Project ID: 24/CSE/1/3

REAL TIME AIR QUALITY MONITORING SYSTEM

A PROJECT REPORT
Submitted By

Sagar Sen
(2100270109009)
Vikash
(2100270109012)
Shivam Saini
(2100270109010)
Mahendra Pratap Singh
(2100270109005)

Under the Guidance of
Mr. Updesh Jaiswal

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering

Department of Computer Science & Engineering
AJAY KUMAR GARG ENGINEERING COLLEGE, GHAZIABAD
DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

May 27, 2024

Declaration

We hereby declare that the work presented in this report entitled "REAL TIME AIR QUALITY MONITORING SYSTEM", was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors / sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors / sources.

We affirm that no portion of our work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

Name : Sagar Sen
Roll No. : 2100270109009

Name : Vikash
Roll No. : 2100270109012

Name : Shivam Saini
Roll No. : 2100270109010

Name : Mahendra Pratap Singh
Roll No. : 2100270109005

Certificate

This is to certify that the report entitled **REAL TIME AIR QUALITY MONITORING SYSTEM** submitted by SAGAR SEN(2100270109009), VIKASH(2100270109012), SHIVAM SAINI(2100270109010) and MAHENDRA PRATAP SINGH(2100270109005) to the Dr. A.P.J.Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in CSE is a bonafide record of the project work carried out by him/her under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mr. Updesh Jaiswal
Project Guide Name
Assistant Professor
Dept. of Computer Science & Engineering
AKG Engineering College

Dr. Shashank Sahu
Professor In-charge, CSE
Dept. of Computer Science & Engineering
AKG Engineering College

Place: Ghaziabad
Date: May 27, 2024

Acknowledgement

We would like to express our sincere gratitude to everyone who contributed to the development and success of the Real-Time Air Quality Monitoring System.

Our heartfelt thanks go to Mr. Updesh Kumar Jaiswal (Assistant Professor), insightful feedback, and unwavering support were instrumental throughout the entire process.

We are also grateful to our technical team, including our team members, for their hard work, expertise, and dedication in developing and implementing the system. Their collaboration and innovative ideas were critical in overcoming the various challenges faced during the project.

Special thanks to the kaggle, whose valuable input on air quality standards and monitoring techniques greatly enhanced the accuracy and reliability of our system.

We would like to acknowledge the local authorities and community members who cooperated with us, providing essential data and feedback that informed our system's development and deployment.

Finally, we are deeply thankful to our families and friends for their encouragement and understanding during the course of this project.

This project would not have been possible without the collective efforts and support of all those mentioned above.

Thank you.

Abstract

Air quality prediction plays a pivotal role in safeguarding public health and environmental sustainability. This paper presents a comprehensive review of methodologies and advancements in air quality prediction models. It outlines the fundamental principles underlying air quality prediction, including the influence of meteorological factors, pollutant emissions, and geographical features. Traditional regression-based models, machine learning techniques, and hybrid approaches are discussed, highlighting their strengths and limitations. Furthermore, the integration of remote sensing data, IoT devices, and emerging technologies such as artificial intelligence and deep learning is explored for enhancing prediction accuracy and spatial resolution. The review also addresses challenges such as data sparsity, model interpretability, and the need for real-time prediction systems. Finally, future directions in air quality prediction research, including the development of ensemble models, uncertainty quantification, and incorporation of socio-economic factors, are proposed to foster more robust and actionable air quality forecasting systems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Air quality is a critical component of environmental health, directly impacting human well-being, ecosystems, and climate. Real-time air quality monitoring systems have emerged as a vital technology to measure, analyze, and report air pollution levels accurately and promptly. These systems utilize advanced sensors, data analytics, and communication technologies to provide immediate information on air quality, helping to address pollution issues effectively.

Importance of Air Quality Monitoring Air pollution is a significant global concern, linked to various health problems such as respiratory diseases, cardiovascular conditions, and premature deaths. Monitoring air quality in real-time enables:

Public Health Protection: By providing current data, these systems can alert vulnerable populations to poor air quality, enabling them to take protective measures. Policy and Regulation Enforcement: Governments and regulatory bodies can use the data to implement and enforce environmental regulations. Research and Development: Accurate data supports scientific research on pollution sources, effects, and mitigation strategies. Public Awareness and Engagement: Real-time information empowers citizens to be informed and involved in air quality issues. Components of Real-Time Air Quality Monitoring Systems Sensors: High-precision sensors detect and measure pollutants such as particulate matter (PM2.5, PM10), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), and volatile organic compounds (VOCs). Data Acquisition: Sensors continuously collect air quality data, which is then transmitted to a central system. Data Processing and Analysis: The collected data is processed using algorithms to analyze pollution levels, trends, and patterns. Communication Network: Data is transmitted via wired or wireless networks to ensure real-time updates. User Interface: Dashboards,

mobile apps, and websites provide accessible and understandable information to users, including real-time air quality indices (AQI). Technologies Used in Air Quality Monitoring IoT (Internet of Things): IoT devices enhance connectivity and data collection, making real-time monitoring more efficient and widespread. Machine Learning and AI: These technologies help in predicting pollution trends and identifying sources of pollution. Geospatial Technologies: GIS and GPS technologies map air quality data to specific locations, offering spatial analysis of pollution. Applications Urban Planning: Cities use air quality data to design greener urban spaces and control traffic emissions. Industrial Monitoring: Industries monitor their emissions to comply with environmental standards and reduce their ecological footprint. Public Health Initiatives: Health agencies develop strategies based on air quality data to mitigate health risks. Environmental Research: Scientists study the effects of pollutants on climate and ecosystems using detailed air quality data. Challenges and Future Directions While real-time air quality monitoring systems are advancing rapidly, challenges remain:

Accuracy and Calibration: Ensuring sensor accuracy and regular calibration is crucial for reliable data. Data Integration: Integrating data from various sources and formats can be complex. Cost: High-quality sensors and infrastructure can be expensive, limiting widespread adoption. Data Privacy and Security: Protecting the collected data from unauthorized access is essential. Future developments may include enhanced sensor technologies, broader adoption of AI for predictive analytics, and more robust and cost-effective systems. As technology progresses, real-time air quality monitoring will continue to play a pivotal role in creating healthier and more sustainable environments.

Chapter 2

Litreture Review

The analysis of nine years of air pollution data in Indian cities, with a focus on twelve air pollutants and the Air Quality Index (AQI) has been addressed . The dataset was preprocessed and cleaned, followed by the application of data visualization techniques to uncover hidden patterns and trends. Furthermore, the paper addresses data imbalance through resampling techniques and compares the performance of five popular ML models using standard metrics. The accuracy of various machine learning (ML) models was evaluated using classical statistical error metrics on both the train and test sets. The XGBoost model demonstrated the highest accuracy, while the SVM model had the lowest. To compare the performance of ML models, metrics such as MAE, RMSE, RMSLE, and R2 were evaluated. Based on these metrics, the XGBoost model was found to be the best overall performer, achieving optimal values during both training and testing phases., the study presents machine learning models for analyzing air pollution using TAQMN data from Taiwan. The dataset spans from 2011 to 2019 and includes records from 76 air pollution stations. The focus of the study is on predicting particulate matter PM2.5 levels using machine learning models, with evaluation based on statistical metrics such as MAE, MSE, RMSE, and R2. The results indicate that the proposed models outperform previous models, with actual and predicted values showing close agreement. Based on the evaluation, the study concludes that the gradient boosting regressor model is the most effective in forecasting air pollution on the TAQMN dataset.

Air quality prediction would encompass a variety of methodologies, datasets, and findings across different research studies.

- Introduction to Air Quality Prediction:
  - Define the importance of predicting air quality.
  - Discuss the impact of poor air quality on human health, the environment, and various sectors of the economy.
  - Highlight the need for accurate and timely air quality forecasts for effective policymaking and public health interventions.
- Datasets:
  - Description of the types of data typically used in air quality prediction, such as meteorological data, satellite imagery, pollution monitoring station data, and land use information.
  - Evaluation of the quality and availability of these datasets, including issues related to spatial and temporal resolution, coverage, and consistency.
- Key Findings:
  - Summary of significant research findings in the field of air quality prediction.
  - Discussion of factors influencing air quality, such as emissions from industrial activities, vehicular traffic, wildfires, and atmospheric conditions.
  - Examination of the effectiveness of different prediction models in capturing these factors and providing accurate forecasts.
- Applications:
  - Review of real-world applications of air quality prediction, including urban planning, transportation management, public health initiatives, and emergency response.
  - Case studies illustrating how air quality forecasts have been used to mitigate the impact of pollution on communities and ecosystems.

- Challenges and Future Directions:
  - Identification of current challenges in air quality prediction, such as data availability, model complexity, and the need for improved understanding of atmospheric processes.
  - Discussion of ongoing research efforts aimed at addressing these challenges and advancing the state-of-the-art in air quality forecasting.
  - Exploration of emerging trends and technologies that may shape the future of air quality prediction, such as the integration of satellite data, the use of sensor networks, and the application of artificial intelligence.

Chapter 3

Basics

Air quality prediction involves forecasting the level of pollutants present in the atmosphere over a certain period of time. Here's a basic outline of the process:
Python: Python is an interpreted, high, general-purpose programming language. Created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructsand object-oriented approach aim to help programmers write clear, logical code for small and 9 large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms including structured (particularly, procedural), object oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Data Collection: Air quality prediction relies heavily on data collection from various sources. This includes monitoring stations, satellites, weather stations, and other environmental sensors. Data collected typically includes information on pollutants such as particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), ozone (O3), carbon monoxide (CO), and volatile organic compounds (VOCs), as well as meteorological data like temperature, humidity, wind speed, and atmospheric pressure.

Data Preprocessing: Once the data is collected, it needs to be preprocessed. This involves cleaning the data to remove outliers and errors, handling missing values, and normalizing or standardizing the data to ensure consistency and comparability across different sources. Feature Selection/Extraction: In this step, relevant features or variables that affect air quality are identified. This may involve statistical analysis, domain

knowledge, or machine learning techniques to select the most important features for prediction.

Analysis and Visualization of Data:

NumPy: NumPy is a python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely. NumPy stands for Numerical Python. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, in NumPy provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

Pandas: Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame.

MatPlotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.

Model Selection: Various machine learning and statistical models can be used for air quality prediction, including linear regression, decision trees, random forests, support vector machines (SVM), neural networks, and more advanced techniques like Long Short-Term Memory (LSTM) networks for time-series data. The choice of model depends on factors such as the complexity of the data, the size of the dataset, and the specific requirements of the prediction task. Model Training: Once a suitable model is selected, it needs to be trained on historical data. This involves feeding the model with input features (such as pollutant concentrations, meteorological data) and corresponding output labels (actual pollutant concentrations at a given time) to allow the model to learn the underlying patterns and relationships in the data. Model Evaluation: After training, the performance of the model is evaluated using evaluation metrics such as mean absolute error (MAE), root mean square error (RMSE), or coefficient of determination (R-squared). This helps assess how well the model generalizes to unseen data and whether it meets the desired accuracy criteria. Prediction: Once the model is trained and evaluated, it can be used to make predictions on new data. This involves feeding the model

with current input features (e.g., current weather conditions) to predict future pollutant concentrations over a specific time period. Deployment and Monitoring: The trained model can be deployed in real-time systems for continuous monitoring and prediction of air quality. It's essential to monitor the model's performance over time and retrain it periodically to ensure that it remains accurate as environmental conditions change. Overall, air quality prediction is a complex task that requires expertise in data science, environmental science, and domain-specific knowledge. Continuous improvement in data collection methods, model development, and computational techniques can lead to more accurate and reliable predictions, which are crucial for mitigating the adverse effects of air pollution on public health and the environment.

Chapter 4

Flow Chart

flowchart for air quality prediction involves a series of steps to outline the process from data collection to prediction. Collect data from various sources such as air quality monitoring stations, weather stations, satellite imagery, and other relevant sources. Include parameters like pollutants concentration (PM2.5, PM10, CO, NO2, etc.), meteorological data (temperature, humidity, wind speed, etc.), geographical features, and any other factors influencing air quality. Clean the collected data to handle missing values, outliers, and inconsistencies. Normalize or scale the data to ensure uniformity across different features. Perform feature engineering to extract relevant features and enhance the predictive power of the model. Feature Selection: Identify the most significant features that contribute to air quality prediction. Utilize techniques such as correlation analysis, feature importance ranking, or domain knowledge to select relevant features. Model Selection: Choose appropriate machine learning or statistical models for air quality prediction. Common models include: Regression models (e.g., linear regression, decision trees, random forests). Time series forecasting models (e.g., ARIMA, SARIMA). Deep learning models (e.g., neural networks, LSTM). Consider the complexity of the model, computational efficiency, and interpretability.

