AIR QUALITY ALERT SYSTEM

**MINI PROJECT (REVIEW2)**

***Submitted by***

# SHYAM KUMAR A 212221230098

***in partial fulfilment for the award of the degree of***

# BACHELOR OF TECHNOLOGY

***in***

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**



**SAVEETHA ENGINEERING COLLEGE, THANDALAM**

**An Autonomous Institution Affiliated to**

# ANNA UNIVERSITY - CHENNAI 600 025

NOVEMBER 2024

****

ANNA UNIVERSITY, CHENNAI

# BONAFIDE CERTIFICATE

Certified that this Project report **“AIR QUALITY ALERT SYSTEM”** is the bonafide work of **SHYAM KUMAR A (212221230098),** who carried out this project work under my supervision.

|  |  |
| --- | --- |
| SIGNATURE **Dr. Lavnya G**  **Associate Professor**  SUPERVISOR Dept of Artificial Intelligence and Data Science  Saveetha Engineering College, Chennai 602105 | SIGNATURE **Dr. Karthi Govindharaju, M.E., Ph.D.,**  **Professor** HEAD OF THE DEPARTMENT Dept of Artificial Intelligence and Data Science  Saveetha Engineering College, Chennai 602105. |

DATE OF THE VIVA VOCE EXAMINATION: …………………………

## INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude to our esteemed Founder President **Dr. N. M. Veeraiyan**, our President **Dr. Saveetha Rajesh**, our Director **Dr. S. Rajesh**, and the entire management team for providing the essential infrastructure.

I extend my sincere appreciation to our principal, **Dr. V. Vijaya Chamundeeswari, M.Tech., Ph.D.,** for creating a supportive learning environment for this project.

I am very thankful to our Dean of ICT, **Mr. Obed Otto,** M.E., for facilitating a conducive atmosphere that allowed me to complete my project successfully.

My thanks go to **Dr. Karthi Govindharaju, M.E., Ph.D.,**, Professor and Head of the Department of Artificial Intelligence and Data Science at Saveetha Engineering College, for his generous support and for providing the necessary resources for my project work.

I would also like to express my profound gratitude to my Supervisor, **<<Supervisor Name with Designation>>**, and my Project Coordinator **Dr. N.S. Gowri Ganesh**, Associate Professor at Saveetha Engineering College, for their invaluable guidance, suggestions, and constant encouragement, which were instrumental in the successful completion of this project. Their timely support and insights during the review process were greatly appreciated.

I am grateful to all my college faculty, staff, and technicians for their cooperation throughout the project. Finally, I wish to acknowledge my loving parents, friends, and well-wishers for their encouragement in helping me achieve this milestone.

**ABSTRACT**

Air pollution has emerged as a significant global health concern, with detrimental effects on human health and the environment. To address this issue, an Air Quality Alert System (AQAS) is proposed. The AQAS leverages advanced IoT technology and real-time data analytics to monitor and analyse air quality parameters in a specific region. By continuously gathering data from a network of strategically placed sensors, the system provides accurate and timely information on pollutant levels, enabling timely alerts and proactive measures.

The AQAS incorporates a user-friendly interface that allows for easy access to air quality data and alerts. This empowers individuals, communities, and authorities to make informed decisions regarding health, activities, and policy implementation. Furthermore, the system integrates with existing infrastructure, such as weather forecasting and emergency management systems, to enhance overall environmental monitoring and response capabilities. By providing a comprehensive and accessible platform for air quality information, the AQAS aims to contribute to improved public health, environmental protection, and sustainable development. Ultimately, it serves as a valuable tool for mitigating the adverse impacts of air pollution and promoting a healthier and more resilient society.

The AQAS utilizes advanced data analytics techniques to process and interpret the collected data, identifying trends, patterns, and potential pollution hotspots. These insights enable targeted interventions and policy development to address specific air quality challenges. Furthermore, the system incorporates machine learning algorithms to improve prediction accuracy and enhance the effectiveness of alerts.

To ensure the reliability and accuracy of the AQAS, regular calibration and maintenance of the sensor network are essential. Additionally, collaborations with local communities, environmental organizations, and government agencies are crucial for data validation, public awareness, and policy implementation. By fostering partnerships and promoting citizen engagement, the AQAS can maximize its impact and contribute to a healthier and more sustainable environment.

# TABLE OF CONTENTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHAPTER NO.** | | | **TITLE** | **Page**  **Number** |
| **1** |  |  | **INTRODUCTION** |  |
|  | 1.1 | Overview of the project | 1 |
|  | 1.2 | Problem Definition | 2 |
| **2** |  |  | **LITERATURE SURVEY** | 3 |
| **3** |  |  | **SYSTEM ANALYSIS** |  |
|  | 3.1 |  | Existing System | 9 |
|  | 3.2  3.3  3.4 |  | Disadvantages Proposed System Advantages | 9  9  10  10 |
|  | 3.5 |  | Feasibility Study | 10 |
|  | 3.6 |  | Hardware Environment | 10 |
|  | 3.7 |  | Software Environment | 10 |
|  | 3.8 |  | Technologies Used | 10 |
|  |  | 3.8. | Python | 11 |
|  |  | 3.8. | Machine Learning | 11 |
| **4** |  |  | **SYSTEM DESIGN** |  |
|  | 4.1 |  | ER- Diagram | 12 |
|  | 4.2 |  | Data Flow Diagram | 13 |
|  | 4.3 |  | UML Diagram | 14 |
|  |  | 4.3.1 | Use Case Diagram | 14 |
|  |  | 4.3.2 | Class Diagram | 15 |
|  |  | 4.3.3 | Sequence Diagram | 16 |
| **5** |  |  | **SYSTEM ARCHITECTURE** |  |
|  | 5.1 |  | Architecture Diagram | 17 |
|  | 5.2 |  | Algorithms | 18 |
|  |  | 5.2.1 | XGB Regressor | 18 |
| **6** |  |  | **SYSTEM IMPLEMENTATION** |  |
|  | 6.1 | Module-1 | 20 |
|  |  | Data Collection and Preprocessing |  |
|  | 6.2 | Module-2 | 21 |
|  |  | Model Training |  |
|  | 6.3 | Module-3 | 22 |
|  |  | Prediction of Output |  |
| **7** |  |  | **SYSTEM TESTING** |  |
|  | 7.1  7.2 | Black Box Testing White Box Testing | 23  23  24 |
|  | 7.3 | Test Cases |  |
| **8** |  |  | **CONCLUSION AND FUTURE** |  |
|  |  | **-ENHANCEMENT** |  |
|  | 8.1 | Conclusion | 26 |
|  | 8.2 | Future Enhancement | 27 |
| **9** | 9.1 |  | **APPENDIX-1**  Source Code | 28 |
| **10** |  |  | **APPENDIX-2** |  |
|  |  | Sample Output |  |
|  | 10.1 | Home Page | 35 |
|  |  | Detection with Details  Alert Message | 36  38 |
| **11** |  |  | **REFERENCES** | 39 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **XG Boost**  **XGB Classifier**  **AQAS** | Extreme Gradient Boosting  Extreme Gradient Boosting Classifier  Air Quality Alert System |

**Chapter 1**

**INTRODUCTION**

* 1. **OVERVIEW OF THE PROJECT**

Air pollution has emerged as a pressing global health concern, with detrimental effects on human health, the environment, and overall quality of life. The increasing urbanization and industrialization have led to a significant rise in pollutants such as particulate matter, nitrogen oxides, sulfur dioxide, and volatile organic compounds. These pollutants can cause a range of respiratory and cardiovascular diseases, as well as contribute to climate change. To address this pressing issue, an Air Quality Alert System (AQAS) is proposed. The AQAS aims to provide real-time monitoring and early warning of air pollution levels, enabling individuals, communities, and authorities to take proactive measures to protect public health and the environment. The AQAS will play a crucial role in promoting public health and environmental sustainability. By providing accurate and accessible air quality information, it will empower individuals to make informed choices about their daily activities, such as exercising outdoors or avoiding exposure to polluted areas.

Additionally, the AQAS will assist government agencies in developing and implementing effective air pollution mitigation strategies, such as improving transportation systems, reducing industrial emissions, and promoting cleaner energy sources. The AQAS will also contribute to climate change mitigation by providing valuable data on air pollutants that contribute to greenhouse gas emissions. By monitoring and analyzing these emissions, the AQAS can support efforts to reduce carbon footprints and transition towards a more sustainable future. Furthermore, the AQAS can be integrated with other environmental monitoring systems, such as weather forecasting and water quality monitoring, to provide a comprehensive understanding of environmental conditions and their potential impacts on human health and ecosystems. In conclusion, the Air Quality Alert System is a critical tool for addressing the pressing issue of air pollution. By providing real-time monitoring, early warning, and data-driven insights, the AQAS can empower individuals, communities, and authorities to take proactive measures to protect public health and the environment.

