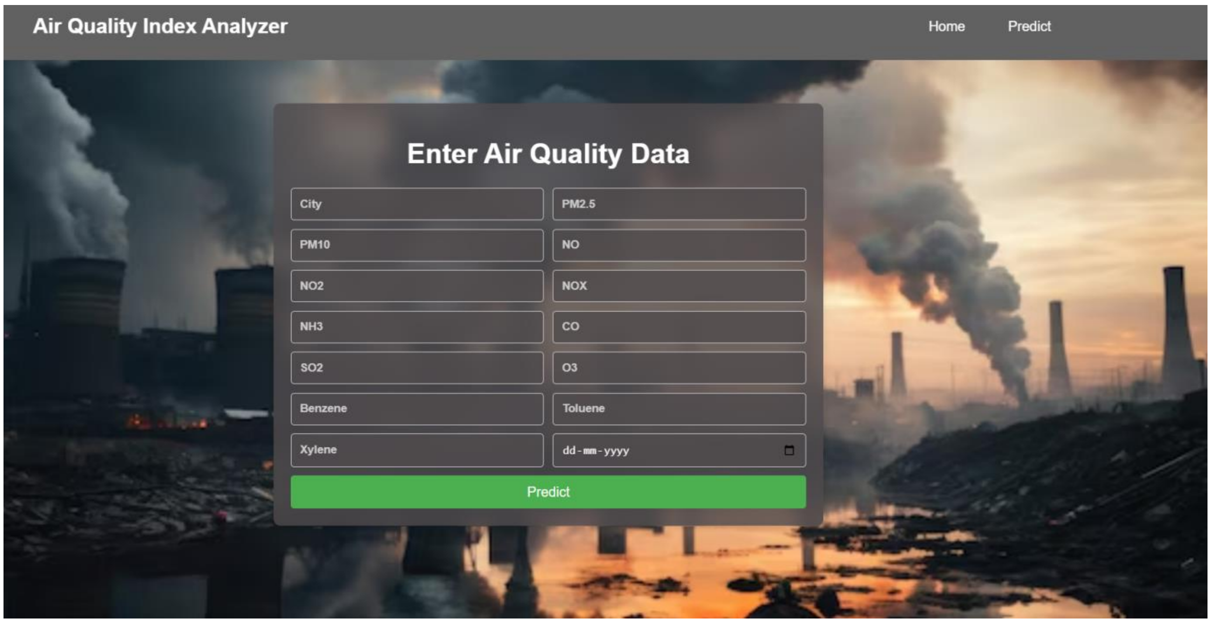
What is AQI?

The Air Quality Index (AQI) is a measure used to communicate how polluted the air currently is or how polluted it is forecast to become. The AQI is calculated based on the levels of various pollutants in the air, including particulate matter (PM10 and PM2.5), ground-level ozone (O3), carbon monoxide (CO), sulfur dioxide (SO2), and nitrogen dioxide (NO2).

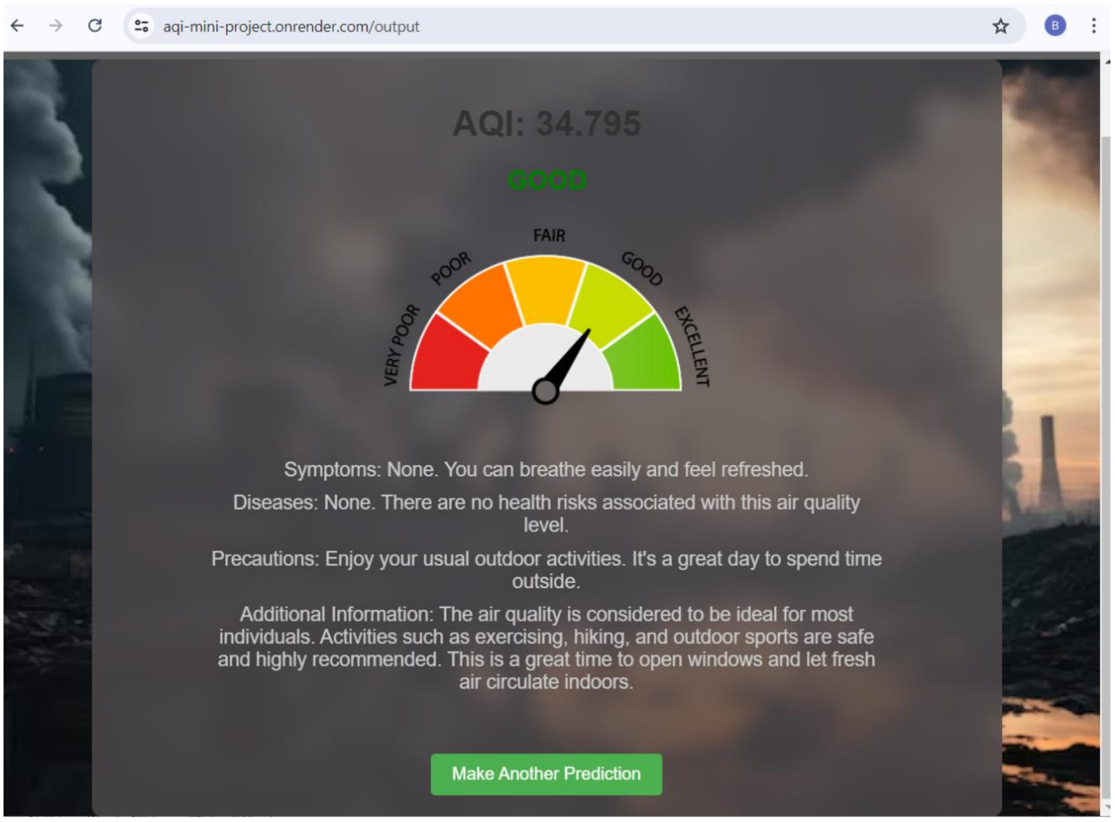
Why is AQI important?

Monitoring and understanding the AQI is crucial for public health as it provides essential information on the safety and quality of the air we breathe. High levels of air pollution can cause a variety of health problems, particularly for vulnerable groups such as children, the elderly, and those with respiratory or cardiovascular conditions. The AQI helps individuals make informed decisions about outdoor activities and take necessary precautions to protect their health.

PREDICTIONS



OUTPUT 1



AQI: 34.795

GOOD

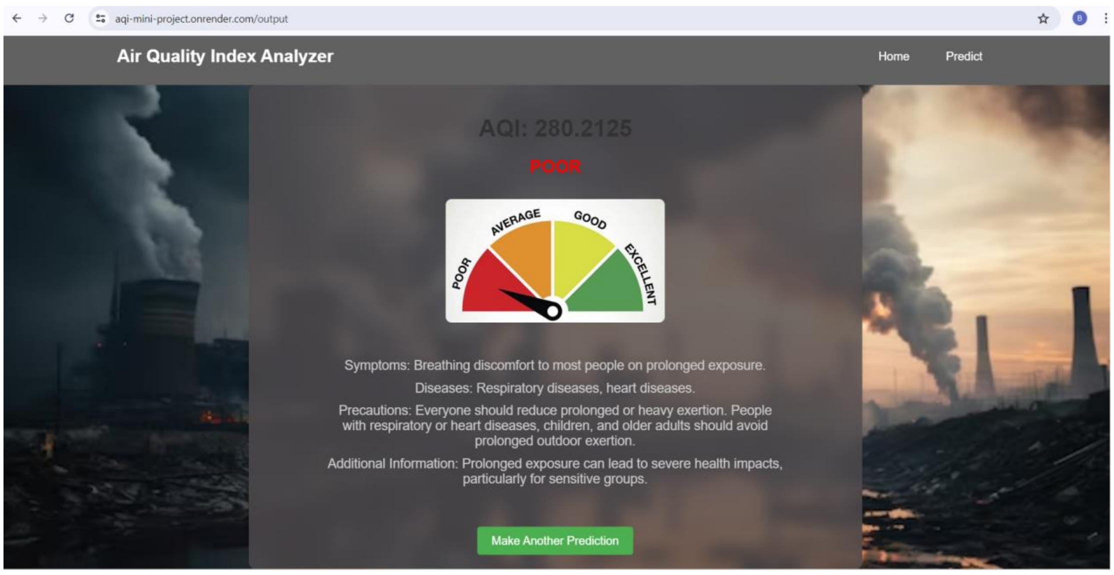
Symptoms: None. You can breathe easily and feel refreshed.

Diseases: None. There are no health risks associated with this air quality level.

Precautions: Enjoy your usual outdoor activities. It's a great day to spend time outside.

Additional Information: The air quality is considered to be ideal for most individuals. Activities such as exercising, hiking, and outdoor sports are safe and highly recommended. This is a great time to open windows and let fresh air circulate indoors.

OUTPUT 2



aqi-mini-project.onrender.com/output

Air Quality Index Analyzer

Home Predict

AQI: 280.2125

POOR

Symptoms: Breathing discomfort to most people on prolonged exposure.

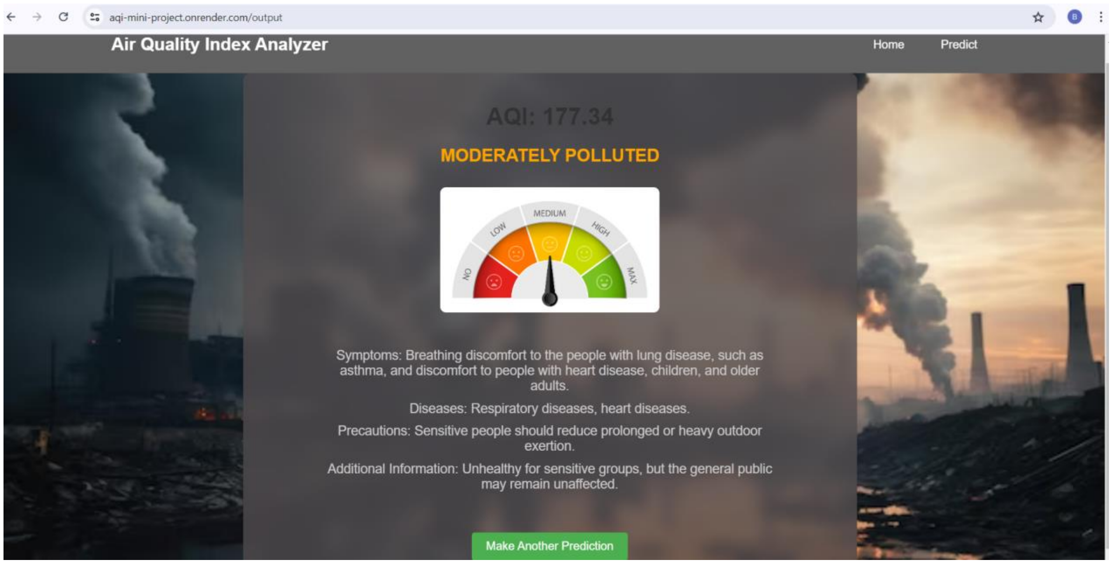
Diseases: Respiratory diseases, heart diseases.

Precautions: Everyone should reduce prolonged or heavy exertion. People with respiratory or heart diseases, children, and older adults should avoid prolonged outdoor exertion.

Additional Information: Prolonged exposure can lead to severe health impacts, particularly for sensitive groups.

Make Another Prediction

OUTPUT 3



aqi-mini-project.onrender.com/output

Air Quality Index Analyzer

Home Predict

AQI: 177.34

MODERATELY POLLUTED

Symptoms: Breathing discomfort to the people with lung disease, such as asthma, and discomfort to people with heart disease, children, and older adults.

Diseases: Respiratory diseases, heart diseases.

Precautions: Sensitive people should reduce prolonged or heavy outdoor exertion.

Additional Information: Unhealthy for sensitive groups, but the general public may remain unaffected.

Make Another Prediction

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. **Enhanced Public Health**: Provides timely air quality information, promoting health protection.

2. **Proactive Decision-Making**: Enables informed decisions regarding outdoor activities.

3. **Environmental Awareness**: Increases awareness of air pollution and its impacts.

4. **Community Engagement**: Engages communities in air quality management and feedback.

5. **Customized Health Information**: Tailors health recommendations to AQI categories.

DISADVANTAGES:

1. **Data Reliance**: Relies on accurate and up-to-date air quality data for precise predictions.

2. **Resource-Intensive**: Requires substantial resources for data collection, modeling, and system maintenance.

3. **Technical Expertise**: Users may need some technical knowledge to interpret AQI and health information.

4. **Model Accuracy**: The system's accuracy depends on the quality of the predictive model.

5. **Privacy Concerns**: User data may raise privacy concerns and necessitate secure data handling.

8.APPLICATIONS

1. **Public Health Protection**: Empowering individuals to make informed decisions regarding outdoor activities, reducing exposure to poor air quality, and minimizing health risks.

2. **Environmental Monitoring**: Assessing the impact of air pollution on the environment, ecosystems, and natural habitats, aiding in conservation efforts.

