Air Quality Index Prediction Tool with User Alerts

Raniel Maranan
Master of Engineering: Artificial Intelligence for Smart Sensors and Actuators
Deggendorf Institute of Technology
Cham, Germany
raniel.maranan@stud.th-deg.de

Manojkumar Mohankumar
Master of Engineering: Artificial Intelligence for Smart Sensors and Actuators
Deggendorf Institute of Technology
Cham, Germany
manoj.mohankumar@stud.th-deg.de

Dalmeet Singh
Master of Engineering: Artificial Intelligence for Smart Sensors and Actuators
Deggendorf Institute of Technology
Cham, Germany
dalmeet.singh@stud.th-deg.de

Nishanth Sundaran
Master of Engineering: Artificial Intelligence for Smart Sensors and Actuators
Deggendorf Institute of Technology
Cham, Germany
nishanth.sundaran@stud.th-deg.de

***Abstract* —** Over 5000 lives are taken daily because of illnesses related to pollution and continues to be an important issue worldwide. Air pollution is considered a major problem in some cities such as Dammam, Lahore, Delhi. This study uses machine learning techniques, with a focus on ARIMAX, Auto-ARIMA, SARIMAX to predict air pollution using advances in information and computing technology. The datasets used here contain daily pollutant data such as Particle matter 2.5 (PM2.5), Particle Matter 10(PM10), Ozone 3(O3), Carbon Monoxide (CO), Sulfur Dioxide (SO2), and Nitrogen Dioxide (NO2) for predicting the air quality predictions of overall India and a particular city. During data analysis the dataset showed a clear seasonality and trend during some months. Assuming evaluating models like ARIMA, Auto-ARIMA, SARIMAX, the project concluded that SARIMAX is the best model. Statistical calculations like Root mean square error (RMSE) and mean absolute error (MAE) are used to check the SARIMAX'S prediction accuracy. Beyond AQI prediction, the group introduced an additional assignment that examines the possible impacts of pollution when an industry, power plants and factory is established in a particular city or all over India. The study promises to provide a complete understanding about the environmental and health effects by increasing the scope to include the effects of industries. Using the user-friendly Tkinter interface helps accessibility when showing AQI forecasts and recommendations.

Keywords— SARIMAX, AQI, Industrial pollution calculation, forecasting, Time series data, RMSE.

I. Introduction

India is one of the most polluted countries and has become known for its poor air quality. Statistics show the immediate need for including measures to reduce the effects of pollution on nature and health. PM2.5, PM10, SO2, NO, CO and O3 were the important factors which influenced the AQI. The World Health Organization has related air pollution with approximately seven million deaths annually and highlighted the significance of pollutants such as particulate matter (PM), Ozone(O3), Nitrogen dioxide (NO2), and Sulphur dioxide (SO2).

Public health can be at risk from particulate matter (PM) that comes from sources like coal-fired power plants and automobiles. Therefore, it is necessary to monitor and regulate atmospheric PM2.5 levels closely to maintain a safe environment and develop health.

This study uses modern techniques for the analysis and forecasting of time series in India like ARIMA, Auto-ARIMA, SARIMAX. By developing a model, this case study aims to examine and predict the air quality index with the help of factors such as traffic congestion, Industrial pollution and the weather. India's overall future trajectory along with one city can be understood fully by using the SARIMAX model, thereby identifying the relationships among time series data.

Beyond simple trend identification, it is a powerful technique which can suggest how to analyze and plan for the future of the city. THE ARIMA model orders can be verified with the graphs which represent the Autocorrelation function (ACF) and Partial Autocorrelation function (PACF). Then, the team used Auto-ARIMAX which helped to further clarify the model parameters. Unfortunately, the model obtained using Auto-ARIMAX is not good for the prediction. To get a good prediction. The model was further hyper tuned with other parameters and was built with the new p, d, q values which have been created.

The model created in this study can help any policy makers to improve the AQI of a city as well as the overall India and can help the government or boards to improve the health and environmental problems by considering all the pollutant factors. Finally, the report gives an outlook for India's future and can be helpful for making policy decisions and planning.

A. Case Study Overview

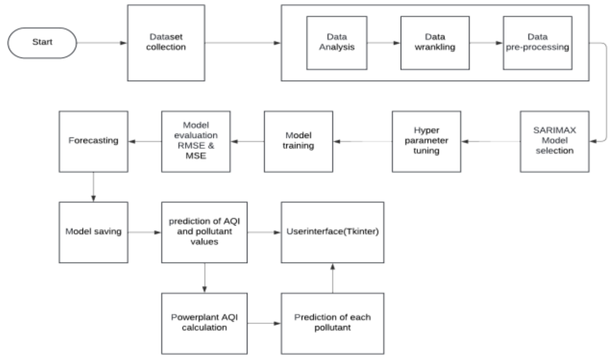


Figure 1- Case Study Overview

The primary objective of the case study is to develop an AQI prediction tool that provides users with an understanding of the future effect of pollution. First, a dataset containing various pollutants needs to be collected to achieve this task. Data wrangling, pre-processing, and exploratory data analysis are then applied to the dataset. The unnecessary columns in the dataset are removed all through this process.

The dataset was modified to use monthly averages by combining daily data to help in AQI prediction. The average monthly AQI of each city is added into a new variable named India_AQI. Three cities with a high missing rate have been excluded to find and eliminate missing values. A seasonal decomposition study is carried out on the dataset to examine the residual components, trends, and seasonality.