Through its contributions to public health, environmental sustainability, and climate change mitigation, the AQAS can help create a healthier and more resilient future for all. One of the major challenges in addressing air pollution is the lack of real-time, accessible information on air quality levels. This makes it difficult for individuals, communities, and authorities to make informed decisions about their activities and policies. Traditional air quality monitoring methods often involve fixed monitoring stations that may not provide adequate coverage of the entire area. To address these challenges, an Air Quality Alert System (AQAS) is necessary. The AQAS will provide real-time monitoring and early warning of air pollution levels, enabling individuals, communities, and authorities to take proactive measures to protect public health and the environment.

The AQAS will also address the issue of limited public access to air quality information. By providing a user-friendly interface and accessible data, the AQAS will empower individuals to make informed choices about their daily activities, such as exercising outdoors or avoiding exposure to polluted areas. Additionally, the AQAS will assist government agencies in developing and implementing effective air pollution mitigation strategies, such as improving transportation systems, reducing industrial emissions, and promoting cleaner energy sources. Furthermore, the AQAS will contribute to climate change mitigation by providing valuable data on air pollutants that contribute to greenhouse gas emissions. By monitoring and analyzing these emissions, the AQAS can support efforts to reduce carbon footprints and transition towards a more sustainable future. Traditional air quality monitoring systems often rely on fixed monitoring stations that measure pollutant levels at specific locations. These stations are typically equipped with sensors to detect various pollutants such as particulate matter, nitrogen oxides, sulfur-dioxide, and volatile organic compounds.

## 1.2. PROBLEM DEFINITION

Air pollution has emerged as a significant global health concern, with detrimental effects on human health, the environment, and overall quality of life. The increasing urbanization and industrialization have led to a rise in pollutants such as particulate matter, nitrogen oxides, sulfur dioxide, and volatile organic compounds. These pollutants can cause a range of respiratory and cardiovascular diseases, as well as contribute to climate change.

**Chapter 2**

**LITREATURE SURVEY**

**2.1 INTRODUCTION**

The impact of air pollution on public health is substantial, and accurate long-term predictions of air quality are crucial for early warning systems to address this issue. Air quality prediction has drawn significant attention, bridging environmental science, statistics, and computer science. This paper presents a comprehensive review of the current research status and advances in air quality prediction methods. Deep learning, a novel machine learning approach, has demonstrated remarkable proficiency in identifying complex, nonlinear patterns in air quality data, yet its application in air quality prediction is still relatively nascent. This paper also conducts a systematic analysis and summarizes how cutting-edge deep learning models are applied in air quality prediction. Initially, the historical evolution of air quality prediction methods and datasets is presented. This is followed by an examination of conventional air quality prediction techniques. A thorough comparative analysis of progress made with both traditional and deep learning-based prediction methods is provided. This review particularly focuses on three aspects: temporal modeling, spatiotemporal modeling, and attention mechanisms. Finally, emerging trends in the field of air quality prediction are identified and discussed. Air quality forecasting relies on diverse data sources, including ground monitoring stations, meteorological satellites, and monitoring aircraft. However, these data sources contain pollutant samples that exhibit high spatial and temporal heterogeneity. This heterogeneity poses challenges in aggregating relevant data to achieve accurate predictions of pollutant changes. Among these data sources, ground air quality monitoring stations play a vital role because they are strategically positioned near areas with high pollutant emissions. This proximity allows them to serve as crucial data sources for studying air quality variations. Nevertheless, several challenges persist in the field of air quality prediction. First, despite some regions successfully establishing numerous regional air quality monitoring stations and even deploying interconnected micro air quality monitoring stations on streets, many areas still lack a comprehensive air quality monitoring system. This approach results in limited and sparse data samples in certain regions. Second, the generation of air pollutants involves complex chemical processes that exhibit significant temporal and spatial variations.

## 2.2 LITERATURE SURVEY

**2.2.1 A systematic survey of air quality prediction based on deep learning**

**Author Name :** Zhen Zhang, Shiqing Zhang

**Year :** 2018

Accurately predicting air quality presents significant challenges. Air quality forecasting relies on diverse data sources, including ground monitoring stations, meteorological satellites, and monitoring aircraft. However, these data sources contain pollutant samples that exhibit high spatial and temporal heterogeneity. This heterogeneity poses challenges in aggregating relevant data to achieve accurate predictions of pollutant changes. Among these data sources, ground air quality monitoring stations play a vital role because they are strategically positioned near areas with high pollutant emissions. This proximity allows them to serve as crucial data sources for studying air quality variations. Nevertheless, several challenges persist in the field of air quality prediction. First, despite some regions successfully establishing numerous regional air quality monitoring stations and even deploying interconnected micro air quality monitoring stations on streets, many areas still lack a comprehensive air quality monitoring system. This approach results in limited and sparse data samples in certain regions. Second, the generation of air pollutants involves complex chemical processes that exhibit significant temporal and spatial variations.

**2.2.2 A multi-graph spatial-temporal attention network for air-quality prediction**

**Author Name : Caimei Chen**, **Jiwei Yuan**

**Year :** 2019

Air pollution poses a grave threat to human health and everyday life. Accurate air-quality prediction plays a crucial role in effectively preventing and controlling air pollution. A multi- graph spatial-temporal attention network is proposed to predict the air quality in a given area by analyzing the interconnections between stations and the individual characteristics of each station. Firstly, this paper constructs multi-scale spatial-temporal graphs from spatial and temporal perspectives to effectively capture the interconnections between stations, thereby comprehensively understanding the spatial-temporal relationships among them. Secondly, incorporating the nonlinear temporal correlation of station data, this paper proposes a temporal multi-graph attention-fusion module to integrate information from both neighboring stations and the station itself. Finally, predictions are made using a spatial-temporal graph network. The experiments in this paper were based on air-quality data from Beijing and Tianjin, and the following experimental results were obtained. The rapid development of society has also resulted in some challenges. Emissions from factories, industrial activities, vehicle exhausts, and the burning of fossil fuels have led to severe air pollution.

**2.2.3 Spatial air quality prediction in urban areas via message passing**

**Author Name :** Sergio Calo, Filippo Bistaffa, Anders Jonsson, Vicenç Gómez,

Mar Viana

**Year :** 2021

Air pollution in urban areas poses a significant and pressing challenge for modern society. Unfortunately, the existing network of pollution detectors in many cities is limited in scope and fails to adequately cover the entire geographical area. Consequently, the implementation of spatial prediction algorithms becomes essential to generate high-resolution data. Air quality is a major concern in urban areas, as poor air quality can lead to a range of negative health outcomes such as respiratory and cardiovascular diseases. Accurate prediction of air quality is essential for mitigating these negative impacts and for developing effective strategies for improving air quality. However, predicting air quality in urban areas is a complex task, as it involves a range of factors such as meteorological conditions, traffic patterns, and emissions from various sources. Traditionally, air quality prediction has relied on models based on physical and chemical principles, which can be computationally expensive and may not accurately capture the complex interactions that occur in real-world urban environments. In recent years, there has been a growing interest in the use of graph-based approaches for predicting air quality in urban areas. These approaches are based on the idea that air quality can be represented as a graph, with nodes representing different locations in the city and edges representing some kind of interaction between these locations. In this paper, we present a new approach to air quality prediction that utilizes message passing in graphs. Graph-based models are a powerful tool for representing complex systems, as they can capture the structural relationships between different elements of the system. This graph-based approach allows the use of Graph Neural Network (GNN) methods