3. **Government Policy**: Assisting governments and regulatory bodies in setting air quality standards, formulating pollution control policies, and conducting effective urban planning

4. **Public Awareness**: Raising public awareness about the importance of air quality and its impact on health, influencing behavior and lifestyle choice.

9.CONCLUSION

 In conclusion, the proposed Air Quality Index (AQI) prediction and management system presents a holistic solution to address the critical issues of air quality monitoring, public health protection, and community engagement. By integrating advanced predictive modeling with user-friendly interfaces, the system empowers individuals to make informed decisions, safeguard their health, and actively participate in air quality management. The project's key components, including data preprocessing, feature selection, model training, and user interface development, create a comprehensive framework for delivering real-time AQI predictions and associated health information.

 In the ever-evolving field of environmental science and technology, this project represents a significant step forward in ensuring cleaner air and healthier living environments. With ongoing research, innovation, and community involvement, the system has the potential to make a positive impact on public health, environmental conservation, and global collaboration in mitigating air pollution issues

10.FUTURE SCOPE

Future Scope of the AQI Prediction and Management System:

1. **Global Expansion**: Extend the system's reach to more regions and countries, addressing air quality issues on a global scale.

2. **Advanced Technology Integration**: Integrate IoT sensor networks and smart city initiatives for real-time air quality monitoring and urban planning.

3. **Air Quality Forecasting**: Enhance the system's capabilities for short and long term air quality forecasting.

4. **Healthcare Integration**: Collaborate with healthcare providers to incorporate AQI information into patient care, particularly for those with respiratory conditions, improving public health outcomes.

11.BIBILOGRAPHY

[1] Anderson H. R., R.W. Atkinson, J. L. Peacock, M. J. Sweeting and L. Marston. Ambient Particulate matter and health effect;Publication bias in studies of short-term association. Epidemiol 16; 2005: 155- 163.

[2] Analitis A.,K. Katsouyanni, E. Dimakopoulou, A. K. Samoli,Y.Nikolouopoulos, G. Petasakis, J. Touloumi, H. Schwartz, H. R.Anderson, K. Cambra, F. Forastiere, D. Zmirou, J. M. Vonk,L.clancy, B. Kriz, J. Bobvos and J. Pekkanen. Short-term effects

of ambient paricles on cardiovascular and resipiratory mortilaity. Epidemiol 17; 2006: 230-233.

[3] Kumar A.,Goyal P. Forecasting of air quality in Delhi using principal component regression technique. Atmospheric Pollution Research. 2 . 2011: 436-444.

[4] Central Pollution Control Board (CPCB).Guidelines for National ambient air quality monitoring, Series:NAAQM/25/2003-04.Parivesh Bhavan,Delhi; 2009:

[5] Central pollution control board (CPCB). National air quality index , Series CUPS/82/2014-15; 2014:

[6] Ekpenyong E. C.,Eltebong E. O., Akpan E. E., Samson T. K.,Danierl E. N. Urban city transportation mode and respiratory health effect of an air pollution: a cross sectional study among transit and non transit worker in Nigeria. BMJ open,doi:10.1136/bmjopen-2012-001253; 2012:

[7] Kaushik. C. P., Ravindra K., Yadav K., Mehta S. and Haritash A.K.. Assessment of ambient air quality in urban centres of haryana(India) in relation to different anthropogenic activities and health risks. Environment Monitoring and Assement. 122; 2006:27-40.

[8] Pipalatkar. P. P. , Gajghate. D.G and Khaparde V.V. Source identification of different size fraction of PM10 Using Factor analysis at residential cum commercial Area of Nagpur city. Bull.Envionment Contam Toxicol. 88;2012: 260-264.

[9] Pope C. A. III and D. W.Dockery Health effects of fine particulate air pollution; lines that connect. Journal of Air and Waste Management Association. 56; 2006: 709-742.

[10] Bhuyan P. K.,Samantray P., Rout S. P. Ambient air quality status in choudwar area of cuttack district.International Journal of Environmental Sciences. 1; 2010:

[11] Ravikumar,P., Prakash, K.L. and Somashekar,R.K.. Air quality Indices to understand the ambient air quality in the vicity of dam site of different irrigation projects in karanataka state, india.International 22journal of science and nature. 5; 2014 : 531-541.

[12] U. S. Environmental Protection Agency (USEPA). Guidelines for reporting of daily air quality- air quality index (AQI), Series EPA-454/B-06-001. Research Trangle Park , North carolina.2006

12.APPENDIX

Model building :

1)Dataset

2)Jupyter Notebook and Spyder Application Building

1. HTML file (Index file,Home file, Predict file,Output file)

1. CSS file
2. Models in pickle format

<u>SOURCE CODE:</u>

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <meta name="description" content="" />
  <meta name="author" content="" />
  <title>AQI Analyzer</title>
  <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />
  <!-- Font Awesome icons (free version)-->
  <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js" crossorigin="anonymous"></script>
  <!-- Google fonts-->
  <link href="https://fonts.googleapis.com/css?family=Varela+Round" rel="stylesheet" />
  <link href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet" />
  <!-- Core theme CSS (includes Bootstrap)-->
  <link href="{{ url_for('static', filename='styles.css') }}" rel="stylesheet" />
  <style>
    .card {
      margin: 10px; /* Adjust the value as needed */
    }
      .card-bottom-space {
      margin-bottom: 20px; /* Adjust the value as needed */
      padding-bottom: 20px; /* Adjust the value as needed */
```

```
        }
    </style>
</head>
<body id="page-top">
    <!-- Navigation-->
    <nav class="navbar navbar-expand-lg navbar-light fixed-top" id="mainNav">
        <div class="container px-4 px-lg-5">
            <a class="navbar-brand" href="#page-top">AQI Analyzer</a>
            <button class="navbar-toggler navbar-toggler-right" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarResponsive" aria-
controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
                Menu
                <i class="fas fa-bars"></i>
            </button>
            <div class="collapse navbar-collapse" id="navbarResponsive">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item"><a class="nav-link"
href="{{url_for('predict')}}">Predict</a></li>
                    <li class="nav-item"><a class="nav-link" href="#about">About</a></li>
                    <li class="nav-item"><a class="nav-link" href="#team">Team</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <header class="masthead">
        <div class="container px-4 px-lg-5 d-flex h-100 align-items-center justify-
content-center">
            <div class="d-flex justify-content-center">
                <div class="text-center">
                    <h1 class="mx-auto my-0 text-uppercase">Air Quality Index
Analyzer</h1>
```

```
                <h2 class="text-white-50 mx-auto mt-2 mb-5">Your go-to platform for
air quality analysis and predictions.</h2>

                <a class="btn btn-primary" href="{{ url_for('home') }}">Get Started</a>

            </div>

        </div>

    </div>

</header>

<!-- About-->

<section class="about-section text-center" id="about">

    <div class="container px-4 px-lg-5">

        <div class="row gx-4 gx-lg-5 justify-content-center">

            <div class="col-lg-8">

                <h2 class="text-white mb-4">Real Time Air Quality India</h2>

                <p class="text-white-50">

                    The Air Quality Index (AQI) is a measure used to communicate how
polluted the air currently is or how polluted it is forecast to become. The AQI is
calculated based on the levels of various pollutants in the air, including particulate
matter (PM10 and PM2.5), ground-level ozone (O3), carbon monoxide (CO), sulfur
dioxide (SO2), and nitrogen dioxide (NO2)

                    <a href="https://www.aqi.in/dashboard/india">Real time
Overview.</a>

                    Monitoring and understanding the AQI is crucial for public health as it
provides essential information on the safety and quality of the air we breathe. High
levels of air pollution can cause a variety of health problems, particularly for
vulnerable groups such as children, the elderly, and those with respiratory or
cardiovascular conditions. The AQI helps individuals make informed decisions about
outdoor activities and take necessary precautions to protect their health.

                </p>

            </div>

        </div>

    </div>

</section>

<!-- Team-->

<section class="about-section text-center" id="team">
```

```
<div class="container px-4 px-lg-5">
  <div class="row gx-4 gx-lg-5 justify-content-center">
    <div class="col-lg-8">
      <h2 class="text-white mb-4">Our Team</h2>
      <div class="row gx-4 gx-lg-5">
        <div class="col-md-6 mb-3 mb-md-0">
          <div class="card py-4 h-100">
            <div class="card-body text-center">
              <h4 class="text-uppercase m-0">Sheshikumar</h4>
              <hr class="my-4 mx-auto" />
              <div class="small text-black-50">21UK1A6614</div>
            </div>
          </div>
        </div>
        <div class="col-md-6 mb-3 mb-md-0">
          <div class="card py-4 h-100">
            <div class="card-body text-center">
              <h4 class="text-uppercase m-0">Sai Priya</h4>
              <hr class="my-4 mx-auto" />
              <div class="small text-black-50">21UK1A29</div>
            </div>
          </div>
        </div>
        <div class="col-md-6 mb-3 mb-md-0">
          <div class="card py-4 h-100">
            <div class="card-body text-center">
              <h4 class="text-uppercase m-0">Nidhi Reddy</h4>
              <hr class="my-4 mx-auto" />
              <div class="small text-black-50">21UK1A58</div>
            </div>
```

```
                    </div>
                  </div>
                  <div class="col-md-6 mb-3 mb-md-0">
                    <div class="card py-4 h-100">
                      <div class="card-body text-center">
                        <h4 class="text-uppercase m-0">Shiva</h4>
                        <hr class="my-4 mx-auto" />
                        <div class="small text-black-50">21UK1A6646</div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
  </section>
  <!-- Footer-->
  <footer class="footer bg-black small text-center text-white-50">
    <div class="container px-4 px-lg-5">Copyright &copy; Your Website
2023</div>
  </footer>
  <!-- Bootstrap core JS-->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></
script>
  <!-- Core theme JS-->
  <script src="js/scripts.js"></script>
  <!-- SB Forms JS-->
  <script src="https://cdn.startbootstrap.com/sb-forms-latest.js"></script>
</body>
</html>
```

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Predict AQI</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='predict.css') }}">
</head>
<body>
  <header>
    <div class="header-container">
      <h1>Air Quality Index Analyzer</h1>
      <nav>
        <ul>
          <li><a href="{{ url_for('index') }}">Home</a></li>
          <li><a href="{{ url_for('predict') }}">Predict</a></li>
        </ul>
      </nav>
    </div>
  </header>
  <div class="container">
    <h1>Enter Air Quality Data</h1>
    <form action="{{ url_for('output') }}" method="POST">
      <select id="city" name="city" required>
        <option value="" disabled selected>Select City</option>
        <option value="Ahmedabad">Ahmedabad</option>
        <option value="Aizawl">Aizawl</option>
        <option value="Amaravati">Amaravati</option>
        <option value="Amritsar">Amritsar</option>
```

```
            <option value="Bengaluru">Bengaluru</option>
            <option value="Bhopal">Bhopal</option>
            <option value="Brajrajnagar">Brajrajnagar</option>
            <option value="Chandigarh">Chandigarh</option>
            <option value="Chennai">Chennai</option>
            <option value="Coimbatore">Coimbatore</option>
            <option value="Delhi">Delhi</option>
            <option value="Ernakulam">Ernakulam</option>
            <option value="Gurugram">Gurugram</option>
            <option value="Guwahati">Guwahati</option>
            <option value="Hyderabad">Hyderabad</option>
            <option value="Jaipur">Jaipur</option>
            <option value="Jorapokhar">Jorapokhar</option>
            <option value="Kochi">Kochi</option>
            <option value="Kolkata">Kolkata</option>
            <option value="Lucknow">Lucknow</option>
            <option value="Mumbai">Mumbai</option>
            <option value="Patna">Patna</option>
            <option value="Shillong">Shillong</option>
            <option value="Talcher">Talcher</option>
            <option value="Thiruvananthapuram">Thiruvananthapuram</option>
            <option value="Visakhapatnam">Visakhapatnam</option>
        </select>
        <input type="number" id="pm25" name="pm25" placeholder="PM2.5"
step="0.001" required>
        <input type="number" id="pm10" name="pm10" placeholder="PM10"
step="0.001" required>
        <input type="number" id="no2" name="no2" placeholder="NO2"
step="0.001" required>
        <input type="number" id="co" name="co" placeholder="CO" step="0.001"
required>
```

```
            <input type="number" id="so2" name="so2" placeholder="SO2" step="0.001"
required>
            <input type="number" id="o3" name="o3" placeholder="O3" step="0.001"
required>
            <input type="date" id="date" name="date" required>
            <button type="submit" class="button">Predict</button>
        </form>
    </div>
</body>
</html>
```

HOME.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Air Quality Index</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='styles2.css') }}">
</head>
<body>
    <header>