After that, the dataset is divided 80/20 into training and testing sets to focus on forecasting. The group decides to apply SARIMAX modeling because of the observed seasonality. For determining suitable p, d, and q values, Auto ARIMA is initially employed, but the outcomes are found to be insufficient. RMSE and MSE strategies are used for evaluating the model, and additional hyperparameter adjustment is done. This process is continued for each pollutant like O3, SO2, NO2, NO, CO, PM2.5, PM10. A graphical user interface (GUI) is employed to load all the model of forecasts that are saved. In addition to that the UI will show the predicted values of each pollutant with the help of some calculations when a power plant or factory is added.

II. REQUIRMENTS

For the project of air quality index prediction tool with user alerts the team brainstormed specific requirements and goals to allow the project to be successful. The main objective of the project is to predict the air quality index for future dates and apply user alerts based on the category of the air quality index.

Within the system, the group would like to accomplish these requirements:

- To create a tool that can generate recommendations or suggestions regarding Air Quality Index that can help decision makers with regarding environmental policies.
- To use precise data sets that will be trained easily that will allow an accurate comparison.
- To add an agricultural factor into the predicted value so it can be more than just a prediction tool.
- Create a user interface that will allow easier accessibility and have a proper display.

A. *Data Sets*

1) *Prediction Tool*

a) Data Collection

To forecast AQI and pollutant values, the dataset initially used was specific to Washington, DC. An important issue occurred from the small dataset was that it can only give AQI values for only one state. The absence of characteristics for individual pollutants was also a challenge to the prediction. This shortage of data makes it difficult to forecast the features for each pollutant. In addition to that, the absence of time granularity in the dataset was another major issue. The lack of a time series element made it more difficult for the project to recognize patterns and trends in the pollutants over time and prevented the building of a more dynamic prediction model.

The dataset used in this project is the India_AQI (2015-2020) which was created with the help of CPCB website and is available at cpcb.nic.in. This is a website under the administration of the Indian pollution control board and gives information on air quality around India which includes all metro cities. The real time air quality data from around thousands of monitoring stations across the cities are measured, which contains all the pollutants like Ozone, PM10, PM2.5, CO, NO2, SO2. This data set consists of 628458 instances and 13 attributes containing categorical and numerical data. Missing data are inputted using the mean for each variable.

b) Data Analysis

The initial experience with the datasets obtained for the project shows a percentage of missing values could make the accuracy of the predictions and conclusions harder. To avoid this, the team began the journey of exploratory data analysis (EDA), data wrangling and pre-processing. A more detailed analysis shows the enormous asymmetry in the AQI columns with the Chennai datasets showing the highest amount of complete data. On the other hand, the missing values were an issue for other cities.

Simplifying the dataset by including the most essential columns for the project was the only way to overcome the challenge. As a result, the columns named City, Date, AQI, and AQI_Bucket were selected and deleted any irrelevant or unnecessary information. The main aim of these steps was to create more relevant and precise datasets for this case study.

To better the ability to compare AQI values between cities, the team investigated data transformation after this filtering. Because the dataset was organized daily, the group converted it to a monthly format, making it easier to handle. For this modification, an entire month's AQI averages of each city were used. In addition to that the group has added a column called India_AQI which gives average AQI for the cities on every date. However, it gives a standard prediction for representative indication of the overall AQI.

Our forecasting model initially focused on the overall AQI. By choosing Chennai from the dataset, the team improved the model additionally to predict the Chennai's AQI.

2) Time series decomposition plot

Data cleaning, transformation, and preparation for further analysis and modeling are known as processing in time series datasets. The missing values in the time-series data were initially identified followed by plotting with graphs with the goal of identifying the appropriate attribution method and measuring the level of missingness. The time series dataset is divided into three parts like trends, seasonality and residual components with the help of time series decomposition. By dividing the time series into its components makes it easier to identify hidden patterns and trends in the raw data. The three main components of time series decompositions are:

- Trend: The long-term direction or pattern of data over time can be seen by the trend component. In time, it could increase, decrease, or remain unchanged.

- Seasonal: Periodic pattern of data or periods, such as those occurring daily, weekly, or monthly, are characterized in the seasonal element.

- Residual: The residual component in data denotes any excess volatility or noise that cannot be explained with a trend or season.

The decomposition plot *Figure 2* gives observed values in the data.

1. Trend: By looking at the graph, the group can see that there is a less clear trend. The main reason for this is the restriction on pollution by the government from 2015 and the last surge in trend is due to the pandemic.
2. Seasonality: The graph shows a clear seasonality throughout the year especially in October and January. By analyzing, the team concluded that the sudden surge during these months is due to the winter aversion and valley effect, burning of fuels for heating and Diwali.
3. Residual: The data points which follows a clear trend as well as seasonality are plotted in the residual graph

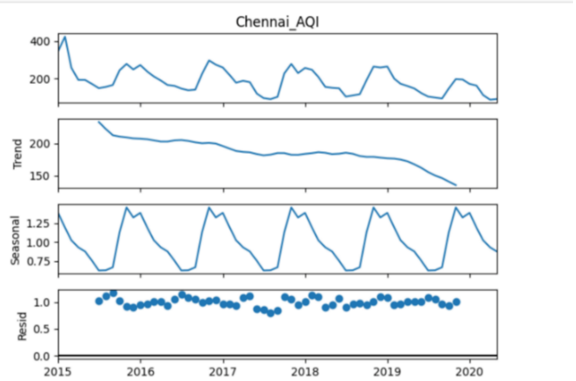


Figure 2- Decomposition

3) Agricultural Tool

Regarding the case study, the group does not only have to predict AQI values but also apply an agricultural factor with the prediction. This feature will help environmentalists possess a tool that will show how certain factors can affect the AQI rating in certain cities/regions by incorporating an additional agricultural factor. The agricultural effect is a very broad topic, and many factors can be used to modify the AQI rating. The team has decided to use the concept of factories, or some environmental loss as the agricultural factor.