**2.3 LITREATURE REVIEW SUMMARY**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Reference** | **Author Name** | **Paper** | **Theme** | **Area of Examination** | **Algorithm** | **Result** |
| 1. | McNeill, V., Woo, J. L.,  Kim, D. D.,  Schwier, A., Wannell, Neal J., Sumner, Andrew J., Barakat, Joseph M. | Aqueous-phase secondary organic aerosol and organo-sulfate formation in atmospheric aerosols: a modeling study | Modeling secondary organic aerosols and organo- sulfates | Atmospheric aerosols | Modeling study | Provides insights into secondary organic aerosol formation processes, crucial for understanding air quality impacts. |
| 2. | Chen, Xiaojun, Liu, Xianpeng, Peng, Xu | IOT-based air pollution monitoring and forecasting system | IoT-based air quality monitoring and forecasting | Air pollution monitoring | IoT and forecasting system | Demonstrates an effective IoT-based system for real-time monitoring and forecasting of air pollution. |
| 3. | Wei, Wenjuan, Ramalho, O., Malingre, Laeticia, Sivanantham, S., Little, John C., Mandin, C. | Machine learning and statistical models for predicting indoor air quality | Indoor air quality prediction | Machine learning and statistical models | Compares different models for indoor air quality prediction, showing the effectiveness of machine | It deliver a practical and economical solution without any human intervention |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | learning techniques. |  |
| 4. | Ijaz, M., Ghassemlooy, Zabih, Pesek, J., Fiser, O., Minh, H. Le, Bentley, E. | Modeling of Fog and Smoke Attenuation in Free Space Optical Communications Link Under Controlled Laboratory Conditions | Fog and smoke attenuation in optical communications | Modeling of fog and smoke | Provides a model for attenuation effects due to fog and smoke, relevant for communication and air quality studies. | NA |
| 5. | Xie, Xingzhe, Šemanjski, I., Gautama, S.,  Tsiligianni, Evaggelia, Deligiannis, N., Rajan, R., Pasveer, F., Philips, Wilfried | Xie, Xingzhe, Šemanjski, I., Gautama, S., Tsiligianni, Evaggelia, Deligiannis, N., Rajan, R., Pasveer, F., Philips, Wilfried | Xie, Xingzhe, Šemanjski, I., Gautama, S., Tsiligianni, Evaggelia, Deligiannis, N., Rajan, R., Pasveer, F., Philips, Wilfried | Xie, Xingzhe, Šemanjski, I., Gautama, S., Tsiligianni, Evaggelia, Deligiannis, N., Rajan, R., Pasveer, F., Philips, Wilfried | Xie, Xingzhe, Šemanjski, I., Gautama, S., Tsiligianni, Evaggelia, Deligiannis, N., Rajan, R., Pasveer, F., Philips, Wilfried | NA |
| 6. | Tsai, Yi-Ting, Zeng, Yu-Ren, Chang, Yue- Shan | Air Pollution Forecasting Using RNN with LSTM | Air pollution forecasting | Recurrent Neural Networks with Long Short-Term Memory (LSTM) | Utilizes RNN with LSTM for effective forecasting of air pollution levels. | NA |
| 7. | Ameer, S.,  Shah, M. A., Khan, Abid, Song, Houbing, Maple, C., Islam, Saif ul., Asghar, | Comparative Analysis of Machine Learning Techniques for Predicting Air Quality in Smart Cities | Machine learning techniques for air quality prediction | Machine learning techniques | Compares various machine learning techniques for predicting air quality, highlighting their | NA |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8. | Conti, G. Oliveri, Heibati, B., Kloog, I.,  Fiore, M., Ferrante, M. | A review of AirQ Models and their applications for forecasting the air pollution health outcomes | AirQ Models for health outcomes | Air pollution forecasting | Reviews AirQ models used for predicting health outcomes related to air pollution. | NA |
| 9. | Nickovic, S., Vukovic, A., Vujadinovic, M.,  Djurdjević, V.,  Pejanovic, G. | Technical Note: High-resolution mineralogical database of dust- productive soils for atmospheric dust modeling | Atmospheric dust modeling | High- resolution mineralogical database | Provides a database for modeling atmospheric dust, relevant for air quality studies involving dust particles. | NA |
| 10. | Metry, Morcos, Shadmand, M., Balog, R., Abu-Rub, H. | MPPT of Photovoltaic Systems Using Sensorless Current-Based Model Predictive Control | MPPT in photovoltaic systems | Model Predictive Control | Focuses on optimizing photovoltaic systems, indirectly related to air quality through energy efficiency. | NA |

**Chapter 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

Traditional air quality monitoring systems often rely on fixed monitoring stations that measure pollutant levels at specific locations. These stations are typically equipped with sensors to detect various pollutants such as particulate matter, nitrogen oxides, sulfur - di- oxide, and volatile organic compounds.

Limited Coverage in fixed monitoring stations may not provide adequate coverage of the entire area, especially in densely populated or geographically diverse regions. This can lead to gaps in air quality data and hinder the ability to identify pollution hotspots or track the dispersion of pollutants. Delayed Updates in data collected from fixed monitoring stations may be subject to delays in processing and dissemination, limiting their real-time value for decision-making. High Costs for installation and maintenance of fixed monitoring stations can be expensive, making it challenging to establish comprehensive networks in many areas. In recent years, advancements in Internet of Things (IoT) technology and sensor miniaturization have enabled the development of more flexible and cost-effective air quality monitoring solutions. These systems utilize a network of low-cost sensors that can be deployed in various locations, providing denser coverage and real-time data. However, these systems may face challenges related to data quality, sensor calibration, and integration with existing infrastructure.

**3.2** **DISADVANTAGES OF EXISTING SYSTEM**

* **Limited Coverage:** Fixed monitoring stations may not provide adequate coverage of the entire area, leading to gaps in air quality data and hindering the ability to identify pollution hotspots or track the dispersion of pollutants.
* **Delayed Updates:** Data collected from fixed monitoring stations may be subject to delays in processing and dissemination, limiting their real-time value for decision-making.
* **High Costs:** Installation and maintenance of fixed monitoring stations can be expensive, making it challenging to establish comprehensive networks in many areas.
* **Reliance on Models:** Some existing air quality alert systems rely on models and algorithms to predict air quality levels, which may not always accurately capture the complex interactions between pollutants and environmental conditions.

**3.3 PROPOSED SYSTEM**

A key component of the Air Quality Alert System is the user interface, designed to be accessible via both web and mobile platforms. Users will receive real-time notifications and alerts about current air quality conditions in their area, along with recommended actions such as limiting outdoor activities or wearing masks. Additionally, the system will feature visualization tools that display AQI trends over time, helping users understand pollution patterns and their potential health impacts. The platform will also include an API for third- party developers and organizations, allowing them to integrate air quality data into their own applications or research projects. Finally, the project aims to support policy-makers and environmental agencies by providing valuable insights into pollution trends and sources. The system can assist in identifying pollution hotspots, seasonal variations, and the effectiveness of pollution control measures. With its comprehensive data analytics capabilities, the Air Quality Alert System can contribute to long-term strategies for improving air quality, reducing public health risks, and ensuring a cleaner environment for future generations. By covering real-time monitoring, accurate prediction, and user engagement, the system has the potential to become a crucial tool in the fight against air pollution.

## 3.4 ADVANTAGES OF PROPOSED SYSTEM

* **Real-time Information:** The system provides real-time notifications and alerts about current air quality conditions, allowing users to make informed decisions about their activities.
* **User-Friendly Interface:** The user interface is accessible via both web and mobile platforms, making it convenient for users to access information and take action.
* **Data-Driven Insights:** The system utilizes data analytics to identify pollution hotspots, seasonal variations, and the effectiveness of pollution control measures, providing valuable insights for policymakers and environmental agencies.
* **API Integration:** The system provides an API for third-party developers and organizations, allowing them to integrate air quality data into their own applications or research projects, expanding the reach and impact of the system.

**3.5 FEASIBILITY STUDY**

A feasibility study for an Air Quality Alert System is essential to assess its technical, economic, and operational viability. Key considerations include the availability and reliability of air quality sensors, real-time data transmission capabilities, computational resources for data processing and analysis, and effective alert dissemination channels. Economic factors encompass initial investment costs, ongoing operational expenses, and potential cost-benefits. Legal and regulatory compliance, data privacy, public acceptance, and environmental impact are also crucial aspects to evaluate. By carefully considering these factors, a comprehensive feasibility study can inform decision-making and ensure the successful implementation of an Air Quality Alert System.