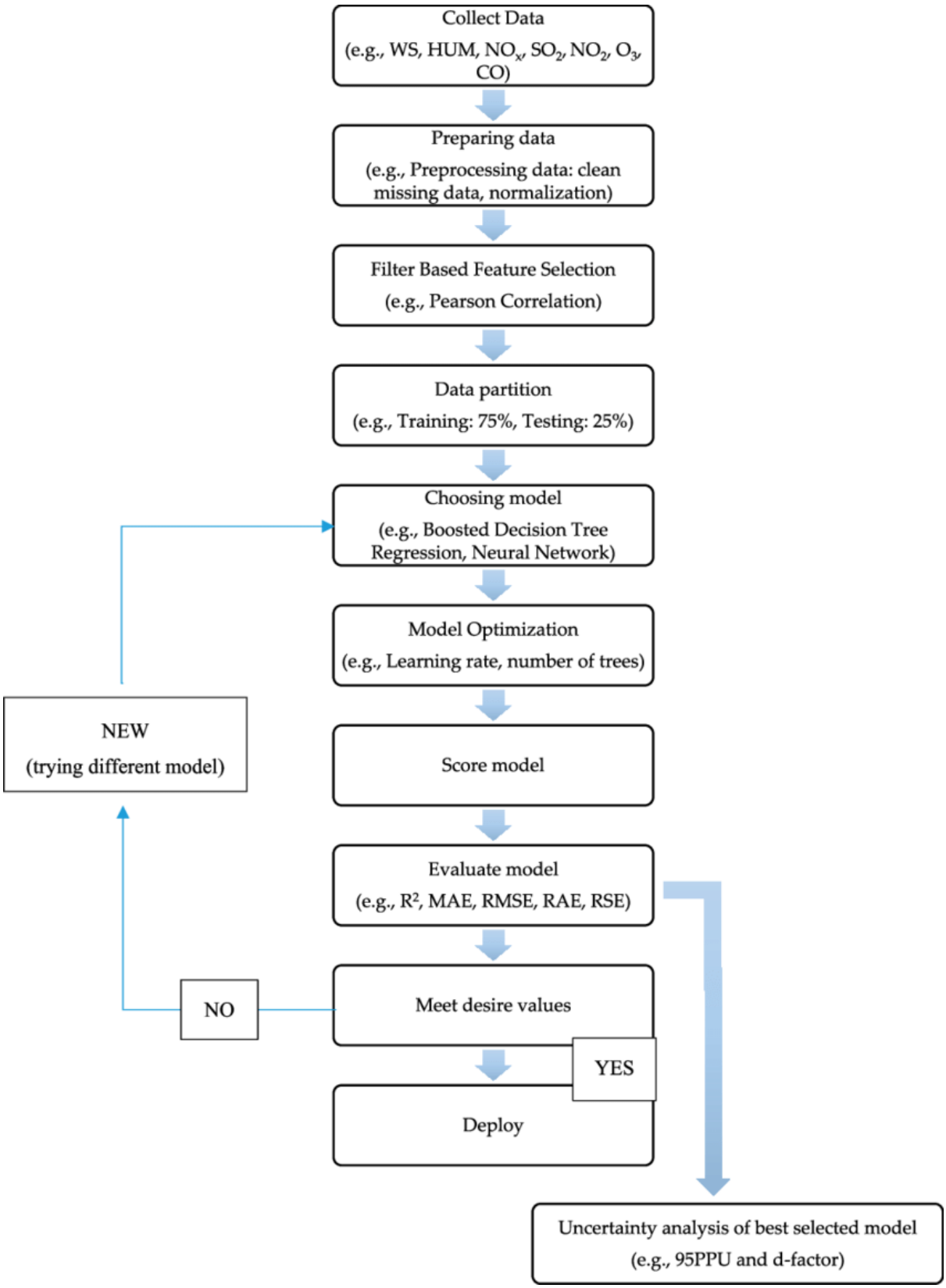


Figure 4.1: Flow Chart

Chapter 5

Logic

Air quality prediction involves a combination of data collection, analysis, and modeling to forecast the concentration of pollutants in the air at a specific location and time. Here's a simplified overview of the logic behind air quality prediction:

1. Data Collection: Various types of data are collected to understand the current state of the atmosphere. This includes:

   (a) Meteorological Data: Information about weather conditions such as temperature, humidity, wind speed, and direction. Weather conditions influence the dispersion and transformation of pollutants in the atmosphere.

   (b) Pollutant Concentration Data: Measurements of pollutants such as particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), ozone (O3), carbon monoxide (CO), and volatile organic compounds (VOCs). These measurements are typically collected from monitoring stations located throughout a region.

   (c) Emission Data: Information about the sources of pollutants, including industrial facilities, vehicles, power plants, and other sources of emissions. Emission inventories provide estimates of the amount and types of pollutants released into the atmosphere.

2. Data Set Generation:The dataset is generated using the CAMS global reanalysis (EAC4) dataset. The original data was in a netCDF format which was transformed into an excel file using a python program. The dataset contains various meteorological as well as aerosol parameters over Ahmedabad from 2011 to 2019 at the interval of 3 hours. The parameters of the dataset are wind speed, temperature, humidity, precipitation, atmospheric pressure, amount of water vapor, boundary

layer height and aerosol optical depth. The dataset was checked for missing values and no missing values were found.

3. Data Preprocessing: Raw data collected from various sources often needs to be preprocessed to remove noise, fill missing values, and ensure consistency. This may involve techniques such as data cleaning, interpolation, and quality control. Data preprocessing is a crucial step in preparing the raw data collected from an air quality monitoring system for machine learning (ML) analysis. This process begins with data cleaning, which involves removing any outliers, missing values, and erroneous readings that could skew the analysis. This can be achieved through methods such as interpolation for missing values and statistical techniques for outlier detection. Next, the data is normalized to ensure that different scales of measurement are standardized, which helps in improving the performance and convergence of many ML algorithms. For example, pollutant concentration levels may be scaled to a uniform range, such as 0 to 1, to eliminate biases due to varying units.

   Subsequently, feature engineering is performed to create new relevant features from the existing data that can provide additional insights to the ML model. This might include generating time-based features (e.g., hour of the day, day of the week) or environmental context features (e.g., temperature and humidity adjustments). The data is then split into training, validation, and testing sets to evaluate the model's performance accurately. During this phase, care is taken to ensure that the split maintains the temporal integrity of the data to prevent leakage and ensure realistic model performance evaluation.

   Dimensionality reduction techniques, such as Principal Component Analysis (PCA), may also be applied to reduce the complexity of the dataset while retaining its most important information. Additionally, labeling the data correctly is essential if supervised learning techniques are to be employed, ensuring that each data point is associated with the correct target variable, such as pollution levels classified into different air quality categories.

4. Feature Selection and Engineering: Relevant features are selected or engineered from the raw data to be used as input variables for the prediction model. This may include combining meteorological data with historical pollution data or incorporating spatial factors such as

proximity to emission sources.

5. Exploratory Data Analysis: The first few rows and columns of the dataset are shown . Different kinds of plots were generated to observe patterns in the data. Pair plots of all the parameters with the target variable were useful in observing the dependency of the target variable on different parameters.As it can be seen lower PM2.5 concentration is found at higher values of wind speed, boundary layer height and temperature. This is due to the fact that when the magnitude of these parameters is higher, there is more dispersion of pollutants. Hence, its concentration decreases.

6. Feature Analysis and Selection:Feature analysis and selection was performed using correlation analysis .Correlation coefficients were calculated for each pair of parameters and their values are shown.Wind speed, temperature, boundary layer height, and pressure show a high level of correlation with the target variable, so they are important to build a model.Feature analysis and selection are pivotal steps in developing an effective machine learning model for an air quality monitoring system. These processes ensure that the model utilizes the most relevant and informative data points, enhancing its accuracy and efficiency. In the context of air quality monitoring, feature analysis involves examining various parameters such as particulate matter (PM2.5 and PM10), ozone (O3), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), and volatile organic compounds (VOCs). Additionally, meteorological factors like temperature, humidity, wind speed, and wind direction are considered, as they significantly influence pollutant dispersion and concentration.

   Using statistical methods and domain knowledge, each feature's importance is assessed, identifying correlations and dependencies among them. Techniques such as correlation analysis, mutual information, and principal component analysis (PCA) help in understanding the contribution of each feature to the target variable, which is typically the air quality index (AQI) or specific pollutant concentrations.

   Subsequently, feature selection methods like Recursive Feature Elimination (RFE), Lasso regression, and tree-based algorithms (e.g., Random Forest) are employed to select the most impactful features while discarding redundant or less informative ones. This selection process not only reduces the dimensionality of the data, thereby improving

computational efficiency, but also mitigates the risk of overfitting, leading to a more robust and generalizable model. By focusing on the most relevant features, the air quality monitoring system can provide accurate predictions and valuable insights, facilitating better environmental management and public health protection.

7. Model Development: Machine learning models are trained using historical data to learn patterns and relationships between input variables (e.g., weather conditions, pollutant emissions) and air quality outcomes (e.g., pollutant concentrations). Common modeling techniques include:

   (a) Regression Models: Linear regression, polynomial regression, or other regression techniques can be used to predict pollutant concentrations based on input variables.

   (b) Time Series Analysis: Time series models such as ARIMA (AutoRegressive Integrated Moving Average) or seasonal decomposition techniques can capture temporal patterns in air quality data.

   (c) Machine Learning Algorithms: Supervised learning algorithms like decision trees, random forests, support vector machines (SVM), or neural networks can be employed to predict air quality.

Chapter 6

Methodology

The methodology for air quality prediction typically involves the following steps:

Data Collection: Gather historical and real-time data on various air quality parameters such as particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), ozone (O3), carbon monoxide (CO), and meteorological factors like temperature, humidity, wind speed, etc.
Particulate Matter (PM2.5 and PM10): These are fine particles suspended in the air.
Ozone (O3): A gas that can cause respiratory problems.
Nitrogen Dioxide (NO2): A pollutant from combustion processes.
Sulfur Dioxide (SO2): Produced by volcanic eruptions and industrial processes.
Carbon Monoxide (CO): A colorless, odorless gas from incomplete combustion.
Volatile Organic Compounds (VOCs): Emitted as gases from certain solids or liquids.

Data Preprocessing: Clean the data by handling missing values, outliers, and inconsistencies. Perform feature engineering to extract relevant features and transform data into a suitable format for modeling. Raw data collected from various sources often needs to be preprocessed to remove noise, fill missing values, and en- sure consistency. This may involve techniques such as data cleaning, interpolation, and quality control

Feature Selection: Identify the most important features that significantly influence air quality using techniques like correlation analysis, feature importance, or domain knowledge.eature analysis and selection was performed using correlation analysis .Correlation coefficients were calcu- lated

for each pair of parameters and their values are shown.Wind speed, temperature, boundary layer height, and pressure show a high level of correlation with the target variable, so they are important to build a model.

Model Selection: Choose appropriate machine learning or statistical models such as regression, time series analysis, neural networks, or ensemble methods based on the nature of the data and the prediction task.

Model Training: Split the data into training and validation sets, and train the selected models using the training data. Optimize hyperparameters and validate model performance using the validation set.Model training for an air quality monitoring system involves several crucial steps to ensure accurate and reliable predictions of air quality indices and pollutant levels. Initially, the collected data, which includes various parameters such as PM2.5, PM10, CO, NO2, SO2, O3, VOCs, temperature, and humidity, is preprocessed. This preprocessing stage involves cleaning the data to remove any noise or outliers, normalizing the values to a standard scale, and handling any missing data through interpolation or imputation methods. Once the data is preprocessed, feature engineering is performed to extract meaningful features that can improve the model's performance.

The next step is to select appropriate machine learning algorithms. Commonly used algorithms for air quality prediction include regression models, decision trees, random forests, support vector machines, and neural networks. The choice of algorithm depends on the complexity of the data and the specific prediction goals. The dataset is then split into training and testing subsets, typically using an 80-20 split to train the model on the majority of the data while reserving a portion for validation.

During the training phase, the model learns the relationships between the input features and the target variables, adjusting its parameters to minimize prediction errors. Techniques such as cross-validation are employed to ensure the model generalizes well to unseen data and to prevent overfitting. Hyperparameter tuning is also conducted to optimize the model's performance. Once the model is trained, it is evaluated using the test dataset, and performance metrics such as mean absolute error (MAE), root mean square error (RMSE), and R-squared (R^2) are calculated to assess its accuracy.

Finally, the trained model is deployed within the air quality monitoring system, where it can make real-time predictions based on live data inputs.

Continuous monitoring and periodic retraining with new data ensure that the model remains accurate and responsive to changing environmental conditions.