With many agricultural factors that can applied to the AQI rating, the group has decided to use data sets that includes pollutant levels that are being released by different types of factories or power plants or about different kinds of deforestation. The group also will consider data sets that are involved with the number of pollutants a car releases or how many pollutants a tree can absorb and clean the air for the environment.

With the goal to find data that was similar as the groups India's AQI data would be the goal because in the data set files, the top 6 pollutants the group would like to use are recorded with the proper units and included the AQI rating for the day. The group knows that the data sets for the agricultural factors will not include an AQI rating but would have the recording of the pollutants released in a day or year.

The agricultural factor will be affecting the AQI rating by increasing the air pollutant levels within the city or region. Doing so will let users or environmentalists see the changes regarding the additional factors in the affected region. Having the effect of the agricultural factor will increase the AQI rating in a way of how much and how polluted the region will be. This will give the group a better insight into the effectiveness of the agricultural factor.

In addition, the AQI value will not also be affected but also the pollutant levels. Those are the values; the group would like to modify and then used the modified pollutant levels to calculate the new AQI ratings. This will allow the environmentalist to see what pollutants are being affected. Depending on the type of agriculture the group will proceed with, different pollutants will be affected but the group assumes particle matter and ozone will be affected drastically.

B. Models

A new model strategy was used for each pollutant to predict the AQI, O3, SO2, NO2, CO, PM10and PM2.5. Initially, the team built a particular model for forecasting AQI. To expand the scope, accuracy and extra study needs of the project, the group has decided to model each pollutant.

1. ARIMA model: The Autoregressive Integrated Moving Average, or ARIMA model is an analytical technique used for modeling a time

series data and predicting a time series data. The idea of an ARIMA model is that the basic procedure that results in the time series and can be modeled with the combinations of moving average (MA), discriminant(I) and autoregressive (AR) factors. The model contains 3 parameters (p, d, q) where p is the order of auto regressive components, d is the degree of differencing and q is the order of MA.

$$AR = A^t = c + \Phi_1 A_{t-1} + \Phi_2 A_{t-2} + .. + \Phi_p A_{t-p} + \varepsilon_t \quad \text{Eq. 1}$$

$$MA = A^t = c + \Phi_1 A_{\varepsilon t-1} + \Phi_2 A_{\varepsilon t-2} + .. + \Phi_p A_{\varepsilon t-q} \quad \text{Eq. 2}$$

Combination of all three models

$$A^t = c + \Phi_1 A_{t-1} + \Phi_2 A_{t-2} + .. + \Phi_p A_{t-p} + \varepsilon_t + \Phi_1 A_{\varepsilon t-1} + \Phi_2 A_{\varepsilon t-2} + .. + \Phi_p A_{\varepsilon t-q} \quad \text{Eq. 3}$$

2. Auto-ARIMA model: This model is a kind of extension of ARIMA models that can generate the best values for the p, d, q themselves and considered as a good set of values which can minimize the Bayesian information criterion. Unfortunately, it did not work for this project. By doing hyper parameter tuning the team got a better result than the Auto-ARIMA. The values obtained from the hyperparameter tuning were varied for every feature.

3. SARIMAX: The hyper parameter tuned ARIMAX model has been modified by adding a seasonal component and variables in the SARIAMAX. For that, SARIMAX parameters like P, D, Q are added and hyper tuned. These represent the seasonal moving average, seasonal difference and seasonal autoregressive respectively. EachP, D, Q value for each model is different.

C. Softwares

1) Excel

Microsoft Excel is a powerful spreadsheet software application that is part of the Microsoft Office suite. Excel is a versatile tool used for data management, analysis, and visualization across various industries and disciplines. Its user-friendly interface, powerful features, and widespread compatibility contribute to its popularity in both professional and personal settings.

The group used it to analyze the dataset which was used for the model. One can easily edit the data as per the requirement load of the model. The quick functions and logical expressions the group used thoroughly while processing the dataset. Later, it can also be manipulated according to the specific model selection for some future work.

2) Jupyter Notebook

The Jupyter Notebooks are an essential tool which was used in this project. It is mainly used for data study, building models and analysis. These processes can be easily done in the Jupyter Notebook. Especially, the separate kernels in the Jupyter helps in this project to determine the errors and preprocessing easily.