* 1. **HARDWARE ENVIRONMENT**
* Processor : Pentium Dual Core 2.00GH
* Hard disk : 120 GB
* RAM : 2GB (minimum)
* Keyboard : 110 keys enhanced
  1. **SOFTWARE ENVIRONMENT**
* Operating system : Windows7 (with service pack 1), 8, 8.1 ,10 and 11
* Language : Python
  1. **TECHNOLOGIES USED**
* IDE - Visual Studio, Google Colab, Jupyter Notebook
* Framework - Stream-lit
* Machine Learning
  + 1. **Python**

Python is a high-level, interpreted programming language that is widely used in various domains such as web development, data science, artificial intelligence, scientific computing, and more. It was first released in 1991 and has since become one of the most popular programming languages in the world. Some key features of Python include:

* Easy to Learn: Python has a simple and easy-to-learn syntax, which makes it an ideal language for beginners.
* Interpreted Language: Python is an interpreted language, which means that the code is executed line by line, making it easier to test and debug.
* Cross-Platform: Python can be run on various platforms, including Windows, macOS, and Linux.
* Large Standard Library: Python has a large standard library that provides a wide range of built-in modules for various tasks, such as file I/O, regular expressions, networking, and more.
* Open Source: Python is open-source software, which means that the source code is freely available to anyone and can be modified and redistributed.
* Object-Oriented: Python is an object-oriented language, which means that it supports object-oriented programming concepts such as encapsulation, inheritance, and polymorphism

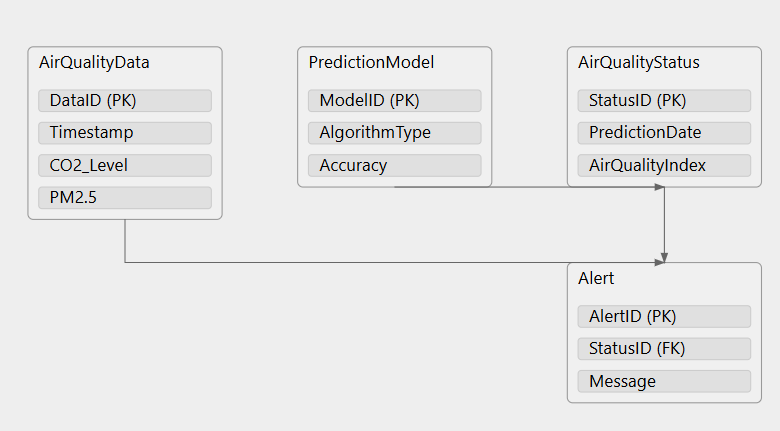
**3.9.2 Machine Learning**

Machine learning algorithms like Random Forest and XG Boost are powerful tools for predicting air quality. Random Forest, an ensemble learning method, creates multiple decision trees and averages their predictions. This approach reduces overfitting and improves accuracy. XG Boost, on the other hand, is a gradient boosting algorithm that iteratively adds decision trees to the model, focusing on correcting errors made by previous trees. This leads to highly accurate and robust predictions. By leveraging historical air quality data, meteorological factors, and other relevant features, these algorithms can effectively model complex patterns and dependencies. This enables the system to accurately forecast air quality levels, identify trends, and issue timely alerts to the public. Additionally, these models can help identify the most significant factors contributing to air pollution, aiding in targeted interventions and policy-making. Random Forest operates by constructing multiple decision trees, each trained on a random subset of the data. The final prediction is made by averaging the predictions of all trees, reducing overfitting and improving generalization. XG Boost, on the other hand, is a gradient boosting algorithm that iteratively adds decision trees to the model, focusing on correcting errors made by previous trees. This approach leads to highly accurate and robust predictions.

# Chapter 4 SYSTEM DESIGN

#### 4.1 ENTITY-RELATIONSHIP DIAGRAM

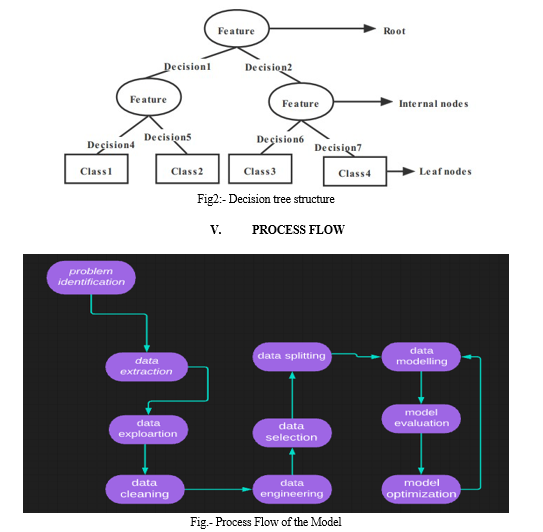
The relationships between database entities can be seen using an entity- relationship diagram (ERD). The entities and relationships depicted in an ERD can have further detail added to them via data object descriptions. In software engineering, conceptual and abstract data descriptions are represented via entity- relationship models (ERMs). Entity-relationship diagrams (ERDs), entity- relationship diagrams (ER), or simply entity diagrams are the terms used to describe the resulting visual representations of data structures that contain relationships between entities. As such, a data flow diagram can serve dual purposes. To demonstrate how data is transformed across the system. To provide an example of the procedures that affect the data flow.

****

**Fig 4.1 Entity Relationship Diagram for Air Quality Prediction and Alert System**

* 1. **DATA FLOW DIAGRAM (DFD)**

The whole system is shown as a single process in a level DFD. Each step in the system's assembly process, including all intermediate steps, are recorded here. The "basic system model" consists of this and 2-level data flow diagrams. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM). Superficially, DFDs can resemble flow charts or Unified Modeling Language (UML), but they are not meant to represent details of software logic. DFDs make it easy to depict the business requirements of applications by representing the sequence of process steps and flow of information using a graphical representation or visual representation rather than a textual description.



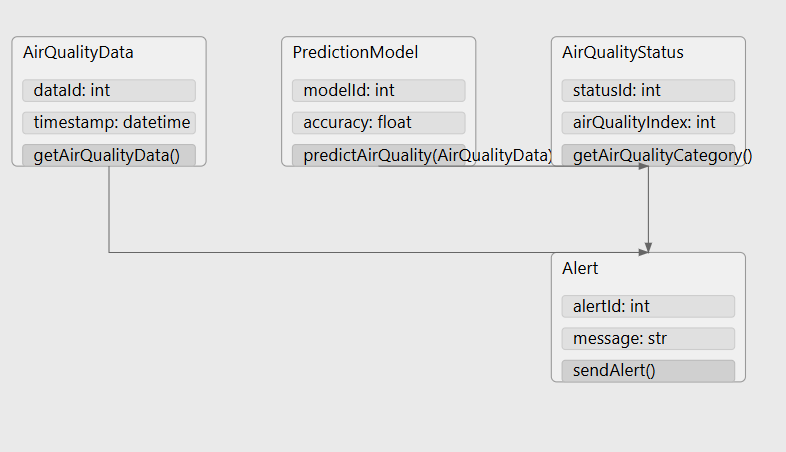
**Fig 4.1.1 Data Flow Diagram for Air Quality Prediction and Alert System**

#### UML DIAGRAMS

* + 1. **Use Case Diagram**

A use case diagram is a type of Unified Modeling Language (UML) diagram that represents the interactions between a system and its actors, and the various use cases that the system supports. It is a visual representation of the functional requirements of the system and the actors that interact with it. Use case diagrams typically include the following elements:

* Actors: Actors are external entities that interact with the system. They can be human users, other systems, or devices.
* Use Cases: Use cases are the specific functions or tasks that the system can perform. Each use case represents a specific interaction between an actor and the system.
* Relationships: Relationships are used to indicate how the actors and use cases are related to each other. The two main relationships in a use case diagram are "uses" and "extends". "Uses" relationship indicates that an actor uses a specific use case, while "extends" relationship indicates that a use case extends or adds functionality to another use case.
* System Boundary: The system boundary is a box that contains all the actors and use cases in the system.



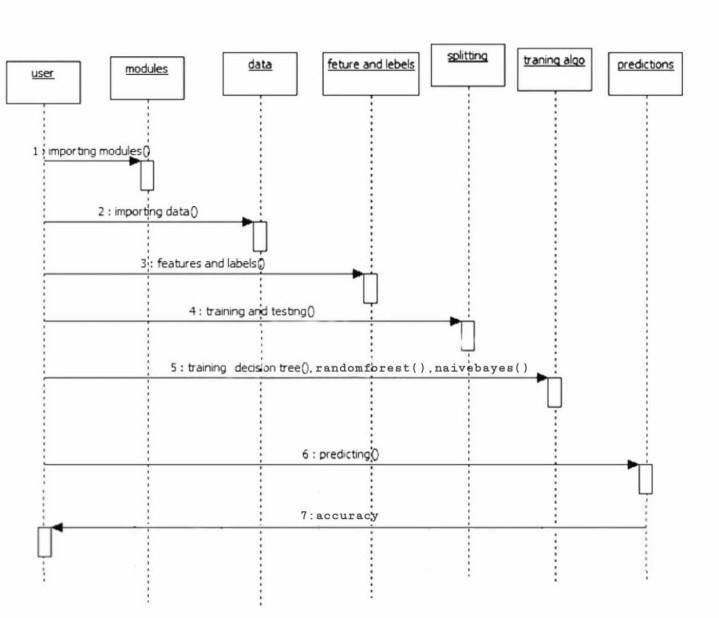
#### Class Diagram

#### The diagram illustrates a simplified machine learning application. It shows a user interacting with a home class, which in turn loads data and triggers predictions. The prediction class, along with a validation class, ensures accurate results. This diagram provides a basic overview of the system's structure and data flow. The diagram illustrates a simplified machine learning application. It depicts the interaction between the user, the home class, the data class, the prediction class, and the validation class. The user initiates the process by providing data to the home class, which then loads and processes the data. The prediction class utilizes the validation class to ensure accurate predictions. This collaborative effort results in the generation of final predictions.