        <div class="header-container">
            <h1>Air Quality Index Analyzer</h1>
            <nav>
                <ul>
                    <li><a href="{{ url_for('index') }}">Home</a></li>
                    <li><a href="{{ url_for('predict') }}">Predict</a></li>
                </ul>
            </nav>
        </div>
    </header>
    <div class="container">
        <div class="content">
            <div class="content-left">
                <img src="{{ url_for('static', filename='AQI.jpg') }}" alt="Air
Quality Image" class="aqi-image">
            </div>
            <div class="content-right">
                <h2>Introduction</h2>
                <h1>What is AQI?</h1>
                <p>The Air Quality Index (AQI) is a measure used to
communicate how polluted the air currently is or how polluted it is
forecast to become. The AQI is calculated based on the levels of various
pollutants in the air, including particulate matter (PM10 and PM2.5),
ground-level ozone (O3), carbon monoxide (CO), sulfur dioxide (SO2),
and nitrogen dioxide (NO2).</p>
                <h3>Why is AQI important?</h3>
                <p>Monitoring and understanding the AQI is crucial for public
health as it provides essential information on the safety and quality of the
air we breathe. High levels of air pollution can cause a variety of health
problems, particularly for vulnerable groups such as children, the elderly,
and those with respiratory or cardiovascular conditions. The AQI helps
individuals make informed decisions about outdoor activities and take
necessary precautions to protect their health.</p>
            </div>
        </div>
    </div>
</body>
```

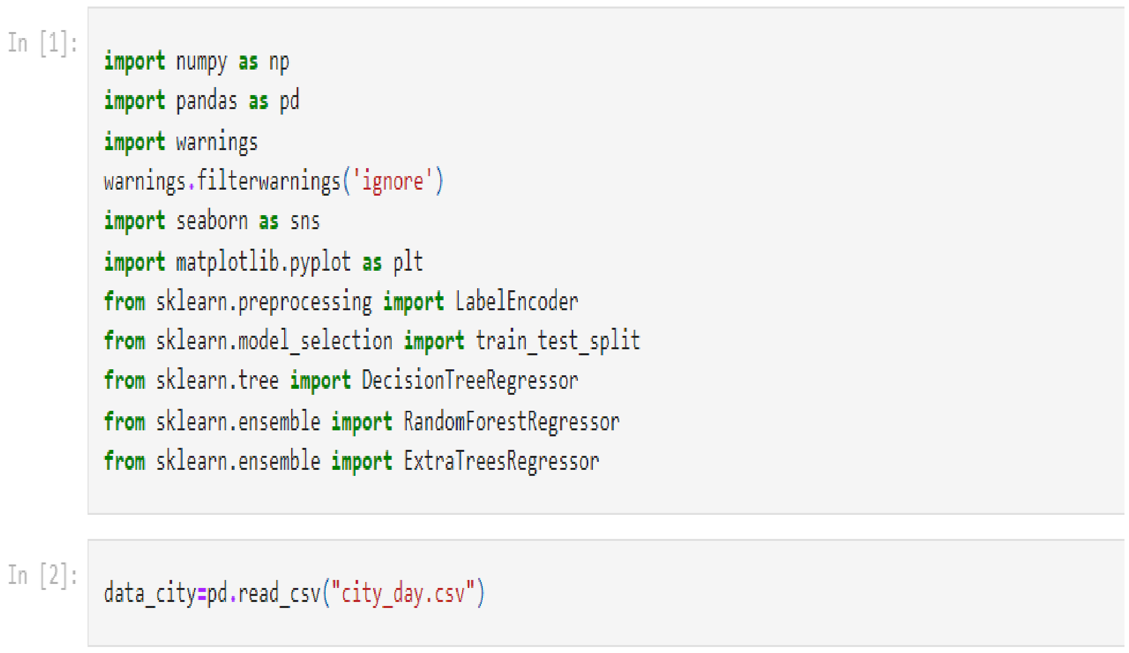
```
</html>
```

APP.PY

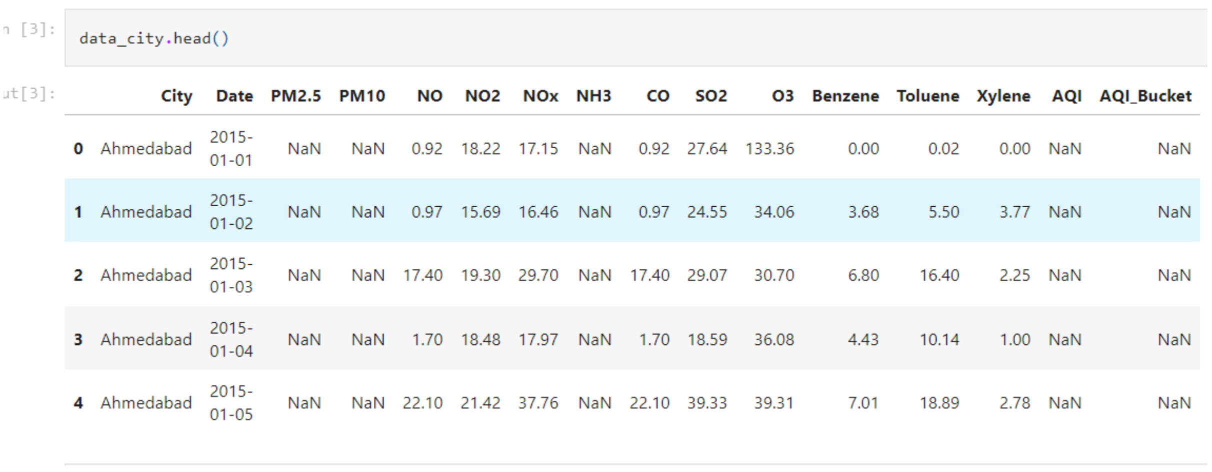
```
from flask import Flask, render_template, request
import joblib
import pandas as pd

app = Flask(__name__)

# Load model and encoded values
model = joblib.load("model_small.pkl")
label_encoder = joblib.load('label_encoder.pkl')

@app.route('/')
def index():
    return render_template("index.html")

@app.route('/home')
def home():
    return render_template("home.html")

@app.route('/predict')
def predict():
    return render_template("predict.html")

@app.route('/output', methods=["POST"])
def output():
    if request.method == 'POST':
        city = request.form["city"].strip()
        pm25 = float(request.form["pm25"])
        pm10 = float(request.form["pm10"])
        no2 = float(request.form["no2"])
        co = float(request.form["co"])
        so2 = float(request.form["so2"])
        o3 = float(request.form["o3"])
        date = request.form["date"]

        # Ensure the city name is valid
        if city not in label_encoder.classes_:
            return render_template("output.html", y="Invalid City", z="Please enter a valid city
name.")

        # Transform city and date fields
        city_encoded = label_encoder.transform([city])[0]
        year = int(date.split('-')[0])
        month = int(date.split('-')[1])

        # Create a DataFrame for the input features
        feature_cols = ['City', 'PM2.5', 'PM10', 'NO2', 'CO', 'SO2', 'O3', 'Year', 'Month']
        data = pd.DataFrame([[city_encoded, pm25, pm10, no2, co, so2, o3, year, month]],
columns=feature_cols)

        # Make prediction
        pred = model.predict(data)
        pred = pred[0]

        # Determine AQI category
        if pred >= 0 and pred < 50:
            res = 'GOOD'
        elif pred >= 50 and pred < 100:
            res = 'SATISFACTORY'
        elif pred >= 100 and pred < 200:
            res = 'MODERATELY POLLUTED'
        elif pred >= 200 and pred < 300:
            res = 'POOR'
        elif pred >= 300 and pred < 400:
            res = 'VERY POOR'
        else:
            res = 'SEVERE'

        return render_template("output.html", y=f"AQI: {str(pred)}", z=res)


if __name__ == "__main__":
    app.run(debug=True)
```

CODE SNIPPETS

MODEL BUILDING



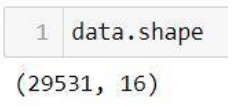
```
In [1]:
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor

In [2]:
data_city=pd.read_csv("city_day.csv")
```



```
n [3]: data_city.head()
```

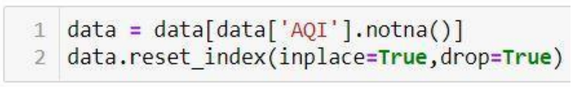
ut[3]:

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN

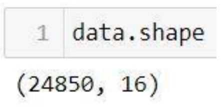


```
data.shape
```

```
(29531, 16)
```

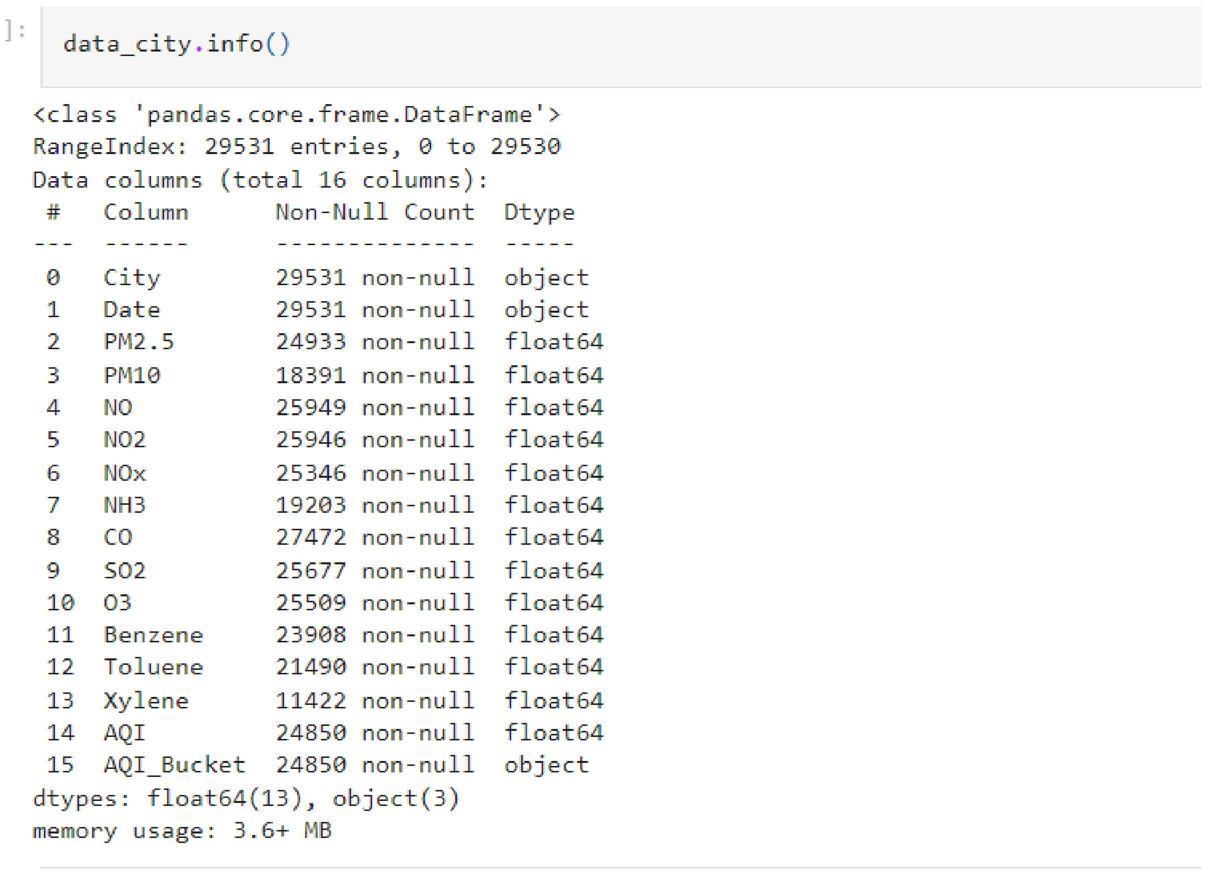


```
data = data[data['AQI'].notna()]
data.reset_index(inplace=True,drop=True)
```



```
data.shape
```

```
(24850, 16)
```



```
]: data_city.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        29531 non-null  object
 1   Date        29531 non-null  object
 2   PM2.5       24933 non-null  float64
 3   PM10        18391 non-null  float64
 4   NO          25949 non-null  float64
 5   NO2         25946 non-null  float64
 6   NOx         25346 non-null  float64
 7   NH3         19203 non-null  float64
 8   CO          27472 non-null  float64
 9   SO2         25677 non-null  float64
 10  O3          25509 non-null  float64
 11  Benzene     23908 non-null  float64
 12  Toluene     21490 non-null  float64
 13  Xylene      11422 non-null  float64
 14  AQI         24850 non-null  float64
 15  AQI_Bucket  24850 non-null  object
dtypes: float64(13), object(3)
memory usage: 3.6+ MB
```