Initially, the group imports the necessary libraries required for the data preprocessing and analysis which includes Pandas, NumPy and scikit-Learn. By using Pandas, the data was loaded into the notebook. The dataset used here is in CSV format.

```
df = pd. read_csv(‘datset_directory’)
df[‘Date’] = pd.to_datetime(df[‘Date’]
```

To understand and analysis the features and structure of the dataset, data exploration was done. With the help of Matplotlib and Seaborn, the team visualized the trends, patterns and relation among pollutants. To get the overall understanding of the data used, the team used Panda’s for summarizing.

```
from statsmodels.tsa. seasonal import
seasonal_decompose
result=seasonal_decompose
(AQi_variable,model=’multiplicative’)
ax=final_df[['AQI_variable']].
plot(figsize=(12,8),grid=True,lw=2,color='Red')
```

Another important function of Jupyter in this project was that the explanations, process and results were stored in the notebook. Also, the dynamic analysis was done with the help of widgets or graphs.

```
predictions.
plot(legend=True)test.plot(legend=True);
```

D. Air Quality Index

Air Quality Index (AQI) is a reporting system that determines the air quality of the air in specific locations like cities or regions. With the AQI rating the system will let the public know what the quality of air in specific places is. Stations collect data of many pollutant types that will then be used to determine the AQI rating. Once the AQI rating has been determined, environmentalists can now notify the public on how the air will affect their health and if it is safe to be outside.

The Air Quality Index (AQI) is a key factor in our case study since it is the numerical value the group is predicting. The group will find the right data sets to use to train so it will result in an accurate prediction. AQI values are not the only data sets the group will be focusing on but also the pollutant levels. The group will use the top 6 contribution pollutants which are Particle matter 2.5 (PM2.5), Particle

Matter 10(PM10), Ozone 3(O3), Carbon Monoxide (CO), Sulfur Dioxide (SO2), and Nitrogen Dioxide (NO2). From research, the group has found that those 6 are the pollutants that would be used to calculate AQI but out of the 6, Particle matter 2.5 (PM2.5), Particle Matter 10(PM10), Ozone 3(O3) are the pollutants that are typically used to rate the AQI rating for the day.

Regarding the regulations of the AQI, the group has found out that different countries would follow the different regulations. The group first believed the world would be following the USA standard as one can see in Figure 3 But after conducting more research especially when the data's sets being used by the group was from India, the group later found out India follows a different set of guidelines. As the user can see in Figure 4 , since India is a much more pullulated place than USA or other parts of the world, their regulations are a bit wider compared to the USA.

These Breakpoints...							...equal this AQI	...and this category
O_3 (ppm) 8-hour	O_3 (ppm) 1-hour[1]	$PM_{2.5}$ (μg/m^3) 24-hour	PM_{10} (μg/m^3) 24-hour	CO (ppm) 8-hour	SO_2 (ppb) 1-hour	NO_2 (ppb) 1-hour	AQI	
0.000 - 0.054	-	0.0 – 12.0	0 - 54	0.0 - 4.4	0 - 35	0 - 53	0 - 50	Good
0.055 - 0.070	-	12.1 – 35.4	55 - 154	4.5 - 9.4	36 - 75	54 - 100	51 - 100	Moderate
0.071 - 0.085	0.125 - 0.164	35.5 – 55.4	155 - 254	9.5 - 12.4	76 - 185	101 - 360	101 - 150	Unhealthy for Sensitive Groups
0.086 - 0.105	0.165 - 0.204	(55.5 - 150.4)[3]	255 - 354	12.5 - 15.4	(186 - 304)[4]	361 - 649	151 - 200	Unhealthy
0.106 - 0.200	0.205 - 0.404	(150.5 - (250.4)[3]	355 - 424	15.5 - 30.4	(305 - 604)[4]	650 - 1249	201 - 300	Very unhealthy
([2])	0.405 - 0.504	(250.5 - (350.4)[3]	425 - 504	30.5 - 40.4	(605 - 804)[4]	1250 - 1649	301 - 400	Hazardous
([2])	0.505 - 0.604	(350.5 - 500.4)[3]	505 - 604	40.5 - 50.4	(805 - 1004)[4]	1650 - 2049	401 - 500	Hazardous

Figure 3 – United States AQI Regulations [1]

O3 (ppm)	PM2.5 (ug/m^3)	PM10 (ug/m^3)	CO (ppm)	SO2 (ppb)	NO2 (ppb)	AQI	Category
0-50	0-30	0-50	0-1	0-40	0-40	0-50	Good
51-100	31-60	51-100	1.1-2	41-80	41-80	51-100	Moderate
101-168	61-90	101-250	2.1-10	81-380	81-180	101-150	Unhealthy for Senestive People
169208	91-120	252-350	10.7-17	381-800	181-280	151-200	Unhealthy
-957	121-250	351-430	17.1-34	801-1600	281-400	201-300	Very unhealthy
748+	250+	430+	34+	1600+	400+	301-500	Hazardous

Figure 4 – India's AQI Regulations [2]

The accuracy of the data sets is very important to the group because with inconstant values then the model will not be completing the requirements. Inconstant can occur within the AQI data because different countries might follow different regulations based on the pollutant levels or AQI rating. To determine if our data are following the same regulations for each category the group will look over the values used from past pollutant levels and AQI ratings, to see if it matches the assumed regulations which is India's.

A way for the group to confirm the accuracy of the predicted values, the group can predict AQI ratings and polluted values from previous days and match it with the record data that can be found on the internet. From the actual value to the predicted value the group can determine if the percentage of error is within the range of the required percent of error the group requires.

When working with the AQI data sets for India, there are different models that can be used when working with a period, seasonal pattern and residual patterns. After researching different types of models to run with the data set the group is using, the group has decided to go with the model of SARIMAX. SARIMAX is a good fit for our data sets because the model allows the group to be trained in the requirements the group wanted to complete. Especially since the data sets the group is using have a timeline from 2015 – 2020, has multiple seasonal patterns occurring during the timer periods.