Model Evaluation: Evaluating the performance of machine learning models in an air quality monitoring system project is a multi-faceted process that ensures the models provide accurate and actionable insights. The evaluation begins with splitting the dataset into training and testing sets to assess the model's generalization capability. Key metrics used in this evaluation include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) score. These metrics provide a quantitative measure of the model's prediction accuracy, highlighting the average magnitude of errors and the model's ability to explain variance in the data. Cross-validation techniques, such as k-fold cross-validation, are employed to ensure robustness by evaluating the model's performance across different subsets of the data. Additionally, feature importance analysis helps in understanding which variables most significantly impact air quality predictions, ensuring the model's interpretability and reliability. Finally, real-time validation with live data streams ensures that the model maintains accuracy over time and adapts to new patterns and anomalies. This comprehensive evaluation framework is crucial for deploying a reliable and effective air quality monitoring system that can support decision-making and policy development.Evaluate the trained models using appropriate metrics such as mean absolute error (MAE), mean squared error (MSE), or root mean squared error (RMSE) on the validation set to assess their predictive accuracy.

Model Deployment: Deploying a machine learning model for an air quality monitoring system involves several crucial steps to ensure effective and reliable predictions. First, the collected data from various sensors, including PM2.5, PM10, CO, NO2, SO2, O3, VOCs, temperature, and humidity, is preprocessed and cleaned to remove noise and handle missing values. This data is then used to train machine learning models, such as regression models for predicting pollutant levels or classification models for categorizing air quality indices. The trained models are validated using a separate dataset to assess their accuracy and generalizability. Once validated, these models are deployed on a cloud-based platform, enabling real-time predictions and scalability. The deployment involves setting up an API

Table 6.1: Sample Data

T	TM	Tm	SLP	H	V	VM	PM 2.5
16.9	25.1	6.6	1021.3	65	1.1	2	284.7958
15.5	24.1	7.7	1021	8	71	11.1	219.7202
14.9	22.8	8	8	1018.4	73	13	182.3929
18.3	24.7	11.5	1018.1	85	1.1	0.5	129.2323

that can handle incoming data from sensors, process it, and return predictions. This setup also includes mechanisms for continuous monitoring and updating of the models to adapt to new data, ensuring that the predictions remain accurate over time. Additionally, an interface is developed for stakeholders to visualize air quality trends and receive alerts on pollution levels, integrating the ML model's output seamlessly into the air quality monitoring system. This deployment not only enhances the system's predictive capabilities but also facilitates proactive measures to manage air quality effectively.

Continuous Monitoring and Updating: Monitor the deployed models regularly to ensure their performance remains satisfactory over time. Update models as new data becomes available or as the underlying factors affecting air quality change.

Feedback Loop: Incorporate feedback from users, stakeholders, or monitoring systems to improve model accuracy and relevance. Iterate on the prediction process to adapt to evolving environmental conditions and user needs.

By following these steps and leveraging advanced modeling techniques and data analytics, accurate and reliable air quality predictions can be achieved, aiding in environmental management and public health decision-making.

Machine Learning: Machine learning for air quality prediction is a promising area that can greatly benefit environmental monitoring and public health. Here's a general outline of how you could approach it:
1. Data Collection: Gather historical data on various air quality parameters such as particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), and meteorological factors like temperature, humidity, wind speed, and precipitation. Data from government monitoring stations, satellites, weather stations, and even crowdsourced data can be valuable.

2. Data Preprocessing: Clean the data by handling missing values, outliers, and inconsistencies. Perform feature engineering to extract relevant features or derive new features that could improve prediction accuracy. For example, you might create features like daily averages, maximums, minimums, or trends over time.
3. Model Selection: Choose appropriate machine learning models for air quality prediction. Common models include:

- Linear Regression: Simple and interpretable, but may not capture complex relationships.
- Decision Trees: Can handle non-linear relationships and interactions between variables.
- Random Forests: Ensemble of decision trees that generally performs well and is robust to overfitting.
- Gradient Boosting Machines (GBM): Builds trees sequentially, focusing on the mistakes of previous models.
- Support Vector Machines (SVM): Effective for small to medium-sized datasets with high dimensionality.
- Neural Networks: Deep learning models that can capture intricate patterns but require large amounts of data and computational resources.

4. Model Training: Split your data into training and testing sets to evaluate model performance. You can further split the training set for cross-validation to tune hyperparameters and prevent overfitting.
5. Model Evaluation: Assess the performance of your models using evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or coefficient of determination (R-squared). These metrics quantify how well your model predicts air quality compared to actual observations.
6. Model Deployment: Once you have a satisfactory model, deploy it to make real-time predictions. This could involve building a web service or integrating the model into an existing environmental monitoring system.
7. Continuous Improvement: Monitor the performance of your model over time and update it as new data becomes available or as environmental conditions change. You may need to retrain your model periodically to maintain its accuracy.

It's crucial to collaborate with domain experts such as environmental scientists and meteorologists throughout the process to ensure that your model captures the relevant factors influencing air quality and produces meaningful predictions.

6.1 Regression Model

A regression model is a statistical tool used to understand the relationship between a dependent variable and one or more independent variables. The primary goal is to predict the value of the dependent variable based on the values of the independent variables. There are several types of regression models, but the most commonly used are linear regression and logistic regression.

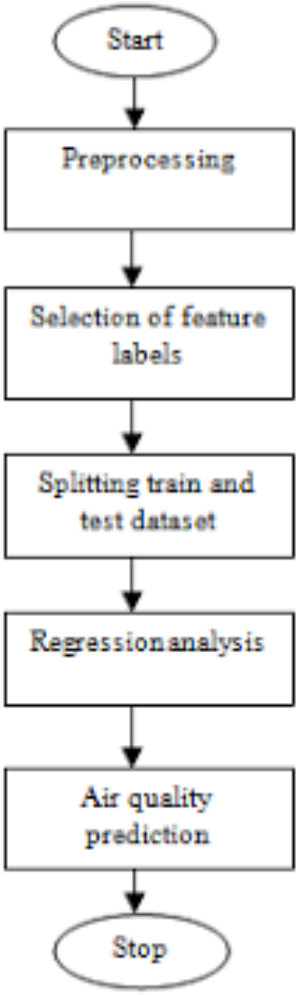


Figure 6.1: Regration Model

Key Components:

- Dependent Variable: The outcome or the variable being predicted (e.g., house price).
- Independent Variables: The predictors or factors influencing the dependent variable (e.g., number of bedrooms, location).
- Coefficients: These quantify the relationship between each independent variable and the dependent variable. Intercept: The expected value of the dependent variable when all independent variables are zero.

Chapter 7

Algorithm

7.1 Data Cleaning

Data cleaning, also known as data cleansing or data scrubbing, is the process of identifying and correcting (or removing) errors and inconsistencies in data to improve its quality and reliability.
This involves several steps, including:

- Removing Duplicates: Eliminating repeated entries to ensure each record is unique.
- Correcting Errors: Fixing typos, misspellings, and incorrect values.
- Handling Missing Data: Addressing gaps in data by filling in missing values, removing incomplete records, or using algorithms to predict missing information.
- Standardizing Data: Ensuring that data is in a consistent format across the dataset, such as standardizing date formats or measurement units.
- Validating Data: Ensuring the data adheres to defined rules or constraints, such as specific ranges or data types.

Effective data cleaning improves the accuracy, reliability, and overall quality of the data, making it more suitable for analysis and decision-making.

7.2 Visualize the distribution of each variable using histogram

A histogram is a graphical representation of the distribution of a dataset. It displays data using contiguous rectangular bars (bins), where the height

of each bar indicates the frequency or count of data points within a particular range or interval. The x-axis represents the intervals or bins, while the y-axis represents the frequency or count of data points in each bin. Histograms are useful for visualizing the shape, central tendency, and variability of a dataset, as well as identifying patterns such as skewness, modality, and outliers.

7.3 Visualize the correlation between variables using a heatmap

A heatmap is a graphical representation used to visualize the correlation between variables in a dataset. It displays data in a matrix format, where each cell in the matrix represents the correlation coefficient between two variables. The cells are color-coded to indicate the strength and direction of the correlation, with different colors representing positive, negative, or no correlation. Typically, a color gradient is used, where darker or more intense colors indicate stronger correlations. Heatmaps are useful for quickly identifying patterns and relationships between multiple variables in a visually intuitive way.

7.4 Visualize the relationship between target variable and the other variable using scatterplots

A scatterplot is a graphical tool used to visualize the relationship between a target variable (often the dependent variable) and another variable (often the independent variable). In a scatterplot, each point represents an observation from the dataset, with its position determined by the values of the two variables: one plotted along the x-axis and the other along the y-axis. This allows for easy identification of trends, correlations, and patterns between the variables. For instance, a positive or negative correlation can be observed if the points tend to follow a line with a positive or negative slope, respectively. Scatterplots are particularly useful for detecting linear and non-linear relationships, clusters, and potential outliers.

7.5 Calculate spearman correlation coefficient

The Spearman correlation coefficient is a measure of the strength and direction of the association between two ranked variables. Here's a brief description of how to calculate it:

- Rank the Data: Convert the raw data for each variable into ranks. For tied values, assign the average rank.
- Calculate the Differences: Find the difference between the ranks of each pair of observations.
- Square the Differences: Square each of these differences.
- Sum the Squared Differences: Add up all the squared differences.
- Apply the Formula: Use the Spearman correlation formula:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where ρ is the Spearman correlation coefficient
$d_i$is the difference between the ranks of each pair of observations
n is the number of observations

The result ranges from -1 to 1, indicating a perfect negative to a perfect positive correlation, with 0 indicating no correlation.

7.6 Formula for calculate AQI Index

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{HI} - BP_{Lo}}(C_p - BP_{Lo}) + I_Lo$$

where I_p = the index for pollutant p
C_p = the truncated concentration of pollutant p
BP_{Hi} = the concentration breakpoint that is greater than or equal to C_p
BP_{Lo} = the concentration breakpoint that is less than or equal to C_p
I_{Hi} = the AQI value corresponding to BP_{Hi}
I_{Lo} = the AQI value corresponding to BP_{Lo}

Chapter 8

Experimental Work

Predicting air quality through AI programming involves several steps, including data collection, preprocessing, model selection, training, and evaluation. Here's an overview of how you might approach it:
1. Data Collection: Gather air quality data from various sources such as monitoring stations, satellites, weather stations, and other relevant sources. This data may include pollutants such as PM2.5, PM10, NO2, SO2, O3, CO, as well as meteorological data like temperature, humidity, wind speed, and direction.
2. Data Preprocessing: Clean the data by handling missing values, outliers, and inconsistencies. Perform feature engineering to extract relevant features from raw data, such as computing rolling averages, hourly averages, or daily averages of pollutant concentrations.
3. Feature Selection: Identify the most important features that influence air quality prediction. This step helps reduce computational complexity and prevents overfitting.
4. Model Selection: Choose an appropriate AI model for air quality prediction. Options include: . Regression Models: Linear regression, polynomial regression, support vector regression (SVR). . Time Series Models: ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal ARIMA), LSTM (Long Short-Term Memory) networks. . Ensemble Models: Random Forest, Gradient Boosting Machines (GBM), XGBoost. . Deep Learning Models: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU).
5. Training: Split the data into training and testing sets. Train the selected model using the training data. During training, the model learns the relationships between input features (e.g., pollutant concentrations, weather data) and the target variable (e.g., air quality index).
6. Evaluation: Evaluate the performance of the trained model using the

testing dataset. Common evaluation metrics for air quality prediction include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) score.
7. Hyperparameter Tuning: Fine-tune the model hyperparameters to optimize performance. Techniques like grid search or random search can be used for hyperparameter tuning.
8. Deployment: Once satisfied with the model performance, deploy the model to make real-time air quality predictions. This may involve integrating the model into a web application, mobile app, or other systems where air quality information is needed.
9. Monitoring and Maintenance: Continuously monitor the performance of the deployed model and update it as new data becomes available or as environmental conditions change. This ensures that the model remains accurate and reliable over time.
10. Dataset Generation:The dataset is generated using the CAMS global reanalysis (EAC4) dataset. The original data was in a netCDF format which was transformed into an excel file using a python program. The dataset contains various meteorological as well as aerosol parameters over Ahmedabad from 2011 to 2019 at the interval of 3 hours. The parameters of the dataset are wind speed, temperature, humidity, precipitation, atmospheric pressure, amount of water vapor, boundary layer height and aerosol optical depth. The dataset was checked for missing values and no missing values were found

A	B	C	D	E	F	G	H
datetime	**ws**	**wd**	**temp**	**rh**	**dew_temp**	**precipitation**	**pressu**
2018-08-03 05:30:00	3.55158854	32.0852165	28.1619263	79.0088208	24.1758728	0.329673171	998.779
2018-08-03 08:30:00	4.64429522	23.8890533	28.5700378	78.8659418	24.54180908	0.224019006	1000.54
2018-08-03 11:30:00	6.21366453	15.7002201	31.9954834	63.0434924	24.08288574	0.597972214	1000.39
2018-08-03 14:30:00	5.87534332	26.6237946	33.8738708	53.5249159	23.11972046	0.74990958	998.660
2018-08-03 17:30:00	4.53648806	41.0227699	34.1471252	53.6204473	23.40170288	0.819622934	997.070
2018-08-03 20:30:00	5.08432579	38.9917831	30.9125671	65.7672295	23.76229858	0.965121865	997.94
2018-08-03 23:30:00	5.09749317	32.5261383	29.1554871	71.4908506	23.47201538	1.041733146	999.518
2018-08-04 02:30:00	4.07942867	33.3100624	28.1891785	76.6207054	23.69158936	1.046886563	998.762
2018-08-04 05:30:00	3.38585186	33.2337723	28.0111389	79.6980843	24.17419434	0.376262367	998.049
2018-08-04 08:30:00	5.60526562	13.4580564	28.6326599	78.0090016	24.42001343	0.311337054	999.536
2018-08-04 11:30:00	6.16605854	12.3580799	31.5552063	63.4895845	23.78469849	0.723979652	999.426
2018-08-04 14:30:00	5.69695711	23.6094494	34.7661438	49.2338756	22.56057739	0.757692575	997.781
2018-08-04 17:30:00	4.8858881	30.0545139	34.7698975	51.2046007	23.21194458	0.757951915	996.304

Figure 8.1: Dataset Generation

11. Correlation Analysis: Throughout the process, it's essential to collaborate with domain experts in environmental science and air quality management to ensure the AI model's predictions.