#### 

#### Sequence Diagram

These are another type of interaction-based diagram used to display the workings of the system. They record the conditions under which objects and processes cooperate. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



# Chapter 5

**SYSTEM ARCHITECTURE**

## ARCHITECTURE DIAGRAM

The architecture diagram illustrates a machine learning workflow starting with raw data and ending with a model ready for user validation. The process begins with a File (Dataset) containing raw data or records, which is then sent to the Data Preprocessing stage. Here, the data is cleaned, standardized, and any inconsistencies or missing values are addressed, improving the quality of the input for modelling. Once pre-processed, the data moves to Data Preparation, where it undergoes further transformations, such as splitting into training and testing sets and applying feature engineering techniques. This stage ensures the data is in the best possible format for machine learning.

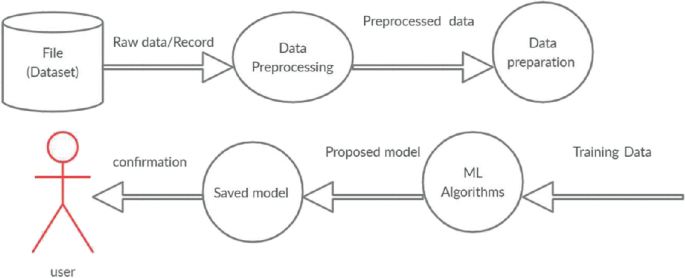


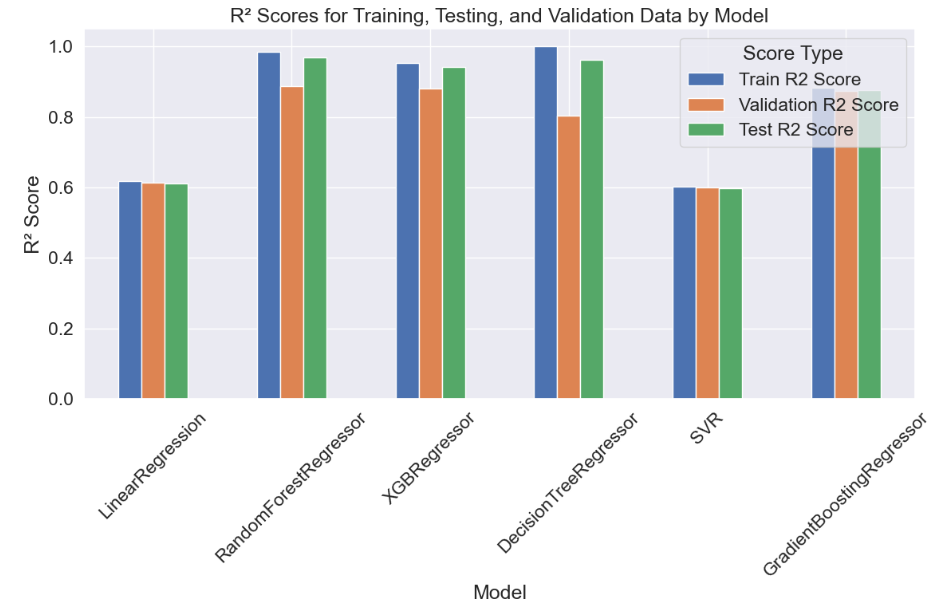
Fig 5.1 Architecture Diagram

The prepared data is then fed into Machine Learning Algorithms to generate a Proposed Model. The choice of algorithms depends on the project goals, whether it involves regression, classification, or another type of analysis. After training and optimization, the model is saved in the Saved Model stage, making it available for future use or deployment. At this point, a User interacts with the system to evaluate the model's performance. This user validation provides a Confirmation step, where the user may test the model’s effectiveness and decide if further tuning is necessary.

#### ALGORITHMS

* + 1. **XGB Regressor**

The XGB Regressor is a machine learning algorithm from the XG Boost (Extreme Gradient Boosting) library, widely used for regression tasks. Built on the principles of gradient boosting, XGB Regressor constructs an ensemble of decision trees in a sequential manner, where each new tree aims to correct errors made by the previous ones. Unlike traditional boosting methods, XG Boost is optimized for speed and performance through various techniques like parallel processing, handling sparse data, and regularization, making it highly efficient for large datasets. One of the key strengths of XGB Regressor is its ability to handle both numerical and categorical features, thanks to built-in mechanisms like missing value handling and feature importance ranking. It minimizes the loss function iteratively, typically using mean squared error (MSE) as the loss metric for regression tasks. Each iteration adds a new tree that focuses on reducing the residuals, or the difference between the actual and predicted values, gradually improving the model’s accuracy.



#### Fig 5.2.1 XGB Regressor Air Quality Prediction Model

# Chapter 6

**SYSTEM IMPLEMENTATION**

## MODULE 1: DATA COLLECTION AND PREPROCESSING

Air quality prediction is a crucial application of data science, especially given its implications for public health. Accurate predictions rely on robust data collection and preprocessing steps, which transform raw data into a refined form suitable for analysis and modelling. For this project, data is gathered from multiple sources, including air quality monitoring APIs, publicly available datasets, and historical air quality records. APIs such as World Air Quality Index provide real-time data on pollutants like PM2.5, PM10, CO, NO2, SO2, and O3. Additionally, meteorological factors like temperature, humidity, wind speed, and atmospheric pressure are collected since they influence pollutant levels. Collecting data from multiple sources ensures a comprehensive dataset that can enhance the model’s accuracy. Once data is collected, preprocessing begins with data cleaning. This involves handling missing values, which can arise from sensor malfunctions or data transmission issues. Missing values are either filled with statistical measures, such as the median or mean, or removed if they constitute a small percentage of the dataset. Duplicate records are identified and removed to avoid redundant data, while data points with extreme outliers are either removed or adjusted based on domain knowledge, as extreme values can skew model performance. Outlier detection techniques, such as z-scores and IQR (interquartile range), are used to ensure the data remains within a realistic range. Next, feature engineering is applied to create new variables that may improve predictive performance. For instance, calculating moving averages of pollutant levels over time or incorporating lagged features helps the model capture trends in air quality fluctuations. Other engineered features might include calculating pollutant ratios or seasonal indicators, which can provide additional insights into pollution levels. Data transformation is another key preprocessing step that prepares the data for machine learning algorithms. Numerical data is often scaled using methods such as Min-Max Scaling or Standardization to ensure that features are on a comparable scale, which is particularly important for distance-based models. For categorical data, encoding techniques like one-hot encoding or label encoding are used to convert text-based information into numeric format. Normalization of data helps achieve a consistent distribution, further improving model accuracy

.

## MODULE 2: MODEL TRAINING

The next step in building the water quality prediction application is model development and training. In this air quality prediction project, the goal is to train a model that accurately forecasts air pollutant levels and can send alert messages when poor air quality is anticipated. After preprocessing the data, we proceed with model training using the XG Boost (XGB) regressor, a highly efficient and accurate machine learning algorithm based on gradient boosting. XG Boost combines the outputs of multiple weak learners, typically decision trees, to form a robust predictive model by iteratively correcting the errors of previous models. It is well-suited for structured tabular data like air quality data and offers excellent computational efficiency, allowing it to handle large datasets and multiple features efficiently. XG Boost also includes built-in regularization techniques (L1 and L2) that prevent overfitting, a key advantage when modeling complex data patterns that may be present in air quality datasets. The process begins by configuring essential hyperparameters. For instance, n\_estimators determines the number of boosting rounds, with a higher number generally leading to more accurate predictions at the risk of overfitting. The learning\_rate controls the step size for each iteration, with lower values helping the model achieve stability and fine-tuned accuracy. The max\_depth parameter limits the depth of each tree, balancing the model’s ability to capture intricate patterns against the potential for overfitting. Additional parameters, such as subsample (which controls the fraction of samples used in each boosting round) are set to improve model generalization and efficiency. These hyperparameters are fine-tuned using grid search or random search methods, which systematically explore parameter combinations to find the optimal setup. Once configured, the model is trained on the processed dataset, typically split into training and validation subsets. During training, XG Boost evaluates each tree’s performance against the target pollutant levels, adjusting subsequent trees to minimize errors. This iterative refinement continues over the specified number of rounds, producing a model that captures pollutant trends effectively. Performance metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are calculated to assess prediction accuracy, with lower scores indicating better alignment with actual air quality levels. After validating the model’s performance on the test set, it is deployed to predict future air quality conditions. When pollutant levels exceed safe thresholds, the system triggers alert messages, providing timely warnings to users. Overall, using the XGB regressor with tuned hyperparameters allows for accurate, efficient, and scalable air quality prediction, supporting proactive health interventions through timely alerts.