AQI does not only get affected with air pollutant levels but also with the climate change occurring within the city or region. Wind speed, precipitation, humidity and pressure are the climate changes that are typically mentions when looking at the weather and those factors does affect the AQI level but with the groups case study, the group has decided now to add those effect when modifying the AQI value because the group is focusing on having agricultural factors affecting the AQI rating rather than climate change.

E. GUI

For getting AQI information from the saved model, the requirement of a clean and seamless user interface with GUI is check box. The GUI has an option for users to input the desired date. Depending upon the user input, GUI will initialize the saved model which is in pickle format. The GUI will show the AQI, pollutant values and suggestions based on the AQI values. The suggestion will include the quality of the air and what activity is suggested. Also, GUI will show the results of potential impact of a factory or power plant on the Air quality. The GUI requires an error handling method to check the input format of the date entered and to check whether the input date falls within the extended date.

III. Concept

A. Prediction Tool

The AQI forecasting and suggestion tool leveraging SARIMA models provides accurate predictions of various pollutants including AQI, NO, SO2, CO, O3, PM2.5 and PM10. The foundation of the tool lies in the utilization of SARIMA model which is trained for AQI and every pollutant. These models which is saved as pickle file are trained on historical Air quality data of India, capturing the seasonal trends. By loading these trained models, the tool ensures reliable and accurate forecast of user input date. For comprehensive predictions the time index is extended from January 2015 to December 2025 of Chennai City.

The tool has an easy-to-use graphical interface that allows users to enter a specific date (spelt out in the format YYYY-MM-DD) for which they would want an air quality forecast. The tool computes and displays predicted air quality levels and accompanying suggestions for each pollutant upon clicking the "Get Forecast and Suggestions" button. Beyond just producing numerical results, the application can also anticipate air quality situations by classifying predicted AQI values into predetermined ranges and giving users clear and concise information about the conditions of the air. The prediction for each pollutant is complemented with recommendations that are unique to the anticipated levels, assisting users in making well-informed choices.

The AQI Forecast and Suggestion Tool takes a comprehensive approach that includes practical recommendations in addition to numerical forecasts. If the AQI forecast, for example, shows readings in the "Good" range (0-50), users are advised to take advantage of the fresh air while minimizing their influence. Higher AQI readings, on the other hand, are associated with health advice, such special measures for those with respiratory disorders or serious effects on those who already have illnesses.

The fact that the equipment can handle a variety of pollutants adds to its relevance. The prediction for each pollutant is determined separately, considering its distinct properties and sources. This guarantees a thorough and detailed picture of the condition of the air quality, enabling users to rank treatments according to pollutants of concern. The tool also provides the user with information about the change in AQI and pollutants due to the potential impact of new factories or industries on the local air quality.

B. Agricultural Factor

As for one of the requirements, the group wanted to add an agricultural factor with the predicted value. With many possible factors that can be used, the group was able to find data sets that focused on industrial factories. The data sets that were founded by the group were the perfect information to use to train and predict as an agricultural factor because of the variety pollutant data that has been recorded and with many different industries options. The only issue it had was the values that were given were in tons/year or kg/year and the group believed they would be able to convert those values to the respected units based on the pollutant units. The group decided to start with 1 or 2 pollutant types first which will allow the group to focus on perfecting the equation and since many of the pollutants follow similar units. The group knew if the equations worked for 2/6 pollutants, the group would have to swap out some variables based on the pollutants and maybe some assumptions. Later, the group found out that it was a very difficult task to convert the data sets values to the respected units because of many reasons like:

- Having many assumptions, which changes by every pollutant type. This will result in inconsistency and not be very accurate.
- Calculations the group believed worked was producing very unrealistic values.
- Many misleading information on how to convert tons/year or kg/year.
- Lack of information given by the authors of the data sets information.

The group has also tried to convert the AQI into pollutant values. This requires reversing the process used to calculate the AQI from the individual pollutant concentrations. For example, calculating the moles of CO2 in a ton of CO2 gas and the annual CO2 concentration in ug/m3. This was a self-experiment to create an equation for converting AQI to pollutants, but due to insufficient and unconvincing results, this method was dropped. The problem with the results was that they were not in the desired units and the resulting values were much higher than expected. The calculation of the AQI for each pollutant depends on the method used by the environmental authority or organization responsible for monitoring air quality. The process was not straightforward as the AQI thresholds are generally not linear and not directly proportional to pollutant concentrations. In addition, the AQI is designed to prioritize health aspects, so the relationship between AQI values and actual pollutant concentrations is not always linear.

Figure 5 represent the first method of calculation the group has performed trying to convert tons/year to particle per millions (ppm). The group was able to find a blog that tried to convert ppm to Kg/day, which was like the groups situations, but all the group had to do was work backwards from the blog's solution since the group was determining kg/year to ppm.

In Figure 5 the group first converted kg/year to kg/day. The group then used the molecular weight of SO2 and assumed it was 65Kg to get rid of the weight. Then used the kilomole of SO2 to determine kmol/day then next canceled out the day and then finally Normal Cubic Meters Per Hour to get the ppm. As shown in the Figure 5 below the full calculations but at the end the group was not confident with the results and the process of the calculations. The group was not sure if $x10^6$ had to be multiplied with the calculated results or if the group can assume these values with any SO2 values. With so many questions that cannot be answered and not having realistic but also confident results the group decided not to go forward with this method.