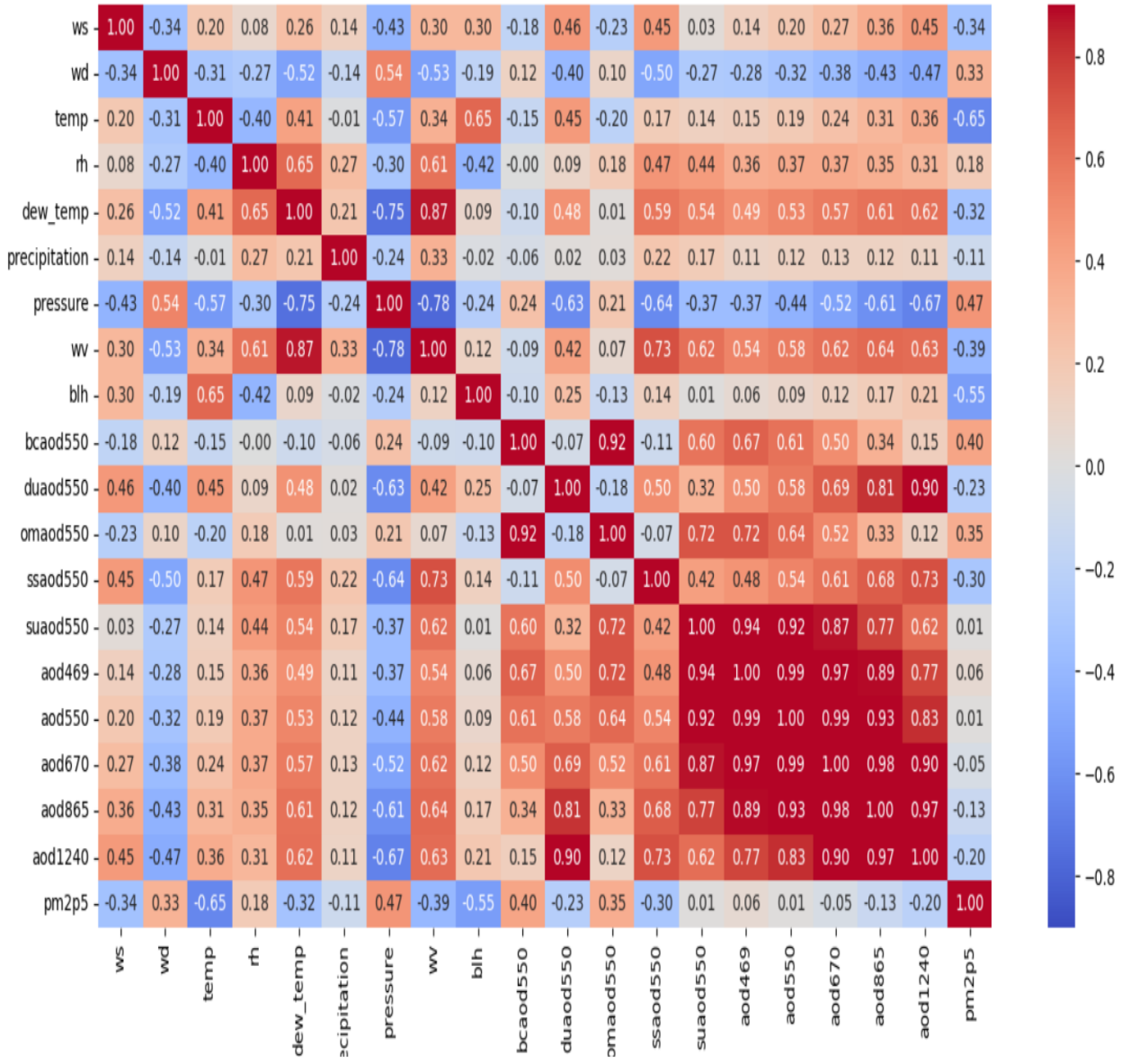


Figure 8.2: Corelation Analysis

Chapter 9

Results and Prediction

Air quality prediction involves forecasting the levels of air pollutants such as particulate matter (PM), nitrogen dioxide (NO2), ozone (O3), sulfur dioxide (SO2), carbon monoxide (CO), and other pollutants in the atmosphere over a certain period. Predicting air quality is crucial for public health, environmental protection, and urban planning. Here's how it's typically done:

1. Data Collection: Historical air quality data is collected from various monitoring stations. This data includes pollutant concentrations, meteorological data (temperature, humidity, wind speed, etc.), geographical data, and other relevant information.

2. Data Preprocessing: The collected data is cleaned, preprocessed, and formatted to be suitable for analysis. Missing values are handled, outliers are identified and dealt with, and data is normalized or standardized if necessary.

3. Feature Selection/Engineering: Relevant features that affect air quality, such as weather conditions, time of day, traffic density, industrial activities, etc., are selected or engineered. Relevant features are selected or engineered from the raw data to be used as input variables for the prediction model. This may include combining meteorological data with historical pollution data or incorporating spatial factors such as proximity to emission sources.

4. Exploratory Data Analysis: The first few rows and columns of the dataset are shown . Different kinds of plots were generated to observe patterns in the data. Pair plots of all the parameters with the target variable

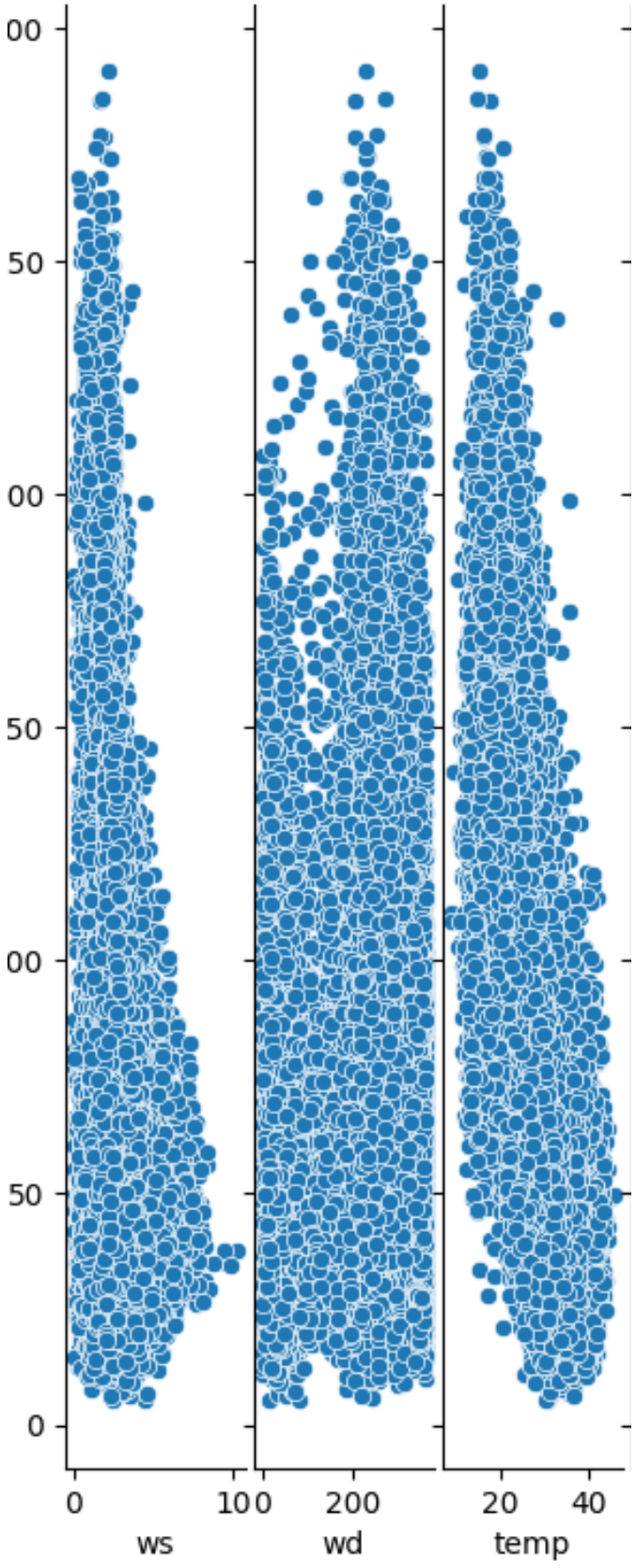


Figure 9.1: Predicted Data

were useful in observing the dependency of the target variable on different parameters.As it can be seen lower PM2.5 concentration is found at higher values of wind speed, boundary layer height and temperature. This is due to the fact that when the magnitude of these parameters is higher, there is more dispersion of pollutants. Hence, its concentration decreases.

5. Model Selection: Various machine learning models such as linear regression, decision trees, random forests, support vector machines (SVM), neural networks, etc., can be used for air quality prediction. The choice of model depends on the complexity of the problem and the available data.
(i)Linear Models: Linear Regression (OLS - Ordinary Least Square), Ridge Regression, and Lasso Regression from scikit-learn python library have been used to develop models. Models Ridge Regression and Lasso Regression were developed with cross-validation by trying different regularizer strengths and data was fitted with the best value of regularizer.
(ii)Kernel-based Models: Ridge Regression model was developed using a periodic kernel with random search of the best parameters. 100 iterations were performed to choose the best combination of alpha, kernel length scale, and kernel periodicity. The model was fit with the dataset and evaluated using different metrics. Randomized Search provided the best values of alpha = 771.7, kernel length scale = 32.0, and kernel periodicity = 0.22 out of 100 iterations. The Gaussian Process Regressor model was developed using a periodic kernel to address daily and seasonal periodicity, and quadratic kernel to address trends in the dataset.

6. Model Evaluation: Models were evaluated based on metrics such as r 2 score, mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE) using scikit-learn library of python. It is evident from table 1 that the linear model performances are not satisfactory as they are linear in nature and there is some periodicity present in the data. Hence, the models with periodic kernel functions were developed further. The performances of models with periodic kernels was still not satisfactory as they were still not able to capture the pattern in the data accurately. Finally the ensemble-based models showed good performance for the dataset. There wasn't a big difference in the performances of these models.

7. Training: The selected model is trained on the preprocessed data. This involves feeding the historical data into the model and adjusting its parameters to minimize the difference between the predicted and actual air quality values.

8. Evaluation: The trained model is evaluated using validation data to assess its performance. Common evaluation metrics include mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), etc.

9. Prediction: Once the model is trained and evaluated, it can be used to make predictions for future air quality levels based on input data such as weather forecasts, traffic patterns, and industrial activities.

10. Deployment: The trained model is deployed in a real-time or near-real-time environment where it continuously monitors incoming data and provides air quality predictions.

11. Monitoring and Maintenance: The deployed model is continuously monitored, and periodic updates and maintenance are performed to ensure its accuracy and reliability.
Overall, air quality prediction is a complex task that requires the integration of various data sources, advanced modeling techniques, and domain knowledge in environmental science and engineering.