```
In [6]: data_city.nunique()
```

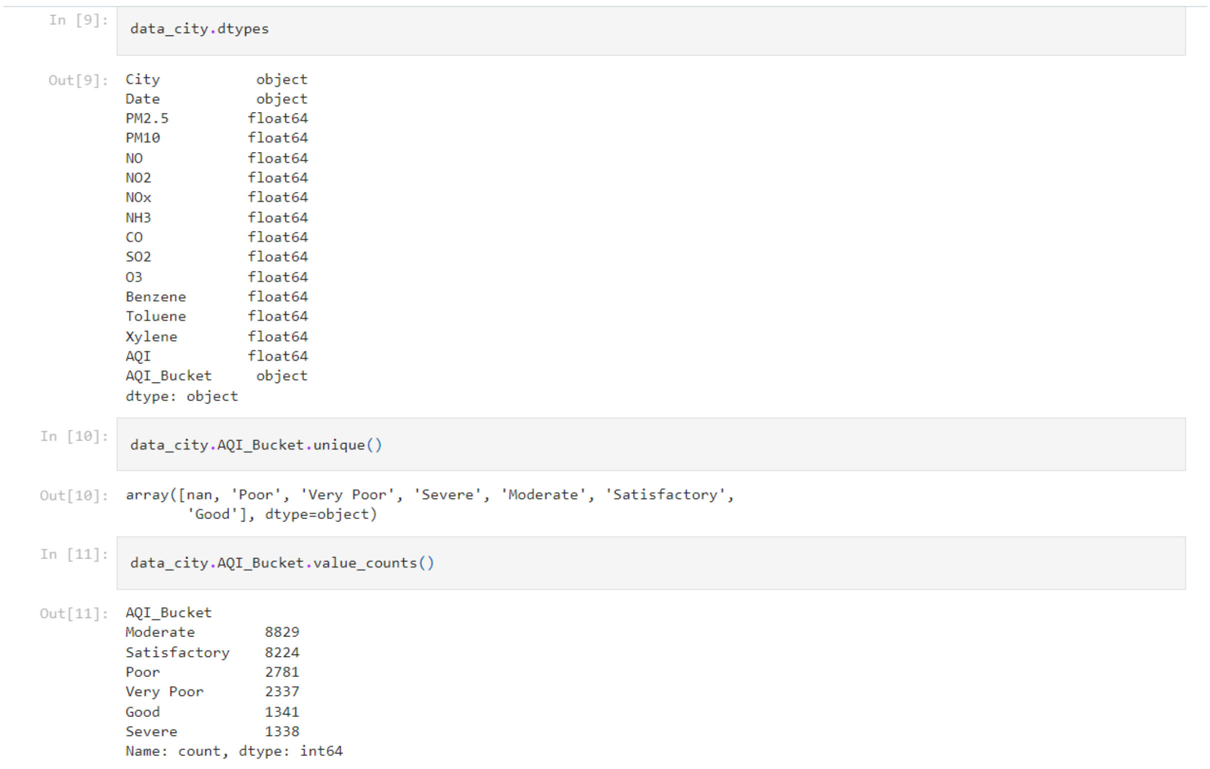
```
Out[6]: City              26
        Date            2009
        PM2.5          11716
        PM10           12571
        NO              5776
        NO2             7404
        NOx             8156
        NH3             5922
        CO              1779
        SO2             4761
        O3              7699
        Benzene         1873
        Toluene         3608
        Xylene          1561
        AQI              829
        AQI_Bucket         6
        dtype: int64
```

```
In [7]: data_city.City.unique()
```

```
Out[7]: array(['Ahmedabad', 'Aizawl', 'Amaravati', 'Amritsar', 'Bengaluru',
               'Bhopal', 'Brajrajnagar', 'Chandigarh', 'Chennai', 'Coimbatore',
               'Delhi', 'Ernakulam', 'Gurugram', 'Guwahati', 'Hyderabad',
               'Jaipur', 'Jorapokhar', 'Kochi', 'Kolkata', 'Lucknow', 'Mumbai',
               'Patna', 'Shillong', 'Talcher', 'Thiruvananthapuram',
               'Visakhapatnam'], dtype=object)
```

```
In [8]: data_city.City.value_counts()
```

```
Out[8]: City
        Ahmedabad             2009
        Delhi                 2009
        Mumbai                2009
        Bengaluru             2009
        Lucknow               2009
        Chennai               2009
        Hyderabad             2006
        Patna                 1858
        Gurugram              1679
        Visakhapatnam         1462
        Amritsar              1221
        Jorapokhar            1169
        Jaipur                1114
        Thiruvananthapuram    1112
        Amaravati              951
        Brajrajnagar           938
        Talcher                925
        Kolkata                814
        Guwahati               502
        Coimbatore             386
        Shillong               310
        Chandigarh             304
        Bhopal                 289
        Ernakulam              162
        Kochi                  162
        Aizawl                 113
        Name: count, dtype: int64
```



```
In [9]:  data_city.dtypes
```

```
Out[9]:  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
```

```
In [10]: data_city.AQI_Bucket.unique()
```

```
Out[10]: array([nan, 'Poor', 'Very Poor', 'Severe', 'Moderate', 'Satisfactory',
                'Good'], dtype=object)
```

```
In [11]: data_city.AQI_Bucket.value_counts()
```

```
Out[11]: AQI_Bucket
         Moderate        8829
         Satisfactory    8224
         Poor            2781
         Very Poor       2337
         Good            1341
         Severe          1338
         Name: count, dtype: int64
```

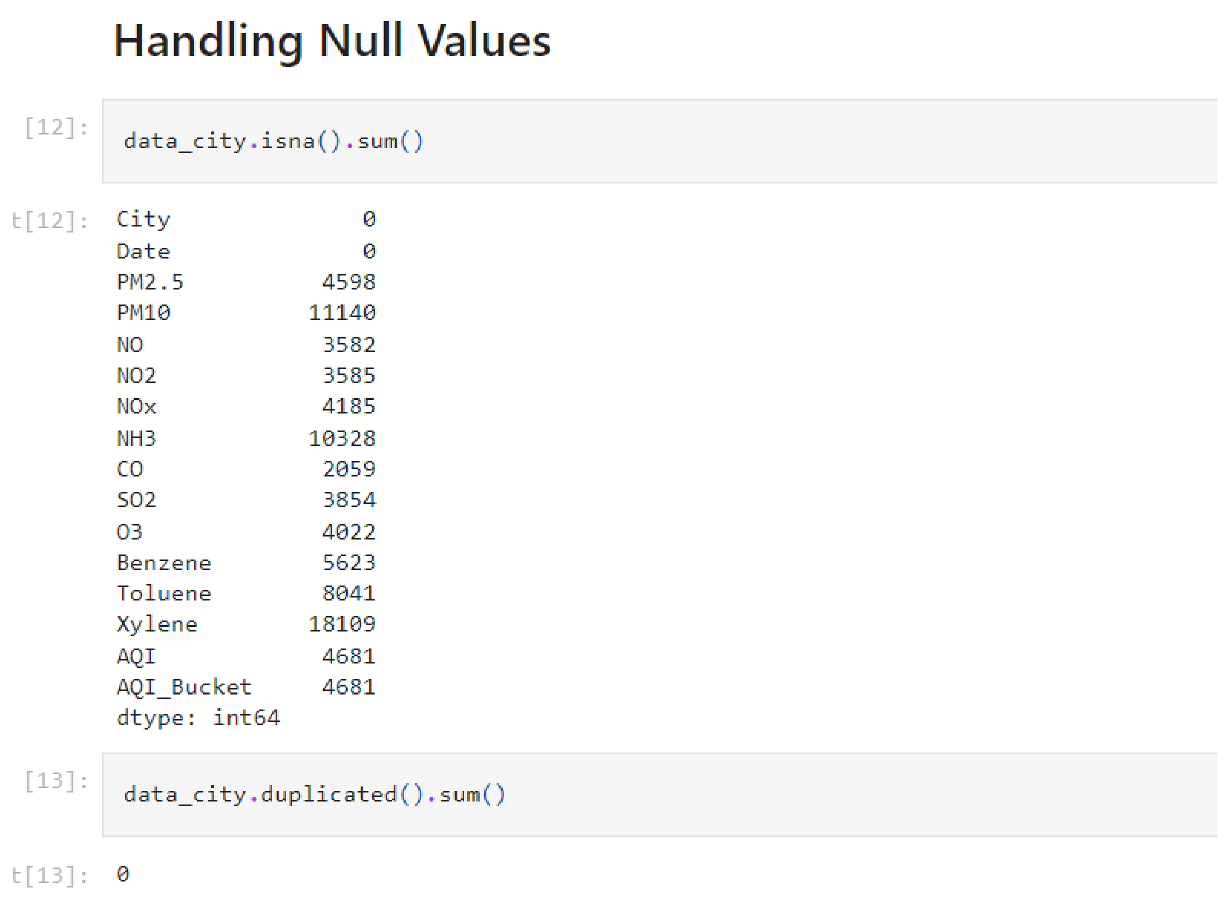
Data Pre-Processing

We need to pre-process the collected data before gaining insights and building our model.

We need to clean the dataset properly in order to fetch good results. This activity includes handling null values and removing unnecessary columns.

Handling Null Values And Removing Unnecessary Columns

Let us first see the count of null values in each column:



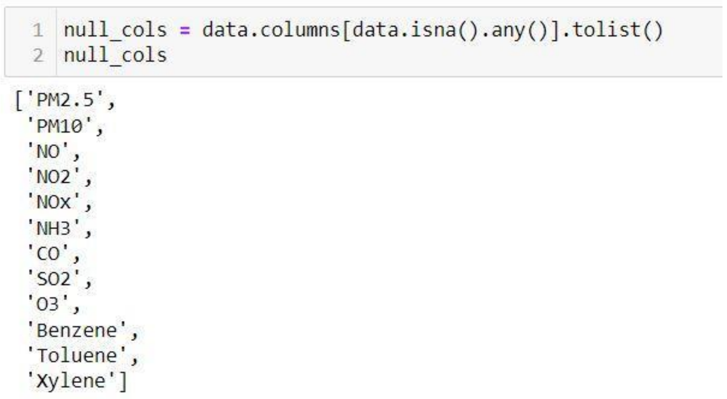
Handling Null Values

```
[12]:  data_city.isna().sum()
```

```
t[12]:  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
```

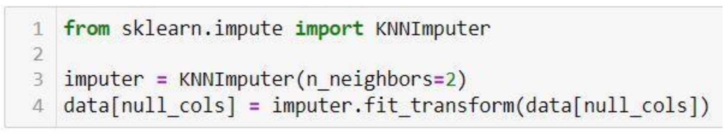
```
[13]:  data_city.duplicated().sum()
```

```
t[13]:  0
```

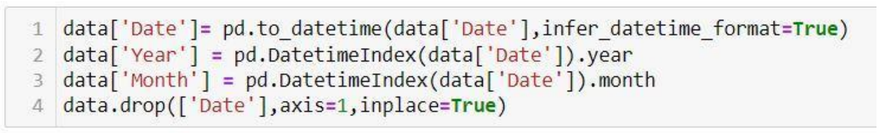


```
null_cols = data.columns[data.isna().any()].tolist()
null_cols
```

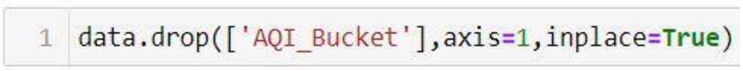
```
['PM2.5',
 'PM10',
 'NO',
 'NO2',
 'NOx',
 'NH3',
 'CO',
 'SO2',
 'O3',
 'Benzene',
 'Toluene',
 'Xylene']
```



```
from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors=2)
data[null_cols] = imputer.fit_transform(data[null_cols])
```