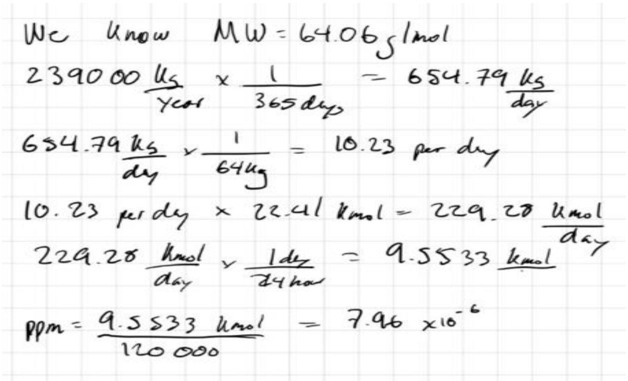


Figure 5 – Sample Calculations

After trying many calculations on converting the pollutants levels from tons/year to an acceptable unit, the group failed to have realistic result for the pollutants. The 3 units that were used were parts per million (ppm), parts per billion (ppb) and micrograms per cubic meter (ug/m^3). Even with the unrealistic conversion the group did not want to go forward with the idea so the group has decided to follow the new EU power plant guidelines that must be followed by 2030 and use the EU guidelines of pollutants that industries can release yearly. The group found that it would be a good alternative as an agricultural factory even when the group is not training g any data sets regarding the agricultural factor.

As shown from Table 1, what the pollutant values industry must follow, the percent decrease EU wants power plant to accompany by and the update pollutants that can be released.

Table 1 – Modification Data Values [3]

Pollutants	EU Guidelines	Power plant EU Guidelines (↓ %)	Modification
PM2.5 *(ug/m^3)*	0.068	56	0.02992
PM10 *(ug/m^3)*	50	56	22
O3 *(ppm)*	0.11	-	0.11
SO2 *(ppb)*	125	66	42.5
NO2 *(ppb)*	0.11	51	0.0539
CO *(ppm)*	6	-	6

With these new values, the group will combine the updated pollutant level to the predicted pollutant level the group has gotten. The updated AQI value should be greater than the predicted value but to see how badly a newly added power plant affects a city that may not have one.

C. Air Quality Index

The AQI equation is represented by Equation 4 [1], the AQI equation is actually very simple. Many of the values that will be used is based on either Figure 3 or Figure 4, depending which regulation the group will be following, the group will have to use the column of the type of pollutant they will be using and the AQI value. To know what each variable means, Table 2. Explains what each variable represents.

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{Hi} - BP_{Lo}} * \left(C_p - BP_{Lo}\right) + I_{Lo} \qquad \text{Eq. 4}$$

Table 2 – Variable Information

I_p	Air Quality Index
C_p	Pollutant level (conectration)
I_{Hi}	AQI value corresponding to BP_{Hi}
I_{Lo}	AQI value corresponding to BP_{Lo}
BP_{Hi}	Pollutant level breakpoint that is greater than or equal to Cp
BP_{Lo}	Pollutant level breakpoint that is less than or equal to Cp

Example:

$$I_{p-PM10} = \frac{50-0}{50-0} * (35-0) + 0$$

$$I_{p-PM10} = 35$$

After calculating the AQI value for the pollutant, the group must calculate all the AQI values for every pollutant level (All 6). The reason for having to do all 6 calculations is because AQI is based on the type of pollutant which results in the highest AQI value. A pollutant might seem small or not be a big impact to the quality of air based on the pollutant level but when used in the AQI equation and applied the values for the variables, it could be very different. So, for whatever pollutant a user will be using to calculate, whichever is the highest AQI rating will be represented as the AQI rating for the day. The pollutant that produced the highest AQI rating will also be the main contributor to the air pollution.

An example could be O3 = 126, PM2.5 = 105 and CO = 90, out of all these pollutants and AQI rating. PM2.5 has the highest AQI rating which means the AQI rating for the day will be 105 because of PM2.5.

Regarding the table and table, which are the guidelines for AQI rating. A quick instruction to use the table is first determining the type of pollutant and the level. Look at Figure 4 , there is a PM10 with a pollutant level of 10, when looking at the table one would go down the column of PM10 and stop at the row where pollutant level is with the range. So, in Figure 4 the group would use the range of 0-50 for PM10 which corresponds to a range of 0-50 for AQI and a category of good. Therefore, when the group puts those values into Equation 4, they will have an AQI value between 0-50.

Mentioned earlier, different countries follow different guidelines. The group was able to use the USA and Indian standard, the two differ each other by just the range of pollutant level to what the AQI rating is based on that range. As India, being a more polluted country as United States, it makes sense for India to have a larger range. They also differ with Ozone where USA does not go greater than 1 as a pollutant level and India does, which just means the two countries calculate a bit differently but still has the same outcome.