Chapter 10

Snapshots

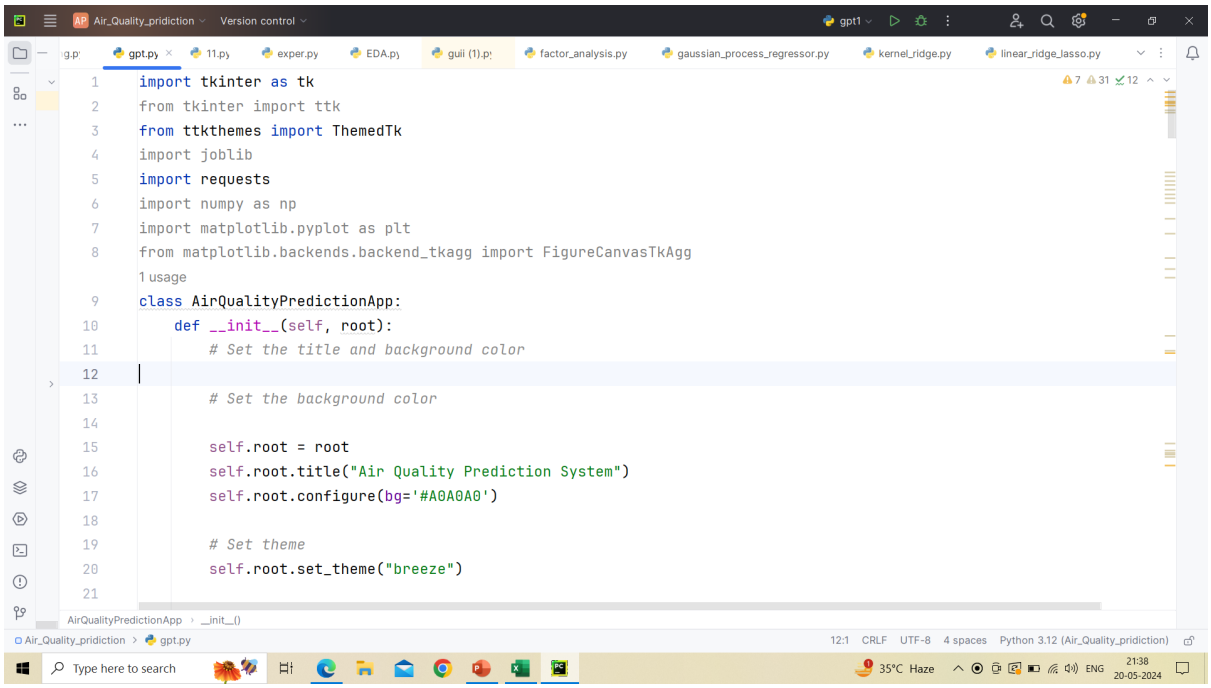
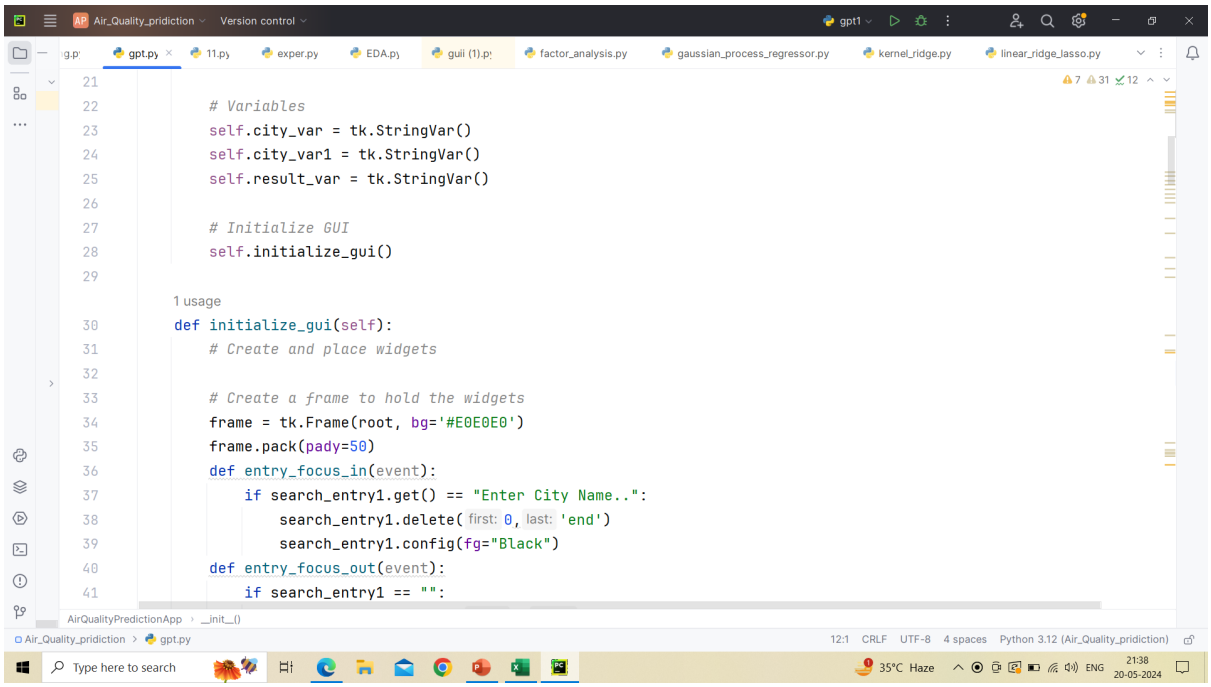
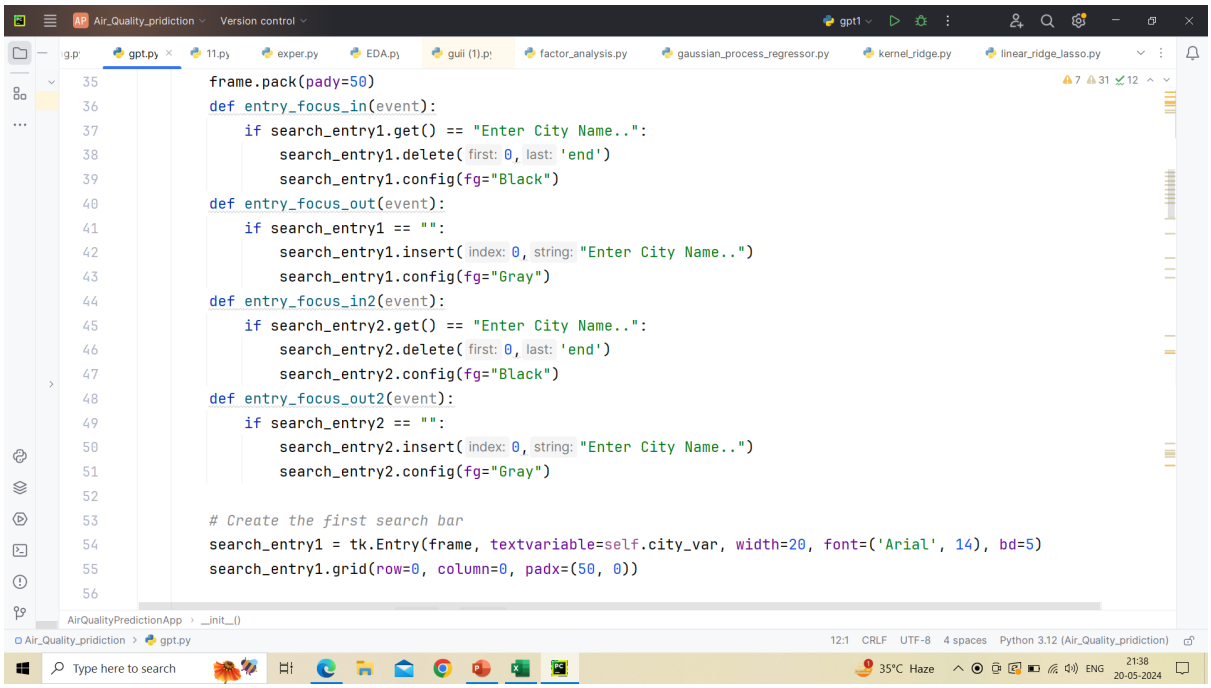


Figure 10.1: User Interface



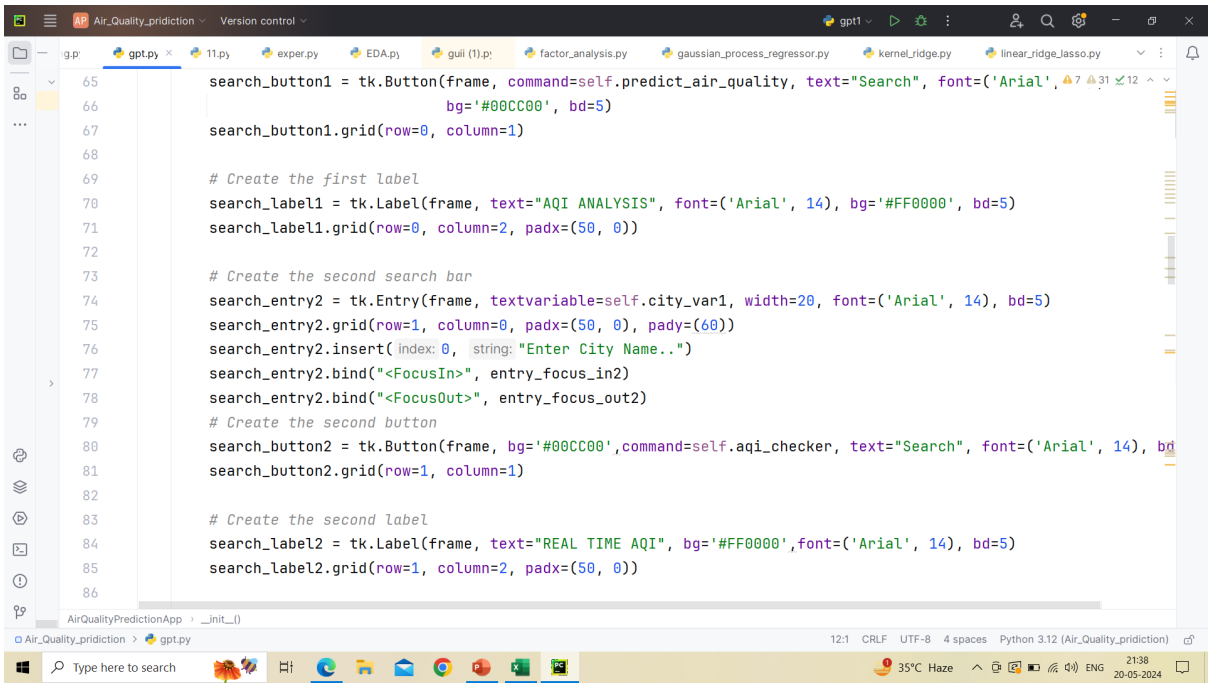
```
        # Variables
        self.city_var = tk.StringVar()
        self.city_var1 = tk.StringVar()
        self.result_var = tk.StringVar()

        # Initialize GUI
        self.initialize_gui()

    1 usage
    def initialize_gui(self):
        # Create and place widgets

        # Create a frame to hold the widgets
        frame = tk.Frame(root, bg='#E0E0E0')
        frame.pack(pady=50)
        def entry_focus_in(event):
            if search_entry1.get() == "Enter City Name..":
                search_entry1.delete( first: 0, last: 'end')
                search_entry1.config(fg="Black")
        def entry_focus_out(event):
            if search_entry1 == "":
```

Figure 10.2: User interface



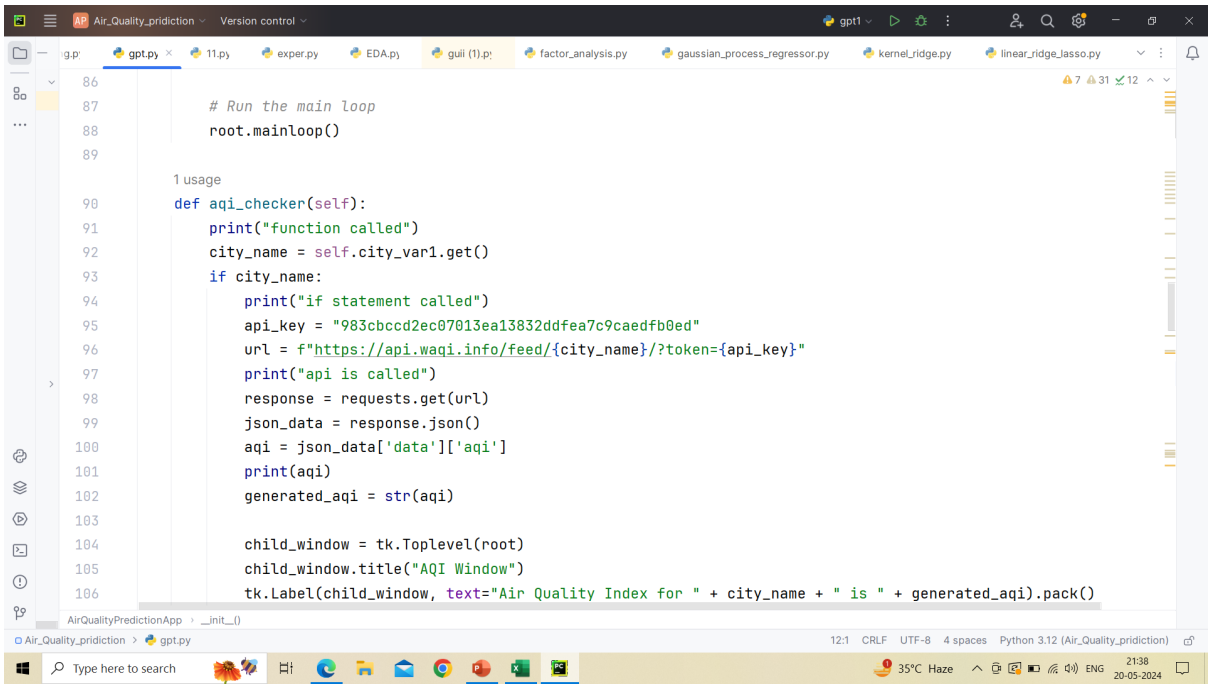
```
        frame.pack(pady=50)
        def entry_focus_in(event):
            if search_entry1.get() == "Enter City Name..":
                search_entry1.delete( first: 0, last: 'end')
                search_entry1.config(fg="Black")
        def entry_focus_out(event):
            if search_entry1 == "":
                search_entry1.insert( index: 0, string: "Enter City Name..")
                search_entry1.config(fg="Gray")
        def entry_focus_in2(event):
            if search_entry2.get() == "Enter City Name..":
                search_entry2.delete( first: 0, last: 'end')
                search_entry2.config(fg="Black")
        def entry_focus_out2(event):
            if search_entry2 == "":
                search_entry2.insert( index: 0, string: "Enter City Name..")
                search_entry2.config(fg="Gray")

        # Create the first search bar
        search_entry1 = tk.Entry(frame, textvariable=self.city_var, width=20, font=('Arial', 14), bd=5)
        search_entry1.grid(row=0, column=0, padx=(50, 0))
```