## MODULE 3: PREDICTION OF OUTPUT

In the air quality prediction project, the final phase involves using the trained XG Boost (XGB) regressor model to make accurate predictions on future air pollutant levels and trigger alert messages when necessary. After the model training and validation steps, the XGB regressor is applied to real-time or new data, which may include pollutant measurements and environmental factors like temperature, humidity, wind speed, and atmospheric pressure. Using this input data, the model predicts concentrations of harmful pollutants such as PM2.5, PM10, CO, NO2, SO2, and O3 for specified future time intervals, helping to anticipate variations in air quality based on patterns learned from historical data. The XG Boost model’s predictive power enables it to generate highly accurate results due to its boosted tree-based ensemble structure, which fine-tunes prediction by minimizing errors from prior iterations. Once pollutant levels are predicted, the model’s output is compared against pre-defined thresholds for safe air quality, often based on Air Quality Index (AQI) standards. If predicted pollutant concentrations exceed safe limits, indicating poor or hazardous air quality, the system automatically generates alert messages. These alerts can be configured to include specific warnings, such as advising people with respiratory conditions or sensitivities to avoid outdoor activities. For instance, if the model forecasts PM2.5 levels above 100 µg/m³, an alert can be sent to indicate “Unhealthy” air quality, encouraging actions such as staying indoors or wearing masks. Alerts can be delivered through various channels like SMS, emails, or app notifications to ensure timely dissemination. The prediction and alert system is evaluated continuously to maintain accuracy and reliability. Regular updates to the model, incorporating new data, help refine its predictive capability and adapt to emerging trends in pollutant behavior. Monitoring performance metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) on fresh data ensures that the model retains precision over time. With this structure, the air quality prediction and alert system serve as a proactive health tool, empowering individuals and communities to make informed decisions and reduce exposure to harmful pollutants. This comprehensive prediction-to-alert mechanism underscores the practical value of the XG Boost regressor in real-world applications, contributing to public health and environmental awareness.

# Chapter 7 SYSTEM TESTING

#### BLACK BOX TESTING

#### The user is not given access to or knowledge of the internal workings or details of the data item under test during this type of testing. This approach does not require prior knowledge of the design or the code; instead, test cases are created or constructed only based on the input and output values. The testers are just aware of what is believed to be possible; they are unaware of how it accomplishes this. For instance, we test the web pages using a browser, authorize the input, and then test and validate the outputs against the desired outcome without knowing anything about the inner workings of the website.



Fig 7.1 Black Box Testing

For example, without having any knowledge of the inner workings of the website, we test the web pages by using a browser, then we authorize the input, and last, we test and validate the outputs against the intended result.

#### WHITE BOX TESTING

During this kind of testing, the user is aware of the internal structure and details of the data item, or they have access to such information. In this process, test cases are constructed by referring to the code. Programming is extremely knowledgeable of the manner in which the application of knowledge is significant. White Box Testing is so called because, as we all know, in the tester's eyes it appears to be a white box, and on the inside, everyone can see clearly. This is how the testing got its name.

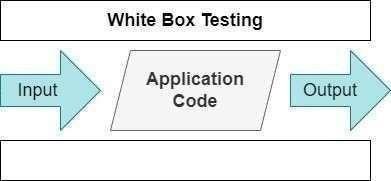


Fig 7.2 White Box Testing

As an instance, a tester and a developer examine the code that is implemented in each field of a website, determine which inputs are acceptable and which are not, and then check the output to ensure it produces the desired result. In addition, the decision is reached by analyzing the code that is really used.

* 1. **TEST CASES TEST REPORT: 01**

**PRODUCT:**  WATE QUALITY PREDICTION BY A MODEL AND DEPLOYING IT AS A WEB APPLICATION

**USE CASE:** UPLOAD DATA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Action** | **Input** | **Expected output** | **Actual output** | **Test Result** |
| 1 | Verify data retrieval from water quality sample | Location: River A | Display current water quality metrics for River A | [Actual metrics] for River A | Pass/Fail |
| 2 | Test prediction model accuracy | Historical water quality data | Prediction within 10% of actual water quality values | Prediction accuracy [actual %] | Pass/Fail |
| 3 | Check alert generation for unsafe water conditions | pH: 3.5, Turbidity: High | System sends alert for unsafe water conditions | Alert message sent | Pass/Fail |
| 4 | Ensure no alert for safe water conditions | pH: 7.2, Turbidity: Low | No alert is sent | No alert generated | Pass/Fail |
| 5 | Verify model deployment on web application | Access URL | Web app loads successfully with prediction model | Web app loaded [actual status] | Pass/Fail |

**Table-7.3.1 Test Case Dataset Upload**

## TEST REPORT: 02

**PRODUCT:**  WATE QUALITY PREDICTION BY A MODEL AND DEPLOYING IT AS A WEB APPLICATION

**USE CASE:** PREDICTING DATA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | Action | Input | Expected output | Actual output | Test Result |
| 1 | Test prediction request through web app | User inputs water quality metrics | Model provides water quality prediction | Prediction displayed [actual output] | Pass/Fail |
| 2 | Check response time of prediction model on web app | User submits input via web form | Prediction response time within 3 seconds | Response time [actual time] | Pass/Fail |
| 3 | Verify user interface for displaying predictions | Open web app | Display predictions in readable format on dashboard | Predictions displayed correctly | Pass/Fail |
| 4 | Test alert customization for different water types | User preferences: Custom thresholds | Alert generated based on user-defined thresholds | Alert generated as per user threshold | Pass/Fail |
| 50 | Ensure real-time data integration on web app | Real-time water quality data feed | Data updated every 5 minutes | Data updated every [actual interval] | Pass/Fail |

**Table-7.3.2 Test Case for Predicting Result**

# Chapter 8

**CONCLUSION AND FUTURE ENHANCEMENT**

## CONCLUSION

In conclusion, this project successfully demonstrated the use of machine learning, particularly the XGBoost (XGB) regressor, for predicting air quality levels and sending timely alert messages to help mitigate the impact of poor air quality on public health. With increasing levels of air pollution globally, the need for systems that can forecast air quality and provide early warnings has become critical. This project addressed this challenge by building a predictive model capable of forecasting air pollution levels in the coming hours or days based on historical data and real-time environmental factors. The integration of alert messaging adds an extra layer of safety, ensuring that individuals can take timely actions to protect their health, especially those with respiratory conditions or other vulnerabilities. The core of this project was the implementation of the XG Boost regressor, which was chosen for its high accuracy, computational efficiency, and ability to handle large, complex datasets. The model was trained on data collected from various sensors, APIs, and publicly available datasets, which included a wide range of features such as PM2.5, PM10, CO, NO2, SO2, and O3 concentrations, as well as meteorological factors like temperature, humidity, and wind speed. Through extensive preprocessing, including data cleaning, handling missing values, feature engineering, and transformation, we ensured that the data was in the best possible form to train a high-performing model. The XG Boost model demonstrated strong predictive power by minimizing prediction errors and providing accurate air quality forecasts. Key parameters of the model, such as the number of estimators, learning rate, and maximum depth, were tuned to ensure optimal performance. The model was evaluated using various performance metrics, including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which showed satisfactory results in predicting pollutant levels. Additionally, the system was able to generate accurate alerts based on air quality thresholds, which could be customized for different regions or health advisories. One of the key outcomes of this project was the development of an alert system that automatically triggers notifications when air quality levels exceed safe thresholds. This feature is crucial for helping people stay informed about the air they breathe, especially in cities or regions with frequent pollution spikes.

## FUTURE ENCHANCEMENT

While the current air quality prediction model using the XG Boost regressor and alert system provides a solid foundation for forecasting pollutant levels and issuing timely alerts, there are numerous opportunities for future enhancements that can improve its accuracy, adaptability, and user experience. One primary enhancement is the refinement of the predictive model itself. Although the XG Boost regressor performs well, integrating other machine learning models, such as Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs), could improve the model’s ability to capture time-series dependencies and long-term patterns in air quality data. Additionally, combining multiple models in an ensemble method could further increase prediction accuracy by mitigating individual model weaknesses. Further, implementing techniques like feature selection and dimensionality reduction could help eliminate irrelevant variables, reduce overfitting, and increase the generalization power of the model. Incorporating additional data sources will be another key area for improvement. While the current system primarily uses sensor and meteorological data, integrating satellite imagery, real-time environmental factors, and social media data could enhance the predictive capabilities of the model. Satellite data could offer a global perspective on air pollution patterns, while social media data, such as tweets or posts about air quality, could provide real-time, crowd-sourced insights. Additionally, including regional and seasonal variations in air quality, such as the impact of seasonal weather changes or local environmental factors like nearby factories, would help the model generate more accurate predictions for different geographical locations. Another crucial area for enhancement is the incorporation of real-time data integration and edge computing. Real-time data pipelines could allow the system to process live sensor data and generate immediate predictions, offering more timely alerts as pollutant levels fluctuate. Implementing edge computing—where data processing occurs closer to the data source, such as on sensors or IoT devices—would reduce latency and computational load, enabling faster predictions and more responsive alerts. This would be especially useful in scenarios where immediate action is required, such as during a sudden spike in pollution levels. User experience can also be significantly improved by allowing for personalized alert preferences. Currently, users receive generic alerts when pollutant levels exceed predefined thresholds. In the future, users could set personalized thresholds based on their health conditions or preferences, such as receiving more urgent alerts if they have respiratory issues.