D. GUI

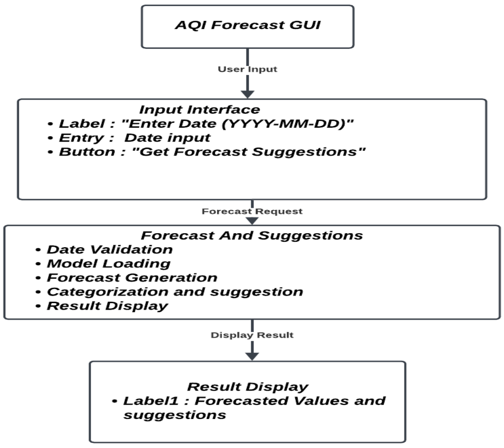


Figure 6 - Flow Chart

Facilitating user engagement with the AQI forecasting is the main goal of GUI. When a user enters a specific date (in the YYYY-MM-DD format), the GUI forecasts the levels of pollutants and suggests applicable options based on the anticipated air quality index.

i. Features

a. Input interface

The GUI includes a User-friendly interface featuring,

- A label Prompt for user to enter the date
- An entry field for data input in “YYYY-MM-DD" format.
- A button for triggering to get forecast and suggestion information

b. Forecast Display

Upon entering the desired date of the user, the following information's will display

- Forecasted values of AQI and various pollutants includes AQI, SO2, O3, CO, NO2, PM2.5, PM10
- Suggestion based on the forecasted AQI value
- Recalculated Values of potential impact AQI/pollutant Values upon installing a factory/Power plant.

c. Error Handling

- To ensure seamless user interface, the GUI incorporates error handling. It checks for invalid date formats and checks whether the input date falls within the range of extended date.

ii. Implementation details

a. Model Loading

Application loads the saved model of all AQI/pollutants model from pickle files. This ensures that the forecasted values are from previously trained and validated models.

b. Forecast Generation

From a single input date, GUI generates forecasted values from SARIMA models and appropriated suggestions are made for the AQI

c. Result Display

Enhancing user understanding, the forecasted result and suggestions are displayed in a clear and organized manner.

IV. Data Preprocessing

Data preprocessing involves Cleaning and transforming raw data into a format that is suitable for analysis and modelling. Due to the time-sensitive and dynamic nature of air quality data, our data preprocessing steps are particularly crucial. All the data preprocessing steps are done for AQI and pollutant models individually. Since the Air quality data contains data of different cities, Chennai location data is filtered out for further focused analysis. Data preprocessing steps includes,

A. Data Frame initialization

To facilitate meaningful analysis and forecasting the temporal granularity is adjusted since the raw data is in granular form with daily observations.

df['Date']= pd.to_datetime(df['Date'])

Data frame is initialized ranging from date time index 2015 January 1 – 2020 May 2. DF initialization simplifies handling of empty/missing values and ensures a consistent structure for sequent operations.

final_df = pd.DataFrame(index=np.arange('2015-01-01', '2020-05-02', dtype='datetime64[D]'), columns=column1)

Resampling kept a daily basis ‘D’.

B. Population loop

Data population loop is implemented to organize the data chronologically and align it with the datetime index. The relevant AQI/pollutant values are extracted from the dataset and assigned to the appropriated column in final_df for each city. This iterative process makes sure that the correct alignment of AQI values with the datetime index, creating a structured and organized dataset.

```
for city, i in zip (cities, final_df.columns):
 n = len(np. array(df[df['City'] == city]['AQI']))
final_df[i][-n:] = np. array(df[df['City'] == city]['AQI'])
```

C. *Data type conversion*

To optimize memory usage and maintain compatibility with analysis tool, it is critical to use the right data types. Here the datatype is converted to float 64 for accurate mathematical operations and statistical studies of AQI/Pollutant.

```
final_df = final_df. astype('float64')
```

D. *Missing data exploration*

We found that there are some missing values in the dataset during our exploration over dataset. We explored through each column to check the missing values. To solve the missing value, we need to understand the distribution of missing values and that will help us to take the appropriate imputation strategy. Since the data is a time series one, the team decided to go with backward filling strategy. This will replace the missing values with the latest available values along the specified axis.

```
final_df = final_df. bfill(axis='rows')
```

E. *Feature selection and filteration*

Since the prediction and calculation is done for India and for Chennai, features are filtered out to make a new DF for Chennai with all the features required for the model. By considering the computational power requirement during training, improved performance of the model and improved interpretability, team decided to train/predict AQI and pollutants as individual models. That reduced features in a single model.

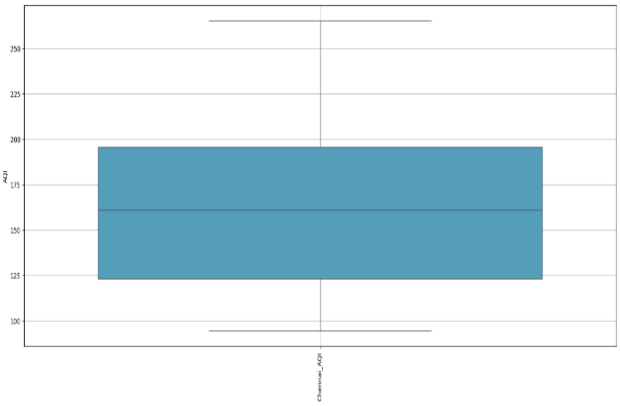


Figure 7 – Filtered Chennai AQI

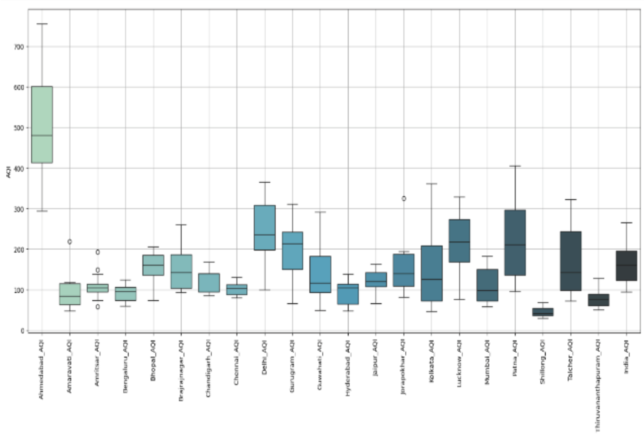


Figure 8 – AQI of Cities

V. MODEL

During data analysis, the team found that the data contains seasonality, for this reason the group decided to go with SARIMA instead of ARIMA since SARIMA has seasonality order, it will perform well with data that contains seasonality.