Figure 10.3: Data Processing



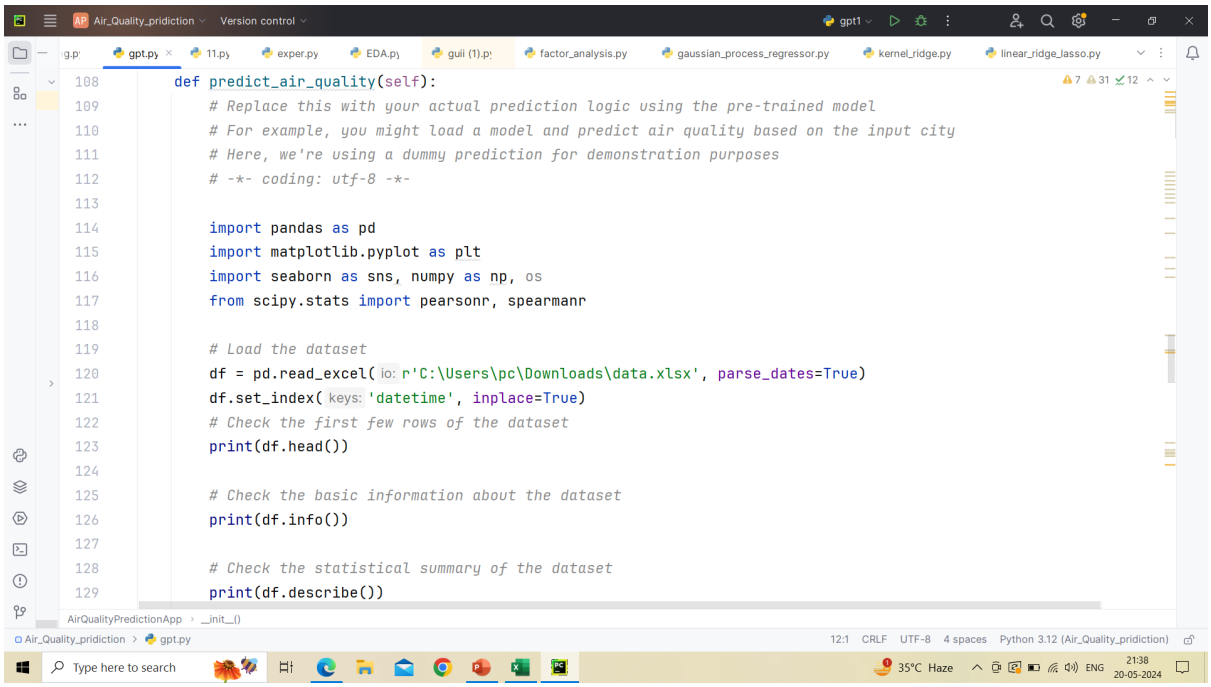
```
search_button1 = tk.Button(frame, command=self.predict_air_quality, text="Search", font=('Arial',
                           bg='#00CC00', bd=5)
search_button1.grid(row=0, column=1)

# Create the first label
search_label1 = tk.Label(frame, text="AQI ANALYSIS", font=('Arial', 14), bg='#FF0000', bd=5)
search_label1.grid(row=0, column=2, padx=(50, 0))

# Create the second search bar
search_entry2 = tk.Entry(frame, textvariable=self.city_var1, width=20, font=('Arial', 14), bd=5)
search_entry2.grid(row=1, column=0, padx=(50, 0), pady=(60))
search_entry2.insert( index: 0, string: "Enter City Name..")
search_entry2.bind("<FocusIn>", entry_focus_in2)
search_entry2.bind("<FocusOut>", entry_focus_out2)
# Create the second button
search_button2 = tk.Button(frame, bg='#00CC00',command=self.aqi_checker, text="Search", font=('Arial', 14), bd
search_button2.grid(row=1, column=1)

# Create the second label
search_label2 = tk.Label(frame, text="REAL TIME AQI", bg='#FF0000',font=('Arial', 14), bd=5)
search_label2.grid(row=1, column=2, padx=(50, 0))
```

Figure 10.4: User Interface



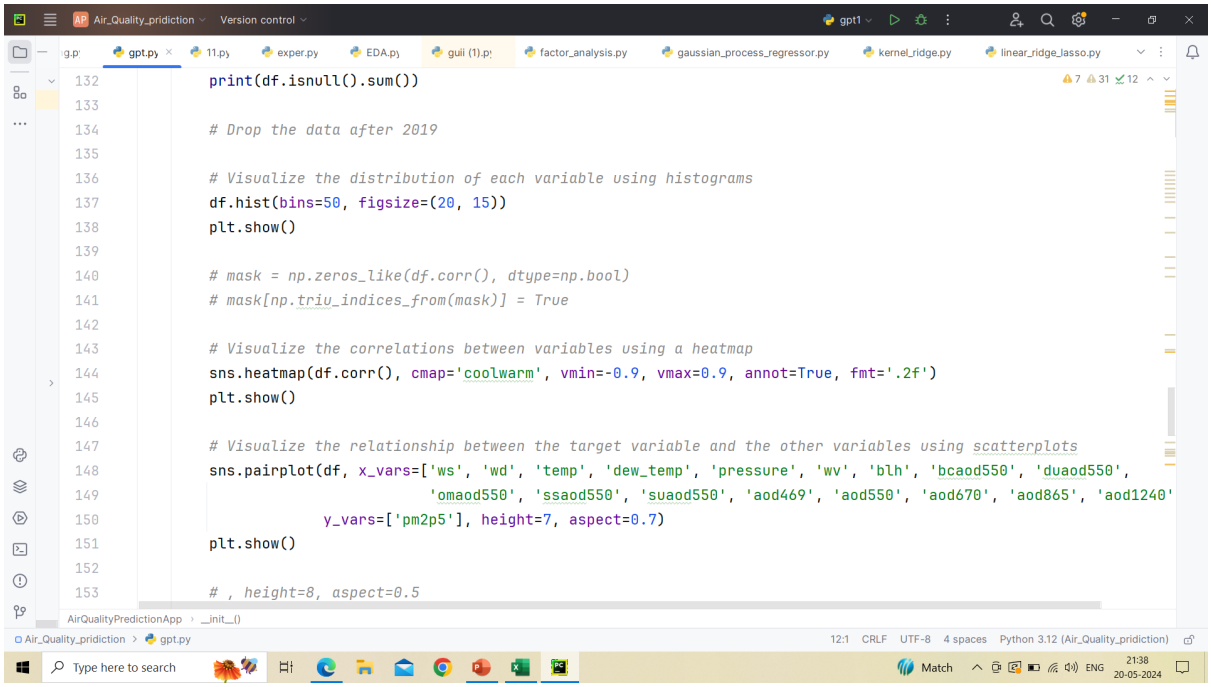
```
# Run the main loop
root.mainloop()

1 usage
def aqi_checker(self):
    print("function called")
    city_name = self.city_var1.get()
    if city_name:
        print("if statement called")
        api_key = "983cbccd2ec07013ea13832ddfea7c9caedfb0ed"
        url = f"https://api.waqi.info/feed/{city_name}/?token={api_key}"
        print("api is called")
        response = requests.get(url)
        json_data = response.json()
        aqi = json_data['data']['aqi']
        print(aqi)
        generated_aqi = str(aqi)

        child_window = tk.Toplevel(root)
        child_window.title("AQI Window")
        tk.Label(child_window, text="Air Quality Index for " + city_name + " is " + generated_aqi).pack()
```

Figure 10.5: Data Processing



```
108    def predict_air_quality(self):
109        # Replace this with your actual prediction logic using the pre-trained model
110        # For example, you might load a model and predict air quality based on the input city
111        # Here, we're using a dummy prediction for demonstration purposes
112        # -*- coding: utf-8 -*-
113
114        import pandas as pd
115        import matplotlib.pyplot as plt
116        import seaborn as sns, numpy as np, os
117        from scipy.stats import pearsonr, spearmanr
118
119        # Load the dataset
120        df = pd.read_excel( io: r'C:\Users\pc\Downloads\data.xlsx', parse_dates=True)
121        df.set_index( keys: 'datetime', inplace=True)
122        # Check the first few rows of the dataset
123        print(df.head())
124
125        # Check the basic information about the dataset
126        print(df.info())
127
128        # Check the statistical summary of the dataset
129        print(df.describe())
```

Figure 10.6: Data processing



```
132        print(df.isnull().sum())
133
134        # Drop the data after 2019
135
136        # Visualize the distribution of each variable using histograms
137        df.hist(bins=50, figsize=(20, 15))
138        plt.show()
139
140        # mask = np.zeros_like(df.corr(), dtype=np.bool)
141        # mask[np.triu_indices_from(mask)] = True
142
143        # Visualize the correlations between variables using a heatmap
144        sns.heatmap(df.corr(), cmap='coolwarm', vmin=-0.9, vmax=0.9, annot=True, fmt='.2f')
145        plt.show()
146
147        # Visualize the relationship between the target variable and the other variables using scatterplots
148        sns.pairplot(df, x_vars=['ws', 'wd', 'temp', 'dew_temp', 'pressure', 'wv', 'blh', 'bcaod550', 'duaod550',
149                                 'omaod550', 'ssaod550', 'suaod550', 'aod469', 'aod550', 'aod670', 'aod865', 'aod1240'
150                     y_vars=['pm2p5'], height=7, aspect=0.7)
151        plt.show()
152
153        # , height=8, aspect=0.5
```

Figure 10.7: Data visualization

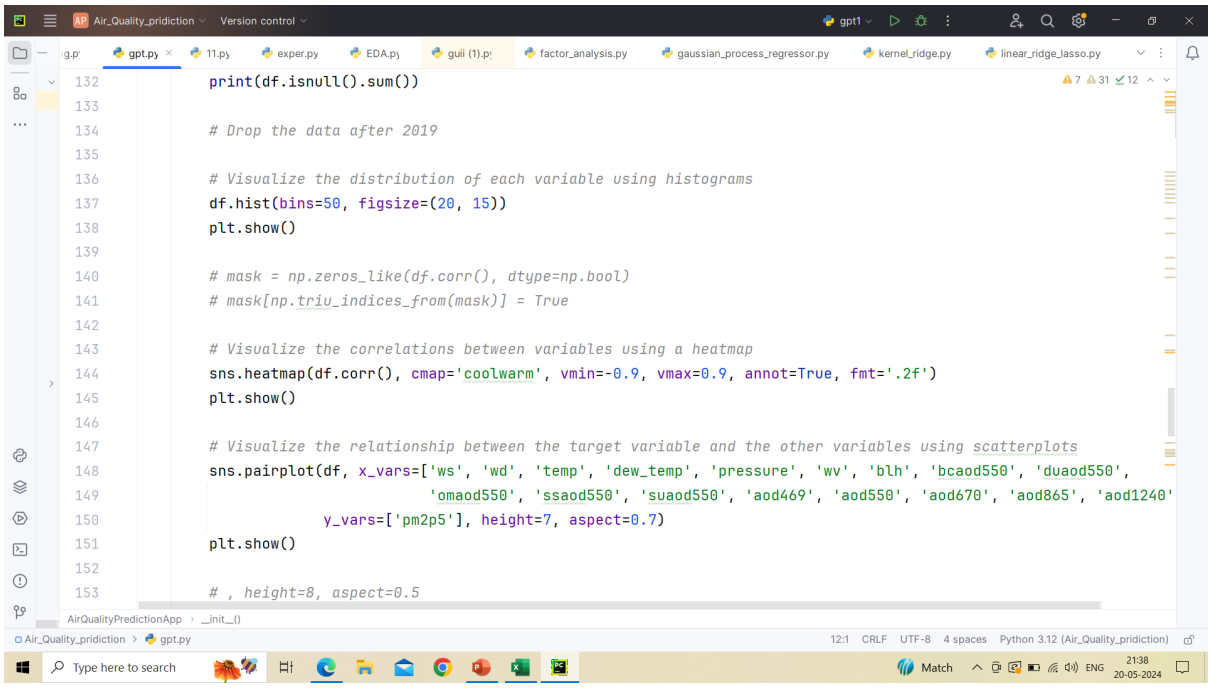


Figure 10.8: Data Visualization

Output:

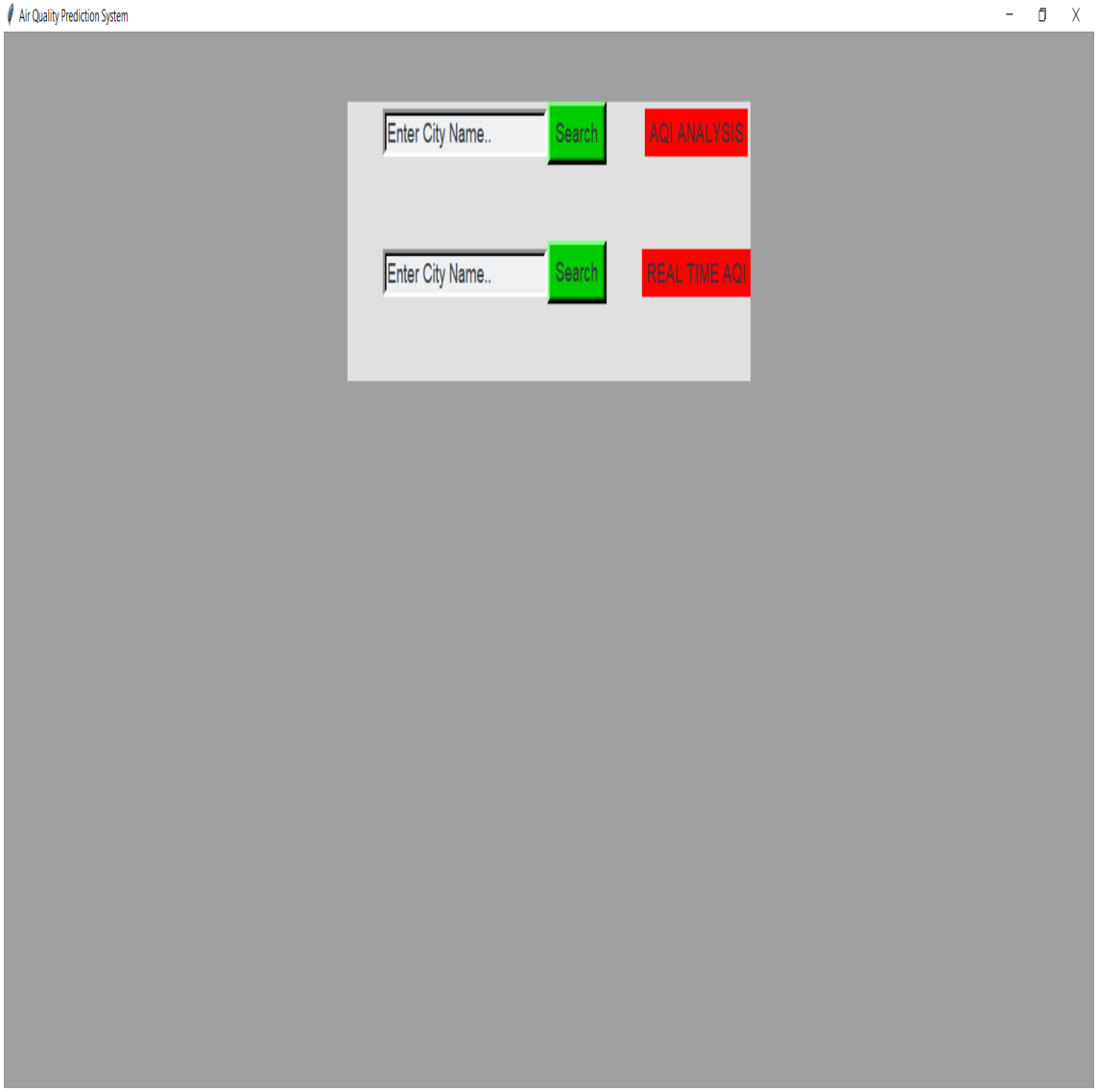


Figure 10.9: User Interface

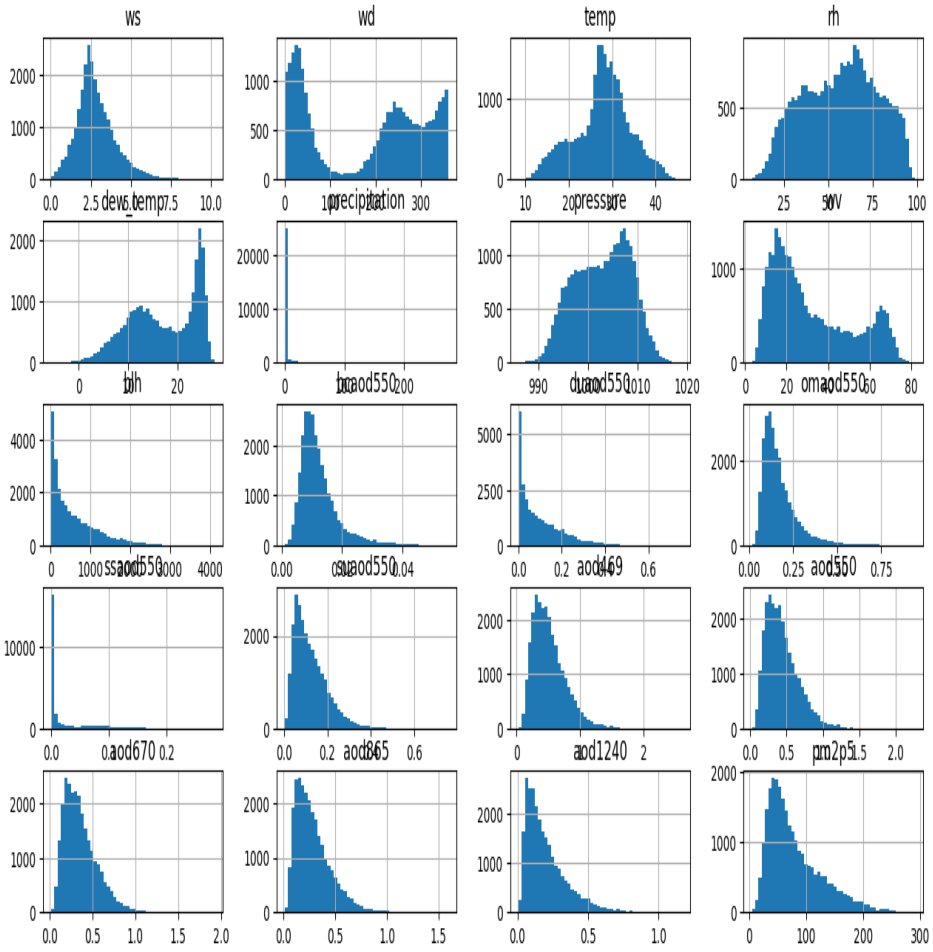


Figure 10.10: Data Histogram Chart

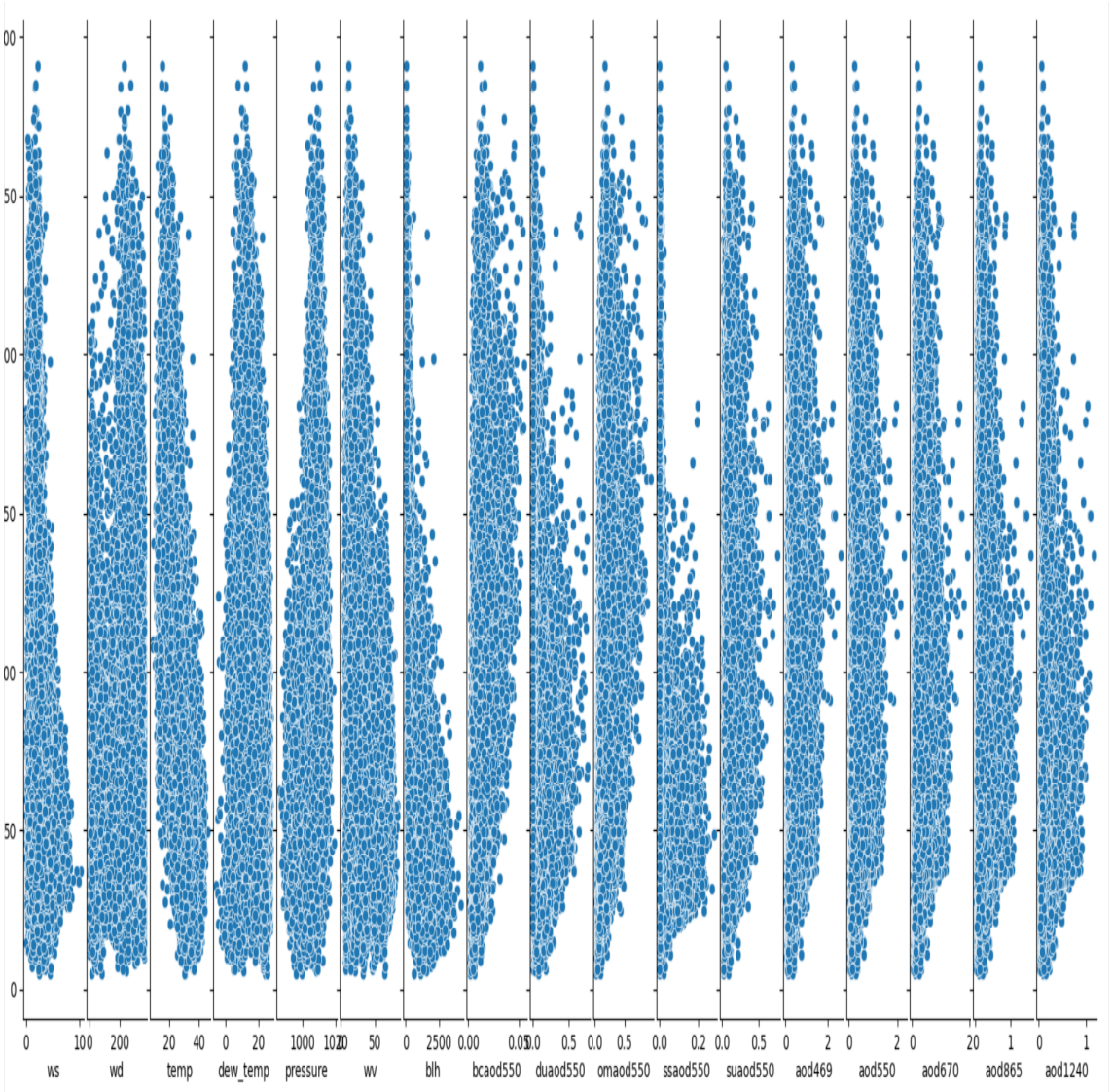


Figure 10.11: Exploratory Data Analysis

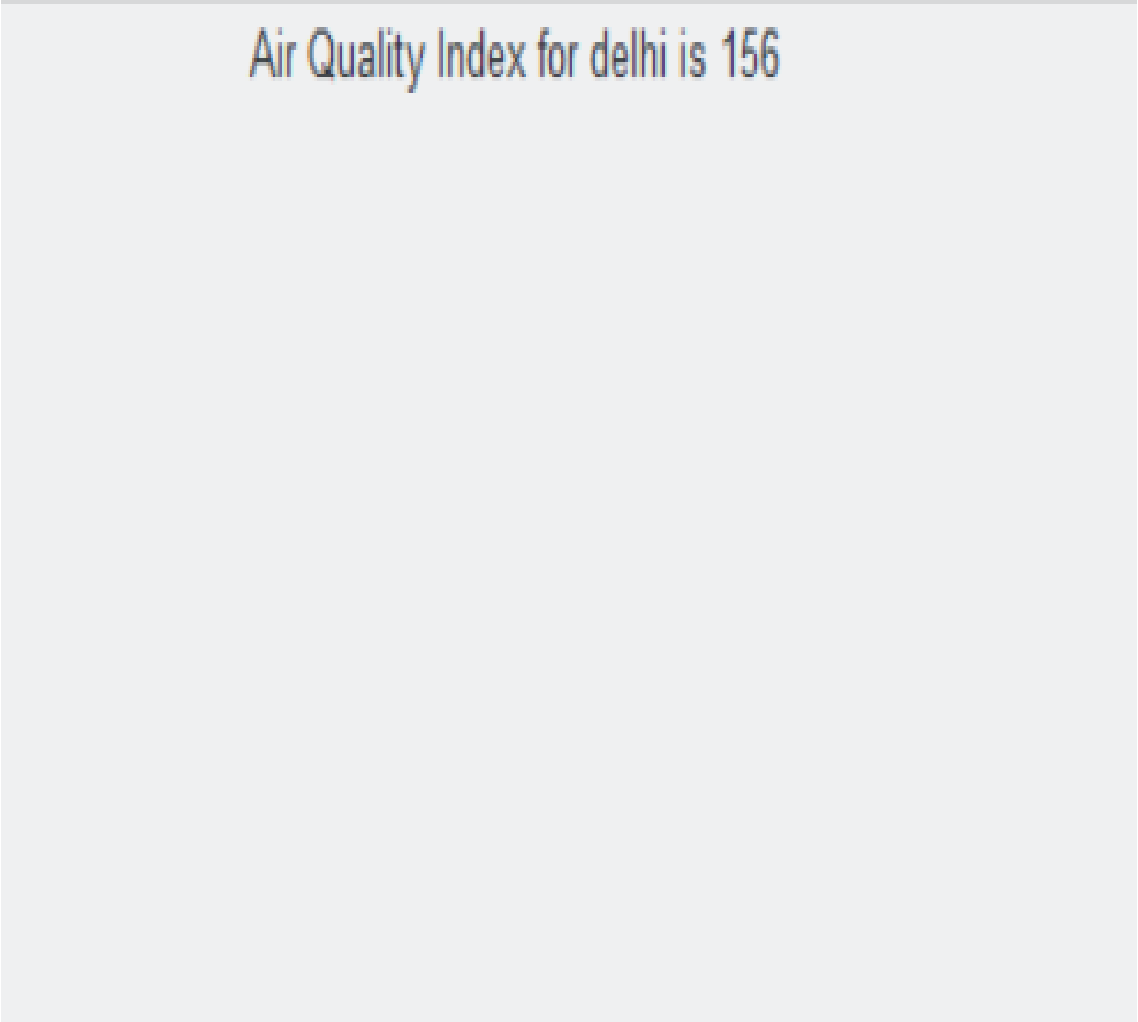


Figure 10.12: API Calling

Chapter 11

Conclusion

The conclusion of an air quality prediction analysis would typically summarize the findings and implications of the study. Here's a sample conclusion:

"In conclusion, our air quality prediction model demonstrates the potential to accurately forecast pollutant levels based on historical data and meteorological factors. Through the utilization of machine learning algorithms such as neural networks and regression analysis, we have developed a robust framework capable of providing timely and reliable forecasts. These predictions not only aid in informing the public about potential health risks associated with poor air quality but also offer valuable insights for policymakers to implement effective mitigation strategies. However, it's essential to acknowledge the inherent uncertainties and limitations of such models, particularly in complex urban environments with dynamic atmospheric conditions. Future research should focus on refining the model's accuracy by incorporating additional variables and improving spatial resolution. Ultimately, proactive measures informed by accurate air quality predictions are crucial for safeguarding public health and promoting environmental sustainability."

Chapter 12

Bibliography

1. Aarnio Pet al. (2005). The concentrations and composition of and exposure to fine particles (PM2.5) in the Helsinki subway system. Atmospheric Environment, 39(28):5059–5066.

2. Abbey DEet al. (1999). Long-term inhalable particles and other air pollutants related to mortality in nonsmokers. American Journal of Respiratory and Critical Care Medicine, 159(2):373–382. [PubMed]

3. Abello Net al. (2009). Protein tyrosine nitration: selectivity, physicochemical and biological consequences, denitration, and proteomics methods for the identification of tyrosine-nitrated proteins. Journal of Proteome Research, 8(7):3222–3238. [PubMed]

4. Acker Ket al. (1998). Atmospheric research program for studying changing emission patterns after German unification. Atmospheric Environment, 32(20):3435–3443.

5. Adams Het al. (2001). Fine particle (PM2.5) personal exposure levels in transport microenvironments, London, UK. Science of the Total Environment, 279(1–3):29–44. [PubMed]

6. Aesif SWet al. (2009). In situ analysis of protein S-glutathionylation in lung tissue using glutaredoxin-1-catalyzed cysteine derivatization. American Journal of Pathology, 175(1):36–45. [PMC free article] [PubMed]

7. Akerstrom Met al. (2013). Associations between urinary excretion of cadmium and proteins in a nonsmoking population: renal toxicity or normal physiology? Environmental Health Perspectives, 121(2):187–191. [PMC free article] [PubMed]

8. Akesson A, Julin B, Wolk Aet al (2008). Long-term dietary cadmium intake and postmenopausal endometrial cancer incidence: a population-based prospective cohort study. Cancer Research, 68(15):6435–6441. [PubMed]

9. Akinaga LMYet al. (2009). Effects of chronic exposure to air pollution from Sao Paulo city on coronary of Swiss mice, from birth to adulthood. Toxicologic Pathology, 37(3):306–314. [PubMed]

10. Alberg Tet al. (2011). Nitrogen dioxide: no influence on allergic sensitization in an intranasal mouse model with ovalbumin and diesel exhaust particles. Inhalation Toxicology, 23(5):268–276. [PubMed]

11. Alexis NEet al. (2010). Low-level ozone exposure induces airways inflammation and modifies cell surface phenotypes in healthy humans. Inhalation Toxicology, 22(7):593–600. [PMC free article] [PubMed]

12. Andersen ZJet al. (2008b). Ambient air pollution triggers wheezing symptoms in infants. Thorax, 63(8):710–716. [PubMed]

13. Dorado-Martínez Cet al. (2001). Effects of different ozone doses on memory, motor activity and lipid peroxidation levels, in rats. International Journal of Neuroscience, 108(3–4):149–161. [PubMed]

14. Downs SHet al. (2007). Reduced exposure to PM10 and attenuated age-related decline in lung function. New England Journal of Medicine, 357(23):2338–2347. [PubMed]

15. Duffin R, Mills NL, Donaldson Ket al (2007). Nanoparticles – a thoracic toxicology perspective. Yonsei Medical Journal, 48(4):561–572. [PMC free article] [PubMed]

16. Ebelt Set al. (2001). Air quality in postunification Erfurt, East Germany: associating changes in pollutant concentrations with changes in emissions.

17. Environmental Health Perspectives, 109(4):325–333. [PMC free article] [PubMed]EC (2001). Ambient air pollution by polycyclic aromatic hydrocarbons (PAH). Position paper.

18. Edwards J, Walters S, Griffiths RKet al (1994). Hospital admissions for asthma in preschool children: relationship to major roads in Birmingham, United Kingdom. Archives of Environmental Health, 49(4):223–227. [PubMed]

19. Edwards RD, Jantunen MJet al (2001). Benzene exposure in Helsinki, Finland. Atmospheric Environment, 35(8):1411–1420.

20. Edwards SCet al. (2010). Prenatal exposure to airborne polycyclic aromatic hydrocarbons and children's intelligence at 5 years of age in a prospective cohort study in Poland. Environmental Health Perspectives, 118(9):1326–1331. [PMC free article] [PubMed]

Appendix A

Basic Terminologies

In an appendix for an air quality prediction project, you would typically include supplementary material that supports and enhances the main body of your project report. Here are some suggestions for what you might include:

1. Data Sources: Provide detailed information about the sources of your air quality data. This could include descriptions of the monitoring stations, data collection methods, and any pre-processing steps applied to the raw data.

2. Feature Engineering: Explain the process of selecting and transforming the features used in your prediction model. Include any domain-specific knowledge or insights that guided your feature selection process.