# Chapter 9

**APPENDIX 1 – SAMPLE CODING**

**AQAS.ipynb**

import pandas as pd

import numpy as np

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

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

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.svm import SVR

from sklearn.tree import DecisionTreeRegressor, plot\_tree

from sklearn.ensemble import RandomForestRegressor

from XG Boost import XGBRegressor

from sklearn.tree import DecisionTreeRegressor

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

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv(‘city\_day.csv’)

df

df.isnull().sum()

df.dropna(how='all', inplace=True)

df.info()

df.Potability.value\_counts()

dist=(df['City'])

distset=set(dist)

dd=list(distset)

dictofwords = { dd[i]: i for i in range(0, len(dd)) }

df['City']=df['City'].map(dictofwords)

dist=(df['AQI\_Bucket'])

distset=set(dist)

dd=list(distset)

dictofwords = { dd[i]: i for i in range(0, len(dd)) }

df['AQI\_Bucket']=df['AQI\_Bucket'].map(dictofwords)

df.fillna(df['PM2.5'].mean(),inplace = True)

df.fillna(df['PM10'].mean(),inplace = True)

df.fillna(df['NO'].mean(),inplace = True)

df.fillna(df['NO2'].mean(),inplace = True)

df.fillna(df['NOx'].mean(),inplace = True)

df.fillna(df['NH3'].mean(),inplace = True)

df.fillna(df['CO'].mean(),inplace = True)

df.fillna(df['SO2'].mean(),inplace = True)

df.fillna(df['O3'].mean(),inplace = True)

df.fillna(df['Benzene'].mean(),inplace = True)

df.fillna(df['Toluene'].mean(),inplace = True)

df.fillna(df['AQI'].mean(),inplace = True)

city\_wise\_AQI = df[['City','AQI']].groupby(['City']).median().sort\_values(['AQI']).reset\_index()

plt.figure(figsize=(15,5))

sns.set(font\_scale=1.5)

sns.barplot(x='City', y='AQI', data=city\_wise\_AQI).set(title ='Cities & AQI')

plt.xticks(rotation=90)

plt.show()

sns.countplot(x=df["AQI\_Bucket"]);

plt.figure(figsize=(15, 5))

sns.boxplot(data=df[[ 'AQI','PM2.5', 'PM10','NO', 'NO2', 'NOx','NH3','O3', 'CO', 'SO2','Benzene', 'Toluene', 'Xylene']],width=0.6)

plt.show()

# Adjust the plot size

plt.figure(figsize=(15, 5))

# Set the float columns for boxplot

sns.boxplot(data=df[[ 'AQI','PM2.5', 'PM10','NO', 'NO2', 'NOx','NH3','O3', 'CO', 'SO2','Benzene', 'Toluene', 'Xylene']],width=0.6)

# Display

plt.show()

plt.figure(figsize=(16, 8))

sns.heatmap(df.corr(),annot=True)

plt.show()

# For Ilustration use the original data and for

# most effective gases for AQI

cols = ['AQI', 'PM2.5', 'PM10', 'CO', 'NO', 'NO2']

# create a pie chart

cmap = plt.get\_cmap('Spectral')

color = [cmap(i) for i in np.linspace(0, 1, 8)]

explode = [0.2, 0, 0, 0, 0, 0, 0, 0]

# fit multiple charts in the display

fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15, 8), dpi=300)

axes = axes.flatten()

# loop around all gasses for each city

for ax, col in zip(axes, cols):

# Group the cities and select the 8 cities with the highest total

x = df\_city\_day.groupby('City')[col].sum().sort\_values(ascending=False)

x = x.reset\_index('City')

top\_cities = x[:8]

sizes = top\_cities[col].values

labels = top\_cities['City'].tolist()

wedges, texts, autotexts = ax.pie(sizes, shadow=True, autopct='%1.1f%%',

colors=color, explode=explode,

wedgeprops={'edgecolor': 'black', 'linewidth': 0.3},

labels=labels)

for text in texts:

text.set\_fontsize(8)

for autotext in autotexts:

autotext.set\_fontsize(8)

ax.set\_title(f'{col}')

for i in range(len(cols), len(axes)):

fig.delaxes(axes[i])

plt.tight\_layout()

plt.show()

def replace\_outliers\_with\_quartiles(df):

for column in df.select\_dtypes(include=['number']).columns

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

df[column] = df[column].apply(

lambda x: Q1 if x < lower\_bound else (Q3 if x > upper\_bound else x)

)

return df

df = replace\_outliers\_with\_quartiles(df)

df

train\_and\_val\_df, test\_df = train\_test\_split(df\_full, test\_size=0.2, random\_state=42)

train\_df, val\_df = train\_test\_split(df\_full, test\_size=0.2)

print('train\_df.shape :', train\_df.shape)

print('val\_df.shape :', val\_df.shape)

print('test\_df.shape :', test\_df.shape)

results = []

models = {

'LinearRegression': LinearRegression(),

'RandomForestRegressor' : RandomForestRegressor(),

'XGBRegressor' : XGBRegressor(),

'DecisionTreeRegressor' : DecisionTreeRegressor(),

'SVR' :SVR(kernel='linear'),

'GradientBoostingRegressor': GradientBoostingRegressor(random\_state=42)

}

def try\_model(model,name):

model.fit(X\_train, train\_target)

train\_preds = model.predict(X\_train)

val\_preds = model.predict(X\_val)

test\_preds=model.predict(X\_test)

train\_r2\_score = r2\_score(train\_target, train\_preds)

val\_r2\_score= r2\_score(val\_target, val\_preds)

test\_r2\_score = r2\_score(test\_target, test\_preds)

print(f"{name} Model:")

print("Train r2\_score : ", train\_r2\_score)

print("Validation r2\_score : ", val\_r2\_score)

print("Test r2\_score : ", test\_r2\_score)

print("-" \* 40)

results.append({'Model': name, 'Train R2 Score': train\_r2\_score, 'Validation R2 Score': val\_r2\_score, 'Test R2 Score': test\_r2\_score})

for name, model in models.items():

try\_model(model,name)

results\_df = pd.DataFrame(results)

results\_df

results\_df.set\_index('Model').plot(kind='bar', figsize=(12, 8))

plt.title('R² Scores for Training, Testing, and Validation Data by Model')

plt.xlabel('Model')

plt.ylabel('R² Score')

plt.legend(title='Score Type')

plt.xticks(rotation=45)

plt.tight\_layout() # Ensures that all elements (axis labels, titles, subtitles, etc.) are properly placed within the figure area.

plt.show()

results\_df.set\_index('Model').plot(kind='bar', figsize=(12, 8))

plt.title('R² Scores for Training, Testing, and Validation Data by Model')

plt.xlabel('Model')

plt.ylabel('R² Score')

plt.legend(title='Score Type')

plt.xticks(rotation=45)

plt.tight\_layout() # Ensures that all elements (axis labels, titles, subtitles, etc.) are properly placed within the figure area.

plt.show()

import twilio.rest

from twilio.rest import Client

account\_sid = “ "

auth\_token = “ "

from\_phone = “ "

to\_phone = “ "

client = Client(account\_sid, auth\_token)

AQI\_THRESHOLD = 50

low\_aqi\_cities = df[df['AQI'] < AQI\_THRESHOLD]['City'].unique()

if low\_aqi\_cities.size > 0:

city = low\_aqi\_cities[0] # Get the first city with low AQI

aqi\_value = df[df['City'] == city]['AQI'].min() # Get the lowest AQI value for this city

message = f"Air quality alert for {city}. AQI is currently {aqi\_value}. Take necessary precautions."

client.messages.create(

body=message,

from\_=from\_phone,

to=to\_phone

)

print(f"Sent alert for {city} with AQI {aqi\_value}")

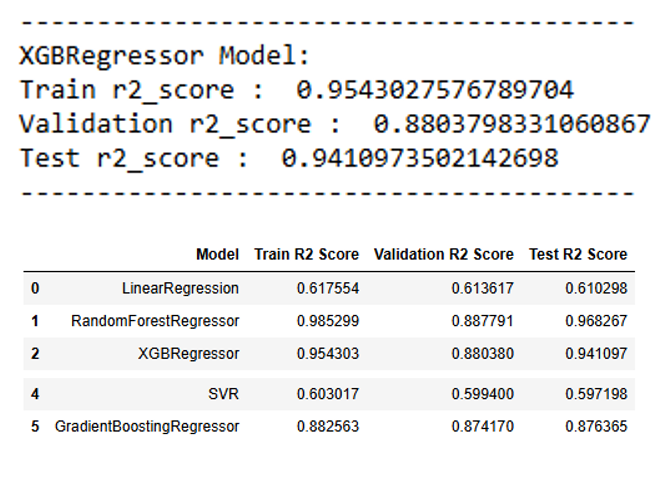
else:

print("No cities with low AQI detected.")