```
data['Date']= pd.to_datetime(data['Date'],infer_datetime_format=True)
data['Year'] = pd.DatetimeIndex(data['Date']).year
data['Month'] = pd.DatetimeIndex(data['Date']).month
data.drop(['Date'],axis=1,inplace=True)
```



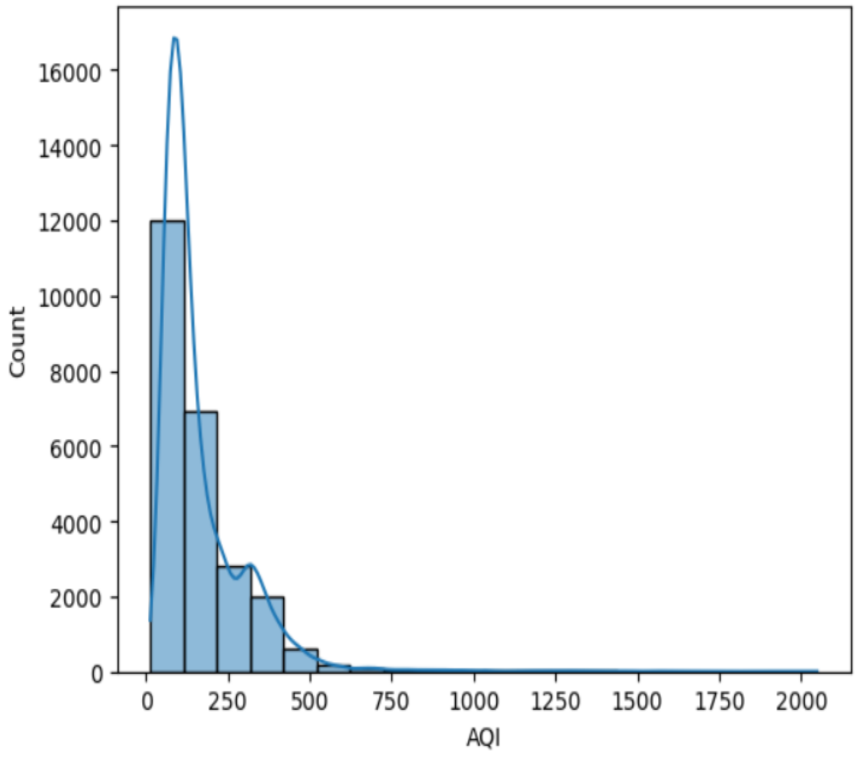
```
data.drop(['AQI_Bucket'],axis=1,inplace=True)
```

Data Analysis And Visualization

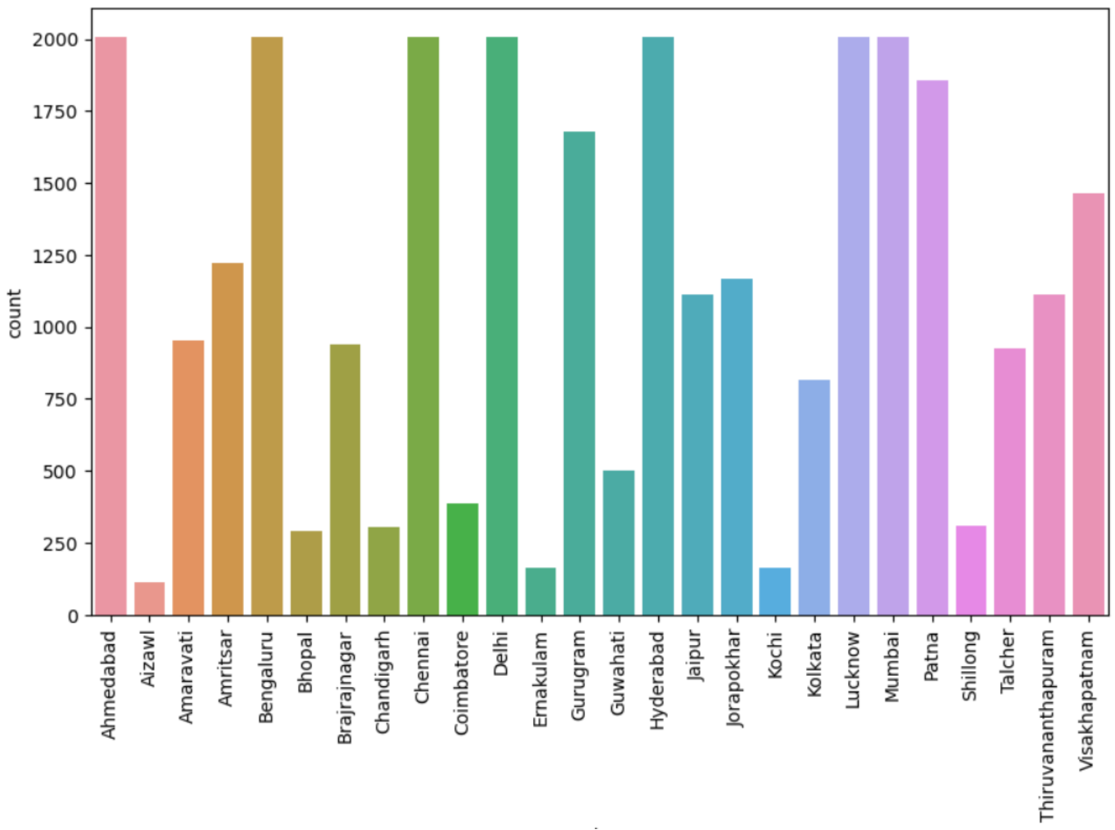
Univariate Analysis

```
[22]: sns.histplot(x=data_city['AQI'],bins=20,kde=True)
```

```
t[22]: <Axes: xlabel='AQI', ylabel='Count'>
```



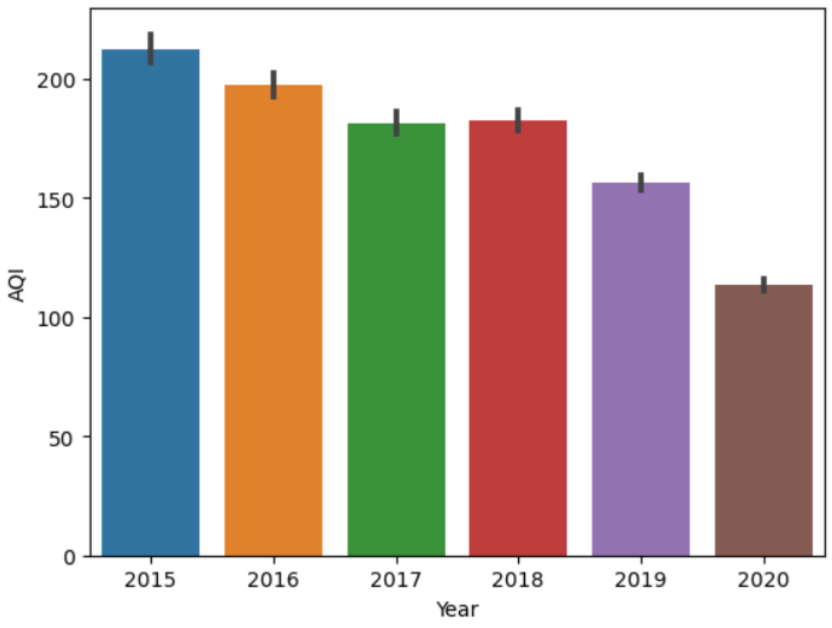
```
]: import matplotlib.pyplot as plt
   plt.figure(figsize=(10, 6))
   sns.countplot(x=data_city['City'])
   plt.xticks(rotation='vertical')
   plt.show()
```



Bivariate Analysis

```
n [19]:  sns.barplot(x=data_city.Year,y=data_city.AQI)
```

```
ut[19]:  <Axes: xlabel='Year', ylabel='AQI'>
```



Descriptive Analysis

n [14]:

```python
data_city.describe()
```

ut[14]:

	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3
count	24933.000000	18391.000000	25949.000000	25946.000000	25346.000000	19203.000000	27472.000000	25677.000000	25509.000000
mean	67.450578	118.127103	17.574730	28.560659	32.309123	23.483476	2.248598	14.531977	34.491430
std	64.661449	90.605110	22.785846	24.474746	31.646011	25.684275	6.962884	18.133775	21.694928
min	0.040000	0.010000	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.010000
25%	28.820000	56.255000	5.630000	11.750000	12.820000	8.580000	0.510000	5.670000	18.860000
50%	48.570000	95.680000	9.890000	21.690000	23.520000	15.850000	0.890000	9.160000	30.840000
75%	80.590000	149.745000	19.950000	37.620000	40.127500	30.020000	1.450000	15.220000	45.570000
max	949.990000	1000.000000	390.680000	362.210000	467.630000	352.890000	175.810000	193.860000	257.730000

n [15]:

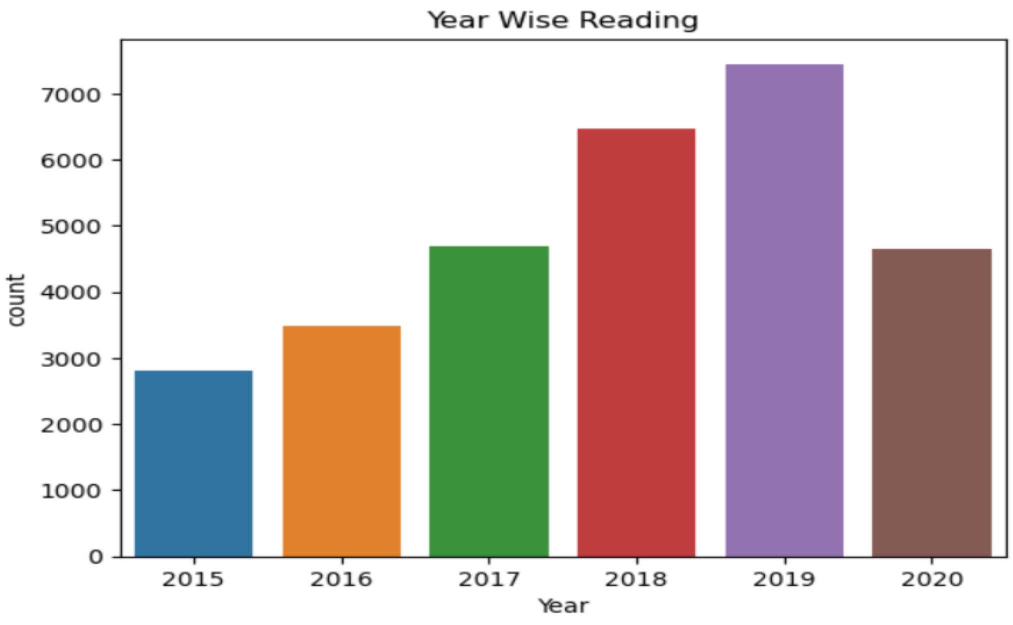
```python
data_city['Date'] = pd.to_datetime(data_city['Date'], infer_datetime_format=True)
data_city['Year'] = pd.DatetimeIndex(data_city['Date']).year
data_city['Month'] = pd.DatetimeIndex(data_city['Date']).month

data_city.drop('Date',axis=1,inplace=True)
```

```python
data_city.head()
```

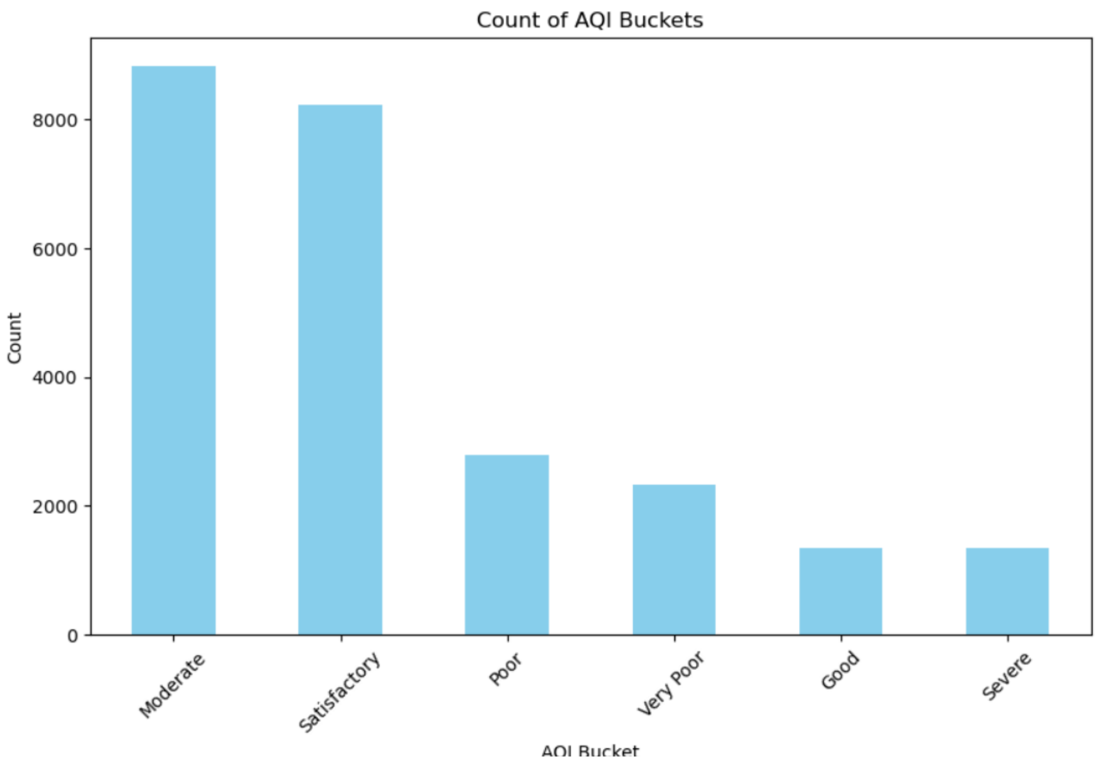
City	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket	Year	Month
Ahmedabad	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN	2015	1
Ahmedabad	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN	2015	1
Ahmedabad	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN	2015	1
Ahmedabad	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN	2015	1
Ahmedabad	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN	2015	1