3. Model Selection: Describe the different machine learning models you experimented with, along with their architectures, hyperparameters, and performance metrics. You could also include summaries of training/validation/test splits used for model evaluation.

4. Evaluation Metrics: Discuss the choice of evaluation metrics used to assess the performance of your prediction model. Explain why these metrics are appropriate for measuring the accuracy and reliability of air quality predictions.

5. Results and Discussion: Present additional results and analysis that didn't fit within the main body of your report. This could include visualizations, tables, or supplementary experiments that provide further insights into your model's performance.

6. Implementation Details: Share any technical details or code snippets

related to the implementation of your prediction model. This could be helpful for readers who want to replicate or extend your work.

7. Limitations and Future Work: Discuss any limitations or constraints of your air quality prediction model, as well as opportunities for future research and improvement. This could include suggestions for incorporating additional data sources, refining model architectures, or exploring alternative prediction techniques.

8. References: Provide a list of references cited throughout the appendix, including relevant research papers, datasets, and software libraries.

9. Data Collection: The first step in air quality prediction is to collect relevant data. This includes historical air quality data collected from monitoring stations, as well as meteorological data such as temperature, humidity, wind speed, and wind direction. Satellite imagery and remote sensing data can also provide valuable information.

10. Data Preprocessing: Once collected, the data needs to be cleaned and preprocessed. This involves handling missing values, outliers, and inconsistencies in the data. Additionally, data from different sources may need to be standardized or normalized to ensure compatibility.

11. Feature Selection/Extraction: In this step, relevant features or variables that influence air quality are selected or extracted from the data. This can involve statistical analysis, correlation analysis, or domain knowledge to identify the most important predictors.

12. Model Development: Various machine learning and statistical models can be used for air quality prediction. Commonly used techniques include regression models, time series analysis, and neural networks. These models are trained using historical data, with the aim of learning the relationship between the selected features and air quality parameters.

13. Model Evaluation: Once trained, the predictive performance of the models is evaluated using validation data. Metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and correlation coefficients are commonly used to assess the accuracy of the predictions.

14. Model Optimization: The performance of the models can be further improved through optimization techniques such as hyperparameter tuning, feature engineering, and ensemble methods.

15. Deployment: Once a satisfactory model is developed and evaluated, it can be deployed for real-time air quality prediction. This involves integrating the model into a software system or platform that can receive input data in real-time and provide predictions to end-users.

16. Monitoring and Feedback: Continuous monitoring of the model's performance is essential to ensure its accuracy and reliability. Feedback from users and stakeholders can also be used to improve the model over time.

By including these elements in your appendix, you can provide readers with a comprehensive understanding of your air quality prediction project and enable them to build upon your work in their own research or applications.

BIO-DATA

Sagar Sen

- **Personal Information**
  - Student Name : Sagar Sen
  - Email : sagar2110003-d@akgec.ac.in
  - Student Number : 2110003-d
  - Roll No. : 2100270109009
  - Section : CSE-1
- **Educational Qualification**
  - College : Ajay Kumar Garg Engineering College
  - Course : Bachelor of Technology(CSE)
  - Session : 2021-24
- **Tech Stack**
  - Python, Django, HTML, CSS
- **Contribution to Project**
  - Development of Frontend
  - Report formating
- **Learning**
  - Concept of Django and user interface design
  - Leadership
  - Team Working
  - Management

Vikash

- **Personal Information**
  - Student Name : Vikash
  - Email : vikash2110002-d@akgec.ac.in
  - Student Number : 2110002-d
  - Roll No. : 2100270109012
  - Section : CSE-1
- **Educational Qualification**
  - College : Ajay Kumar Garg Engineering College
  - Course : Bachelor of Technology(CSE)
  - Session : 2021-24
- **Tech Stack**
  - Python, MySQL, SQLite, Pandas
- **Contribution to Project**
  - Data Analysis
  - Integration Part
- **Learning**
  - How we manage data sets part and maintaining the modulartity
  - Concept of python data analysis libraries like NumPy, Scipy, Pandas
  - Leadership
  - Team Working
  - Management

Shivam Saini

- **Personal Information**
  - Student Name : Shivam Saini
  - Email : shivam2110008-d@akgec.ac.in
  - Student Number : 2110008-d
  - Roll No. : 2100270109009
  - Section : CSE-1
- **Educational Qualification**
  - College : Ajay Kumar Garg Engineering College
  - Course : Bachelor of Technology(CSE)
  - Session : 2021-24
- **Tech Stack**
  - Python, Django, API
- **Contribution to Project**
  - Development of API
  - Integration Part
- **Learning**
  - How we manage backend part
  - Concept of API
  - Leadership
  - Team Working
  - Management

Mahendra Pratap Singh

- **Personal Information**
  - Student Name : Mahendra Pratap Singh
  - Email : mahendra2110001-d@akgec.ac.in
  - Student Number : 2110001-d
  - Roll No. : 2100270109005
  - Section : CSE-1
- **Educational Qualification**
  - College : Ajay Kumar Garg Engineering College
  - Course : Bachelor of Technology(CSE)
  - Session : 2021-24
- **Tech Stack**
  - Python, HTML, CSS
- **Contribution to Project**
  - Documentation work
  - Report editing
- **Learning**
  - Concept of Python
  - Leadership
  - Team Working
  - Management