# Chapter 10

# APPENDIX 2 – SAMPLE OUTPUT

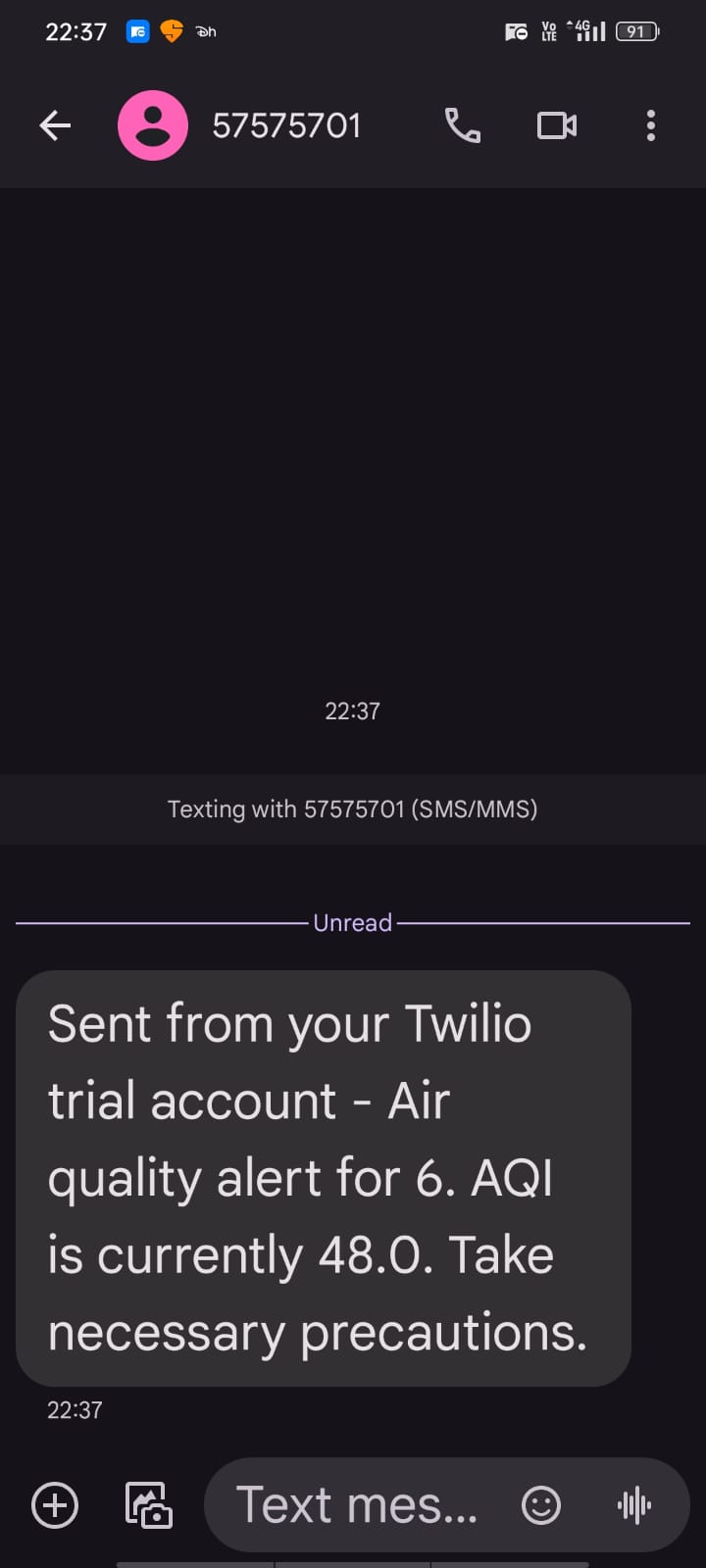
#### 10. 1 R2 Score Performance



#### R2 score performance for the given dataset

## The image provides a summary of the R-squared (R²) performance metrics for various regression models, focusing on the XGBRegressor and comparing it to other models. At the top, the XGBRegressor’s performance is shown with a train R² score of 0.954, indicating it has effectively learned the patterns in the training data, while the validation R² score of 0.880 demonstrates its ability to generalize to unseen data reasonably well. Its test R² score of 0.941 confirms strong predictive power on completely new data. Below this, a table displays a side-by-side comparison of five different regression models—LinearRegression, RandomForestRegressor, XGBRegressor, SVR (Support Vector Regressor), and GradientBoostingRegressor—each evaluated on training, validation, and test datasets. LinearRegression has lower scores across datasets, with a test R² of 0.610, suggesting it’s less effective at capturing the data’s complexity. In contrast, RandomForestRegressor excels with a high train R² of 0.986 and a test R² of 0.968, showing both high accuracy and generalizability. The XGBRegressor also performs consistently well across datasets, while SVR has relatively low scores, with a test R² of 0.597, indicating weaker predictive capacity. Finally, GradientBoostingRegressor achieves balanced scores around the 0.87 range across all datasets, showing good but slightly lower performance than XGB and RandomForest. Overall, RandomForestRegressor and XGBRegressor emerge as the best-performing models, with RandomForest slightly outperforming on test data and XGBRegressor showing stable and balanced results, making both models strong candidates for accurate and reliable predictions.

**10.2 Alert Message**

****

#### Result for given input data

The current Air Quality Index (AQI) is 48.0, indicating moderate air pollution. This may cause minor breathing discomfort, especially for sensitive groups. To protect your health, consider reducing outdoor activities, staying indoors in air-conditioned spaces, and using air purifiers. Additionally, wearing a mask outdoors, drinking plenty of water, and consulting a doctor for persistent symptoms are recommended. Stay informed about air quality updates and take necessary precautions to safeguard your well-being.

# Chapter 11 REFERENCES

1. Abdullah Kadri, Elias Yaacoub, Mohammed Mushtaha, And Adnan Abu-Dayya “Wireless Sensor Network For Real-Time Air Pollution Monitoring” IEEE Forum On Strategic Technology -2013.
2. Haibao Wang, Tingting Wu ,And Guangjie Wu,”Air Quality Monitoring System Based On Frequency Hopping System 2010 IEEE.
3. Robert L.Byer,Lawrence A.Shepp, “Two-Dimensional Remote AirPollution Monitoring Viatomography”.Vol.4/ March 1979 / OPTICS LETTERS.
4. Kostandina Veljanovska and Angel Dimoski, "Air Quality Index Prediction Using Simple Machine Learning Algorithms", International Journal of Emerging Trends Technology in Computer Science (IJETTCS), 2018.
5. Xiaosong Zhao, Rui Zhang, Jheng-Long Wu, Pei-Chann Chang and Yuan Ze, "A Deep Recurrent Neural Network for Air Quality Classification", Journal of Information Hiding and Multimedia Signal Processing, 2018.
6. Savita Vivek Mohurle, Richa Purohit and Manisha Patil, "A study of fuzzy clustering concept for measuring air pollution index", International Journal of Advanced Science and Research, 2018.
7. 4.C R Aditya, Chandana R Deshmukh, D K Nayana and Praveen Gandhi Vidyavastu, "Detection and Prediction of Air Pollution using Machine Learning Models", International Journal of Engineering Trends and Technology (IJETT), 2018.
8. Shan Zhang, Xiaoli Li, Yang Li and Jianxiang Mei, "Prediction of Urban PM2.5 Concentration Based on Wavelet Neural Network" in , IEEE, 2018.
9. Timothy M. Amado and Jennifer C. Dela Cruz, "Development of Machine Learning-based Predictive Models for Air Quality Monitoring and Characterization" in , IEEE, 2018.
10. Arwa Shawabkeh, Feda Al-Beqain, Ali Rodan and Maher Salem, "Benzene Air Pollution Monitoring Model using ANN and SVM" in , IEEE, 2018.