```python
plt.title("Year Wise Reading")
sns.countplot(x=data_city.Year)
```

```
<Axes: title={'center': 'Year Wise Reading'}, xlabel='Year', ylabel='count'>
```



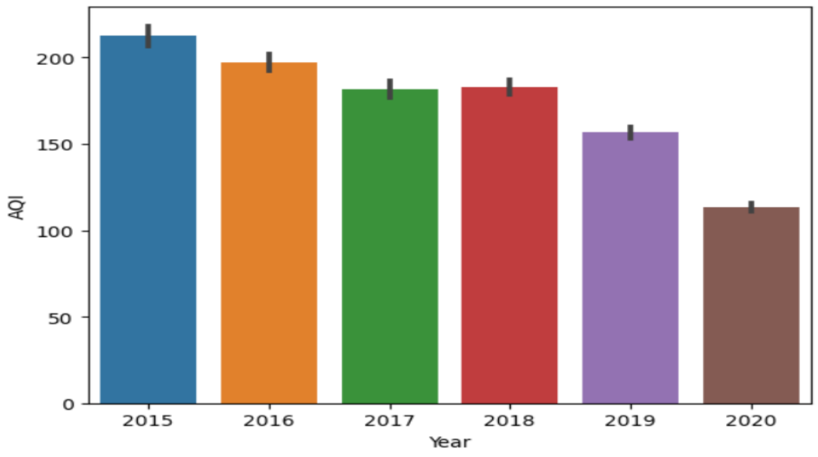
```python
aqi_buckets_count = data_city['AQI_Bucket'].value_counts()
plt.figure(figsize=(10, 6))
aqi_buckets_count.plot(kind='bar', color='skyblue')
plt.title('Count of AQI Buckets')
plt.xlabel('AQI Bucket')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Bar plot

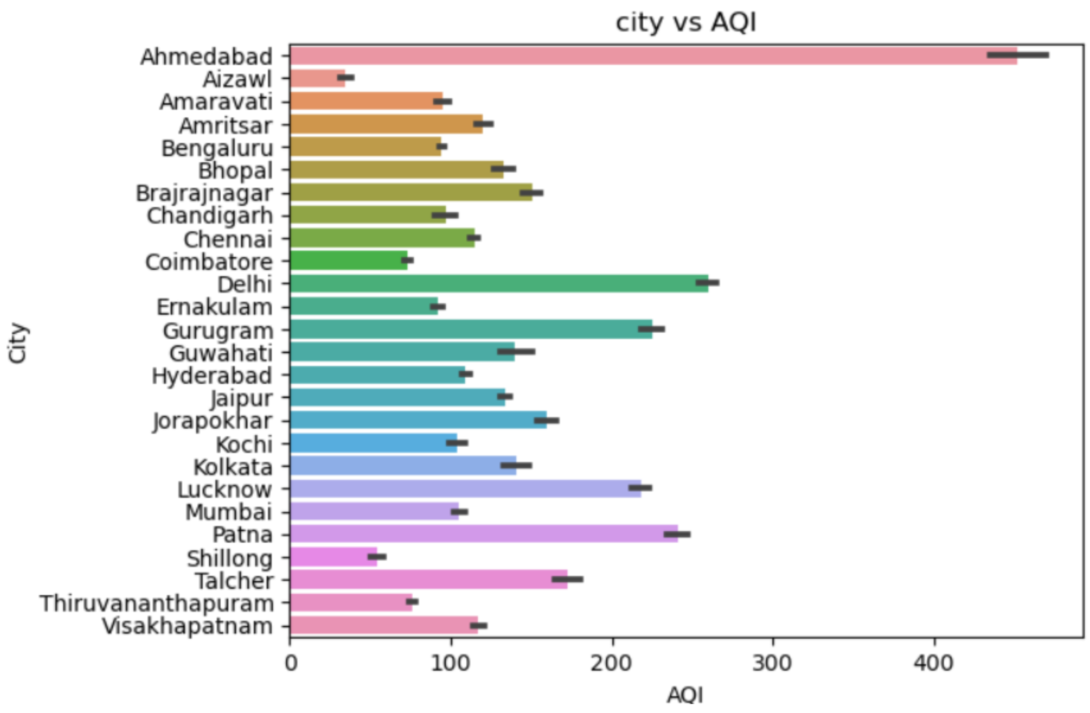
```
3…  sns.barplot(x=data_city.Year,y=data_city.AQI)
```

```
3…  <Axes: xlabel='Year', ylabel='AQI'>
```

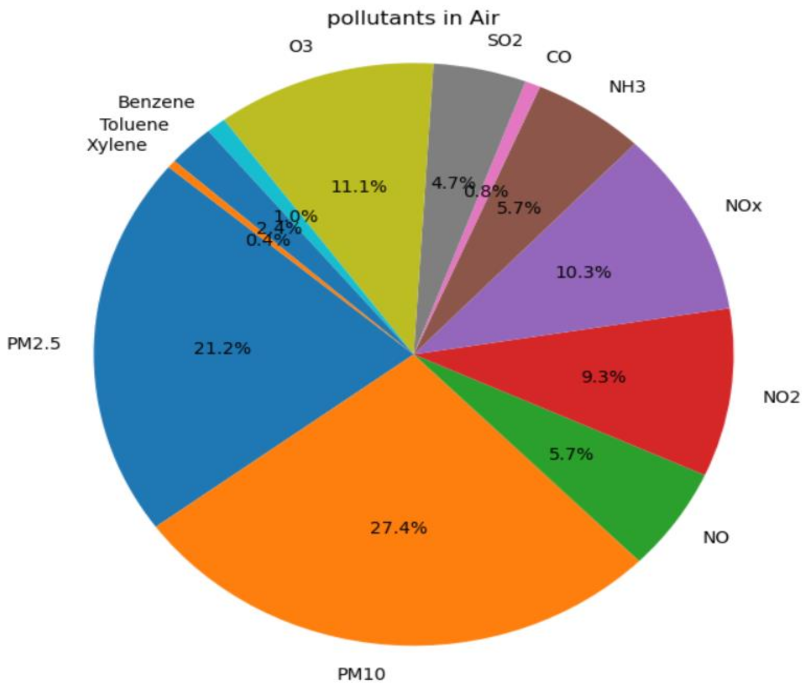


```
sns.barplot(y=data_city.City,x=data_city.AQI)
plt.title("city vs AQI")
```

```
Text(0.5, 1.0, 'city vs AQI')
```



```
parameter = data_city[['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene', 'Toluene', 'Xylene']]
plt.figure(figsize=(10, 7))
plt.pie(parameter.sum(), labels=parameter.columns, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title("pollutants in Air")
plt.show()
```

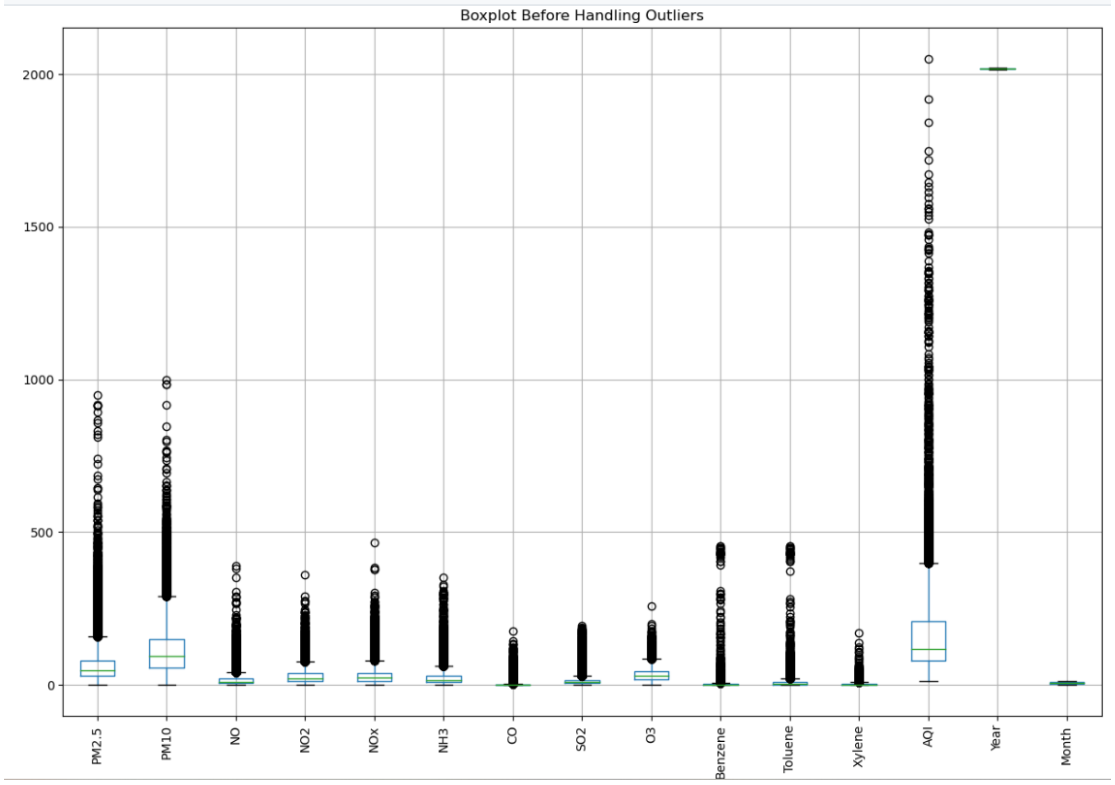


Box ploting

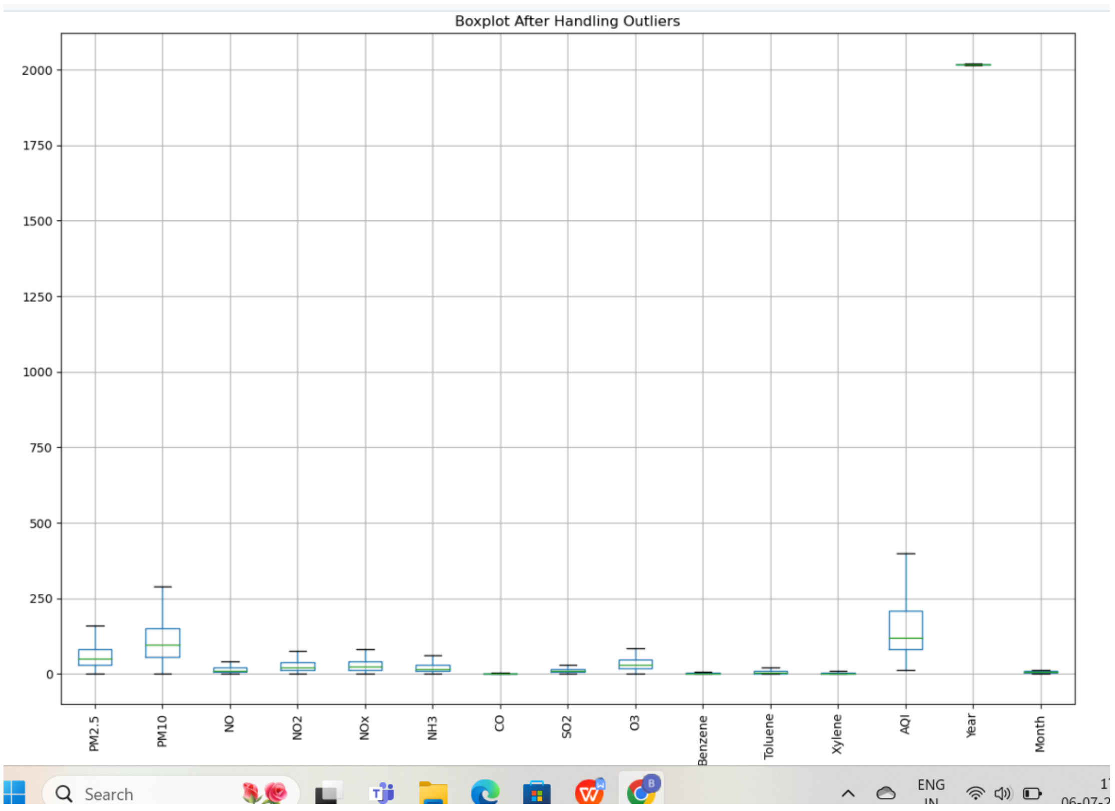
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

def handle_outliers(df):
    # Plot boxplots before handling outliers
    plt.figure(figsize=(15, 10))
    df.boxplot(rot=90)
    plt.title('Boxplot Before Handling Outliers')
    plt.show()

    for column in df.columns:
        if pd.api.types.is_numeric_dtype(df[column]):
            Q1 = df[column].quantile(0.25)
            Q3 = df[column].quantile(0.75)
            IQR = Q3 - Q1
            lower_bound = Q1 - 1.5 * IQR
            upper_bound = Q3 + 1.5 * IQR

            # Cap the outliers
            df[column] = np.where(df[column] < lower_bound, lower_bound,
                                  np.where(df[column] > upper_bound, upper_bound, df[column]))

    # Plot boxplots after handling outliers
    plt.figure(figsize=(15, 10))
    df.boxplot(rot=90)
    plt.title('Boxplot After Handling Outliers')
    plt.show()

    return df
```

```
data_city = handle_outliers(data_city)
```

Before Handling Outliers



After Handling Outliers



```python
from sklearn.impute import KNNImputer
```

```python
imputer=KNNImputer(n_neighbors=2)
```

```python
data_city[null_cols]=imputer.fit_transform(data_city[null_cols])
```

```python
data_city[null_cols]
```

	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
0	34.515	154.750	0.92	18.22	17.15	8.975	0.92	27.640	85.635	0.00	0.02	0.00	93.00
1	25.830	226.235	0.97	15.69	16.46	9.095	0.97	24.550	34.060	3.68	5.50	3.77	125.50
2	36.205	72.125	17.40	19.30	29.70	6.880	2.86	29.070	30.700	6.80	16.40	2.25	238.00
3	25.830	226.235	1.70	18.48	17.97	9.085	1.70	18.590	36.080	4.43	10.14	1.00	177.50
4	54.440	72.125	22.10	21.42	37.76	7.915	2.86	29.545	39.310	7.01	18.89	2.78	254.25
...	...	...	...	...	...	...	...	...	...	...	...	...	...
29526	15.020	50.940	7.68	25.06	19.54	12.470	0.47	8.550	23.300	2.24	12.07	0.73	41.00
29527	24.380	74.090	3.42	26.06	16.53	11.990	0.52	12.720	30.140	0.74	2.21	0.38	70.00
29528	22.910	65.730	3.45	29.53	18.33	10.710	0.48	8.420	30.960	0.01	0.01	0.00	68.00
29529	16.640	49.970	4.05	29.26	18.80	10.030	0.52	9.840	28.300	0.00	0.00	0.00	54.00
29530	15.000	66.000	0.40	26.85	14.05	5.200	0.59	2.100	17.050	0.00	0.00	0.00	50.00

29531 rows × 13 columns

```
data_city.isna().sum()
```

```
City       0
PM2.5      0
PM10       0
NO         0
NO2        0
NOx        0
NH3        0
CO         0
SO2        0
O3         0
Benzene    0
Toluene    0
Xylene     0
AQI        0
Year       0
Month      0
dtype: int64
```

```
data_city
```

	City	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	Year	Montl
0	Ahmedabad	34.515	154.750	0.92	18.22	17.15	8.975	0.92	27.640	85.635	0.00	0.02	0.00	93.00	2015.0	1.(
1	Ahmedabad	25.830	226.235	0.97	15.69	16.46	9.095	0.97	24.550	34.060	3.68	5.50	3.77	125.50	2015.0	1.(
2	Ahmedabad	36.205	72.125	17.40	19.30	29.70	6.880	2.86	29.070	30.700	6.80	16.40	2.25	238.00	2015.0	1.(
3	Ahmedabad	25.830	226.235	1.70	18.48	17.97	9.085	1.70	18.590	36.080	4.43	10.14	1.00	177.50	2015.0	1.(
4	Ahmedabad	54.440	72.125	22.10	21.42	37.76	7.915	2.86	29.545	39.310	7.01	18.89	2.78	254.25	2015.0	1.(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
:9526	Visakhapatnam	15.020	50.940	7.68	25.06	19.54	12.470	0.47	8.550	23.300	2.24	12.07	0.73	41.00	2020.0	6.(
:9527	Visakhapatnam	24.380	74.090	3.42	26.06	16.53	11.990	0.52	12.720	30.140	0.74	2.21	0.38	70.00	2020.0	6.(
:9528	Visakhapatnam	22.910	65.730	3.45	29.53	18.33	10.710	0.48	8.420	30.960	0.01	0.01	0.00	68.00	2020.0	6.(
:9529	Visakhapatnam	16.640	49.970	4.05	29.26	18.80	10.030	0.52	9.840	28.300	0.00	0.00	0.00	54.00	2020.0	6.(
:9530	Visakhapatnam	15.000	66.000	0.40	26.85	14.05	5.200	0.59	2.100	17.050	0.00	0.00	0.00	50.00	2020.0	7.(

)531 rows × 16 columns

```
numerical_cols=pd.DataFrame(data_city.select_dtypes(exclude='object'))
```

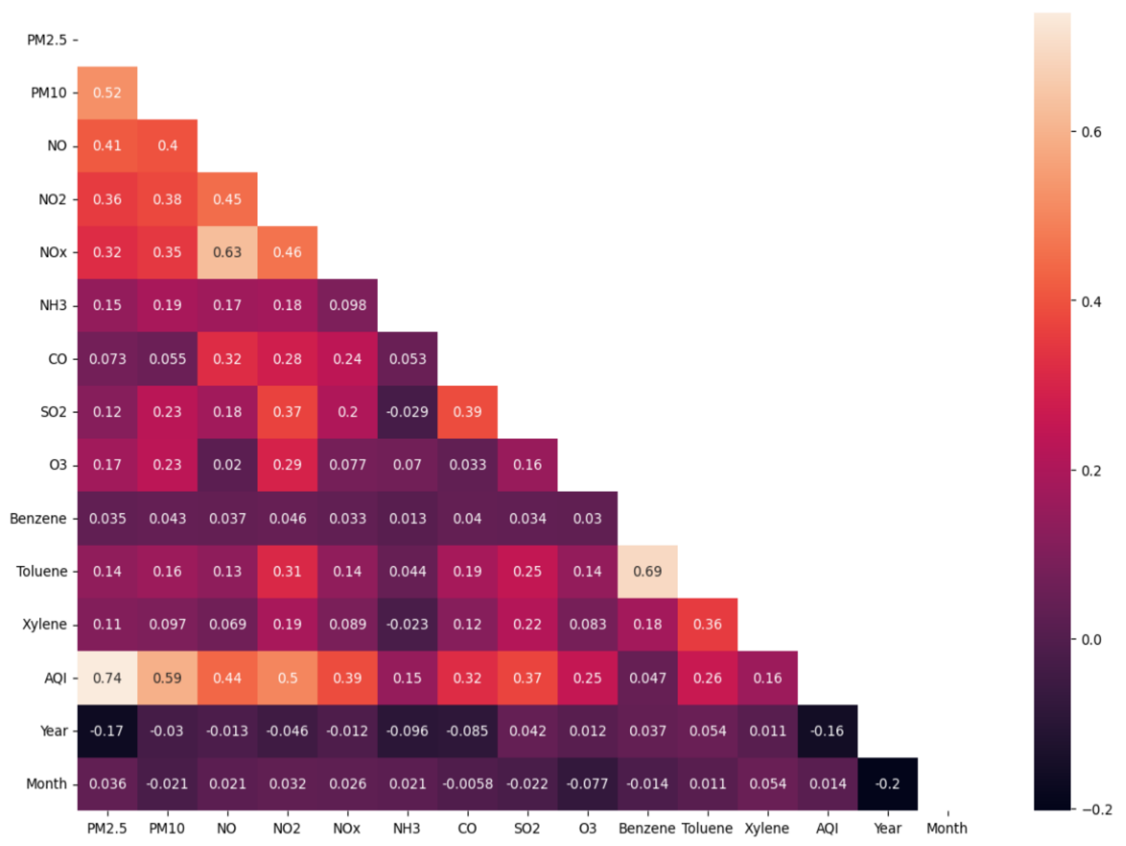
```
numerical_cols
```

	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	Year	Month
0	34.515	154.750	0.92	18.22	17.15	8.975	0.92	27.640	85.635	0.00	0.02	0.00	93.00	2015.0	1.0
1	25.830	226.235	0.97	15.69	16.46	9.095	0.97	24.550	34.060	3.68	5.50	3.77	125.50	2015.0	1.0
2	36.205	72.125	17.40	19.30	29.70	6.880	2.86	29.070	30.700	6.80	16.40	2.25	238.00	2015.0	1.0
3	25.830	226.235	1.70	18.48	17.97	9.085	1.70	18.590	36.080	4.43	10.14	1.00	177.50	2015.0	1.0
4	54.440	72.125	22.10	21.42	37.76	7.915	2.86	29.545	39.310	7.01	18.89	2.78	254.25	2015.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29526	15.020	50.940	7.68	25.06	19.54	12.470	0.47	8.550	23.300	2.24	12.07	0.73	41.00	2020.0	6.0
29527	24.380	74.090	3.42	26.06	16.53	11.990	0.52	12.720	30.140	0.74	2.21	0.38	70.00	2020.0	6.0
29528	22.910	65.730	3.45	29.53	18.33	10.710	0.48	8.420	30.960	0.01	0.01	0.00	68.00	2020.0	6.0
29529	16.640	49.970	4.05	29.26	18.80	10.030	0.52	9.840	28.300	0.00	0.00	0.00	54.00	2020.0	6.0
29530	15.000	66.000	0.40	26.85	14.05	5.200	0.59	2.100	17.050	0.00	0.00	0.00	50.00	2020.0	7.0

29531 rows × 15 columns

```
plt.figure(figsize=(15,10))
mask=np.triu(corr)
sns.heatmap(corr,annot=True,mask=mask)
```

<Axes: >



Handling Categorical Values

```
In [42]: encoded=LabelEncoder()
```

```
In [43]: data_city['City']=encoded.fit_transform(data_city['City'])
```

```
In [44]: import joblib
```

```
In [45]: joblib.dump(encoded,"label_values")
         joblib.dump(encoded, 'label_encoder.pkl')

Out[45]: ['label_encoder.pkl']
```

```
In [46]: data_city['City'].unique()

Out[46]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19, 20, 21, 22, 23, 24, 25])
```

```
In [47]: data_city.columns

Out[47]: Index(['City', 'PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3',
                'Benzene', 'Toluene', 'Xylene', 'AQI', 'Year', 'Month'],
               dtype='object')
```

```
In [48]: X=data_city.drop('AQI',axis=1)
         y=data_city['AQI']
```

```
In [49]: X
Out[49]:
```

	City	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	Year	Month
0	0	34.515	154.750	0.92	18.22	17.15	8.975	0.92	27.640	85.635	0.00	0.02	0.00	2015.0	1.0
1	0	25.830	226.235	0.97	15.69	16.46	9.095	0.97	24.550	34.060	3.68	5.50	3.77	2015.0	1.0
2	0	36.205	72.125	17.40	19.30	29.70	6.880	2.86	29.070	30.700	6.80	16.40	2.25	2015.0	1.0
3	0	25.830	226.235	1.70	18.48	17.97	9.085	1.70	18.590	36.080	4.43	10.14	1.00	2015.0	1.0
4	0	54.440	72.125	22.10	21.42	37.76	7.915	2.86	29.545	39.310	7.01	18.89	2.78	2015.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29526	25	15.020	50.940	7.68	25.06	19.54	12.470	0.47	8.550	23.300	2.24	12.07	0.73	2020.0	6.0
29527	25	24.380	74.090	3.42	26.06	16.53	11.990	0.52	12.720	30.140	0.74	2.21	0.38	2020.0	6.0
29528	25	22.910	65.730	3.45	29.53	18.33	10.710	0.48	8.420	30.960	0.01	0.01	0.00	2020.0	6.0
29529	25	16.640	49.970	4.05	29.26	18.80	10.030	0.52	9.840	28.300	0.00	0.00	0.00	2020.0	6.0
29530	25	15.000	66.000	0.40	26.85	14.05	5.200	0.59	2.100	17.050	0.00	0.00	0.00	2020.0	7.0

29531 rows × 15 columns

```
In [50]: y
Out[50]: 0         93.00
         1        125.50
         2        238.00
         3        177.50
         4        254.25
                   ...
         29526     41.00
         29527     70.00
         29528     68.00
         29529     54.00
         29530     50.00
         Name: AQI, Length: 29531, dtype: float64
```

Splitting Data Into Train, Validation And Test Sets

```
In [51]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

In [52]: X_train.shape,X_test.shape
Out[52]: ((20671, 15), (8860, 15))

In [53]: y_train.shape,y_test.shape
Out[53]: ((20671,), (8860,))

In [54]: X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.3, random_state=42)

In [55]: X_train.shape,X_test.shape,X_val.shape,y_val.shape
Out[55]: ((14469, 15), (8860, 15), (6202, 15), (6202,))
```

Model Building

```
from sklearn.tree import DecisionTreeRegressor
```

```
dt=DecisionTreeRegressor(random_state=42)
```

```
dt.fit(X_train,y_train)
```

```
▾ DecisionTreeRegressor
DecisionTreeRegressor(random_state=42)
```

```
print("R2 Score :{}".format(dt.score(X_test,y_test)))
```

```
R2 Score :0.8070208658711717
```

```
from sklearn.ensemble import RandomForestRegressor


rf_regressor = RandomForestRegressor(random_state=42,n_estimators=20)
rf_regressor.fit(X_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(n_estimators=20, random_state=42)
```

```
print("R2 Score :{}".format(rf_regressor.score(X_test,y_test)))
```

```
R2 Score :0.894994541209092
```

```
from sklearn.ensemble import ExtraTreesRegressor
```

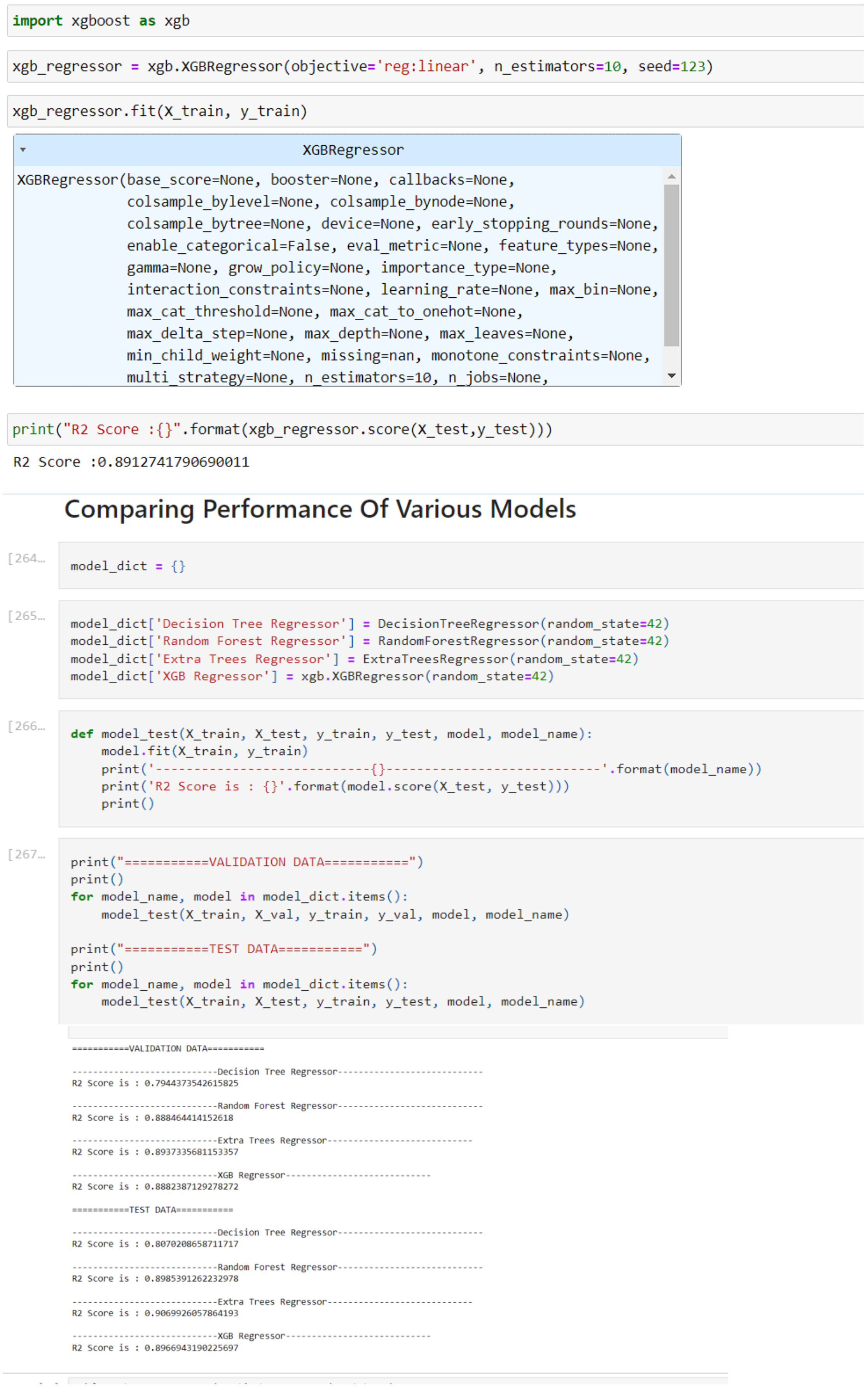
```
et_regressor = ExtraTreesRegressor(n_estimators=100, max_depth=10, random_state=23)
```

```
et_regressor.fit(X_train, y_train)
```

```
ExtraTreesRegressor
ExtraTreesRegressor(max_depth=10, random_state=23)
```

```
print("R2 Score :{}".format(et_regressor.score(X_test,y_test)))
```

```
R2 Score :0.8989213134566164
```



```python
import xgboost as xgb
```

```python
xgb_regressor = xgb.XGBRegressor(objective='reg:linear', n_estimators=10, seed=123)
```

```python
xgb_regressor.fit(X_train, y_train)
```

```
XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=10, n_jobs=None,
```

```python
print("R2 Score :{}".format(xgb_regressor.score(X_test,y_test)))
```

```
R2 Score :0.8912741790690011
```

Comparing Performance Of Various Models

```python
[264…  model_dict = {}
```

```python
[265…  model_dict['Decision Tree Regressor'] = DecisionTreeRegressor(random_state=42)
       model_dict['Random Forest Regressor'] = RandomForestRegressor(random_state=42)
       model_dict['Extra Trees Regressor'] = ExtraTreesRegressor(random_state=42)
       model_dict['XGB Regressor'] = xgb.XGBRegressor(random_state=42)
```

```python
[266…  def model_test(X_train, X_test, y_train, y_test, model, model_name):
           model.fit(X_train, y_train)
           print('---------------------------{}---------------------------'.format(model_name))
           print('R2 Score is : {}'.format(model.score(X_test, y_test)))
           print()
```

```python
[267…  print("===========VALIDATION DATA===========")
       print()
       for model_name, model in model_dict.items():
           model_test(X_train, X_val, y_train, y_val, model, model_name)

       print("===========TEST DATA===========")
       print()
       for model_name, model in model_dict.items():
           model_test(X_train, X_test, y_train, y_test, model, model_name)
```

```
===========VALIDATION DATA===========

---------------------------Decision Tree Regressor---------------------------
R2 Score is : 0.7944373542615825

---------------------------Random Forest Regressor---------------------------
R2 Score is : 0.888464414152618

---------------------------Extra Trees Regressor---------------------------
R2 Score is : 0.8937335681153357

---------------------------XGB Regressor---------------------------
R2 Score is : 0.8882387129278272

===========TEST DATA===========

---------------------------Decision Tree Regressor---------------------------
R2 Score is : 0.8070208658711717

---------------------------Random Forest Regressor---------------------------
R2 Score is : 0.8985391262232978

---------------------------Extra Trees Regressor---------------------------
R2 Score is : 0.9069926057864193

---------------------------XGB Regressor---------------------------
R2 Score is : 0.8966943190225697
```

```
model = ExtraTreesRegressor(n_estimators=100, random_state=42)
```

```
model.fit(X_train,y_train)
```

```
▾          ExtraTreesRegressor
ExtraTreesRegressor(random_state=42)
```

```
print("Val R2 score:{},test R2 score:{}".format(model.score(X_val,y_val),model.score(X_test,y_test)))
```

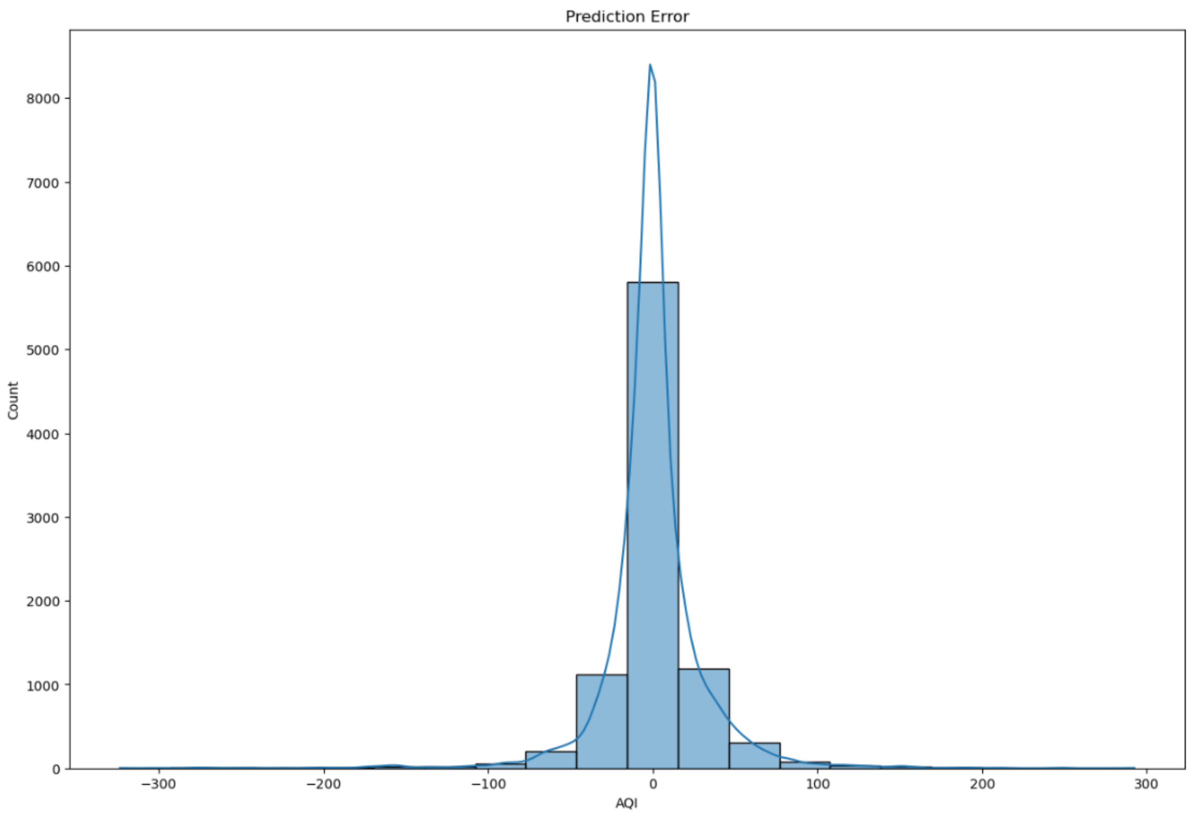
```
Val R2 score:0.8937335681153357,test R2 score:0.9069926057864193
```

```
y_pred=model.predict(X_test)
```

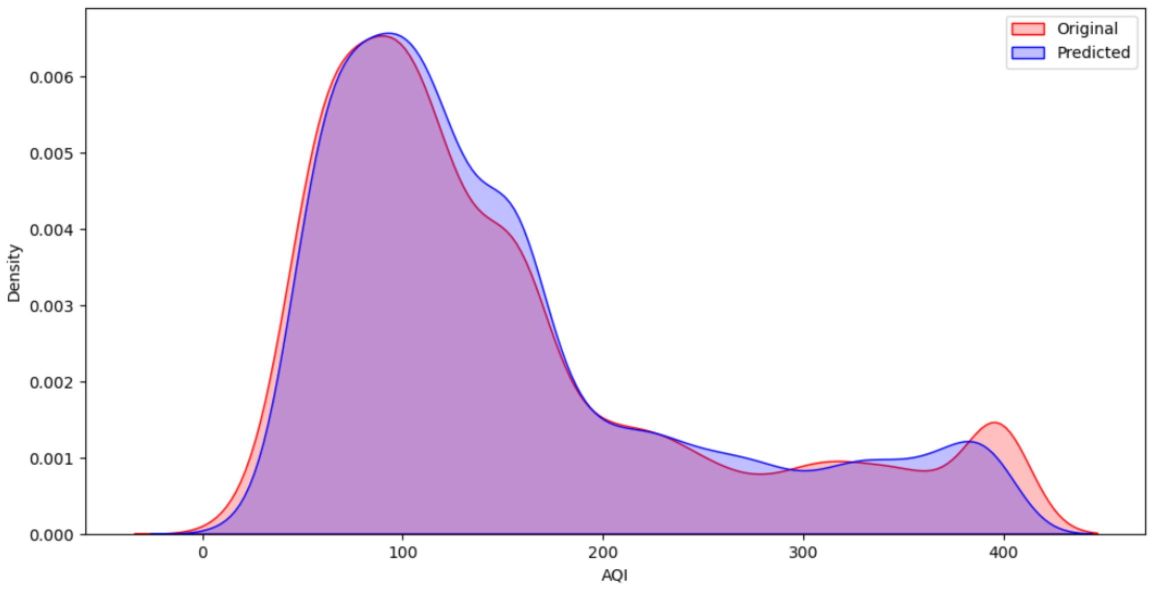
Evaluating Model Performance

```
plt.figure(figsize=(15,10))
sns.histplot(y_test-y_pred,bins=20,kde=True)
plt.title("Prediction Error")
```

```
Text(0.5, 1.0, 'Prediction Error')
```



```
80]: fig, ax = plt.subplots(figsize=(12, 6))
     sns.kdeplot(data=y_test, color='red', label='Original', fill=True, ax=ax)
     sns.kdeplot(data=y_pred, color='blue', label='Predicted', fill=True, ax=ax)
     plt.legend()
     plt.xlabel('AQI')
     plt.ylabel('Density')
     plt.show()
```



Saving The Final Model

```
import pickle as pkl
```

```
pkl.dump(model,open("model.pkl","wb"))
```

```
import joblib

# Load the model
model = joblib.load('model.pkl')

# Save the model to a new file with a smaller size limit
joblib.dump(model, 'model_small.pkl', compress=9)
```

```
['model_small.pkl']
```