A. *Define Model*

For defining our model, the group needs to find the model hyper parameters and seasonality with our data. This can be done by data visualization and using library function of stat model.

```
auto_arima
(y=chennai_AQI,start_p=0,start_P=0,start_q=0,start_Q=0,seasonal=True, m=12).summary()\
```

Here, the group called our newly created data frame for location Chennai to get parameter and seasonality order.

Out[19]:

SARIMAX Results

Dep. Variable:	y	No. Observations:	65
Model:	SARIMAX(0, 1, 0)x(1, 0, [1], 12)	Log Likelihood	-317.367
Date:	Mon, 15 Jan 2024	AIC	640.734
Time:	14:02:15	BIC	647.211
Sample:	01-01-2015	HQIC	643.286
	- 05-01-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.S.L12	0.9665	0.066	14.574	0.000	0.837	1.097
ma.S.L12	-0.7031	0.279	-2.524	0.012	-1.249	-0.157
sigma2	964.6870	206.343	4.675	0.000	560.262	1369.112

Ljung-Box (L1) (Q):	0.59	Jarque-Bera (JB):	17.23
Prob(Q):	0.44	Prob(JB):	0.00
Heteroskedasticity (H):	0.23	Skew:	-0.72
Prob(H) (two-sided):	0.00	Kurtosis:	5.09

Figure 9 – SARIMAX Results

From the summary, the group will get the order and seasonality parameters. (0,1,0) seasonality order (1,0,1,12). Even though these parameters values may not give the Best RMSE/MSE values, the parameters are again hyper tuned later by undergoing visualization.

B. Fit Model

Once the model parameters are defined, the group needs to fit our training data to our model. The group can use model.fit () function. For splitting the index into training and testing, the len of index is calculated using len (chennai_AQI) and found that len size is 65. For len size 65, the maximum range to split the training and testing data is 54. So, it is kept split 41 for training and 12 observations for testing the data.

train = chennai_AQI [:41]
test = chennai_AQI [42:54]

Model is formed using,

model=SARIMAX (train,order=(1,1,1),seasonal_order=(1,0,1,12),)
results=model.fit ()
results. summary()

The predicted results are plotted against the actual values from summary ().

C. Make Predictions

To make predictions from the results, the group can use *predictions = results. predict(start=64, end=200, typ='levels').rename('Predictions')* . for prediction, the group needs to give a start and end point. Since our data frame size is 65, the group can start the prediction from 64 up to a desired end point. The sampling rate will depend upon the frequency the group selects. For visualizing it for a longer period till 2030. For visualization purpose the group sampled to Month start (MS), that will show the prediction values of every start of the month. The group can plot the prediction from predictions ().

D. Model evaluation

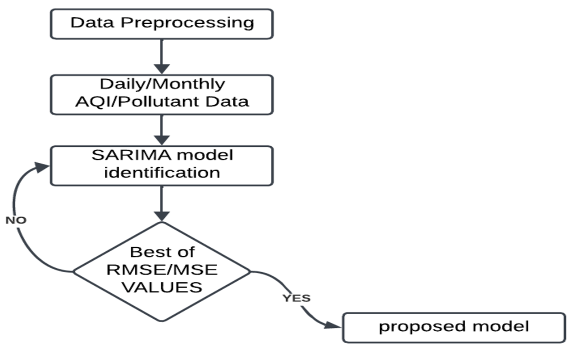


Figure 10 - Model Evaluation Diagram

The model is evaluated based on RMSE values, RMSE = $\sqrt{\sum_{i=1}^{n}(Xobs,i - Xmodel,i)} \wedge 2$. Here, Xobs, i is an observed value whereas Xmodel,i is known as modelled value at the time i.

E. Forecasting

To forecast, the group saved the model using pickle in pkl format. The saved model is then imported using pickle library. With a desired date for forecast or a series of forecast can be done. Since all the Model of AQI, NO2, SO2, CO, O3, PM2.5, PM10 are trained in same Date index, data frame initialization and frequency, the group can call all the model at a time to forecast by a single input.

VI. RESULT

A. Prediction Tool

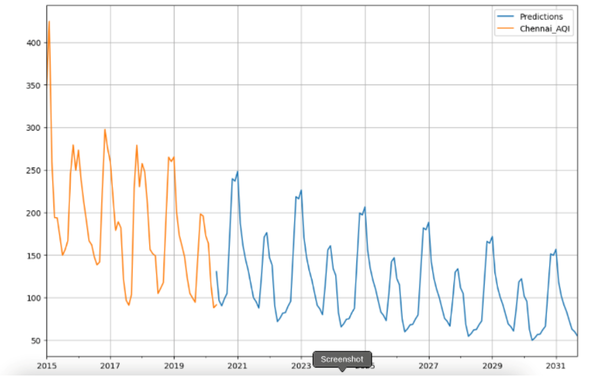


Figure 11 - Chennai AQI Prediction

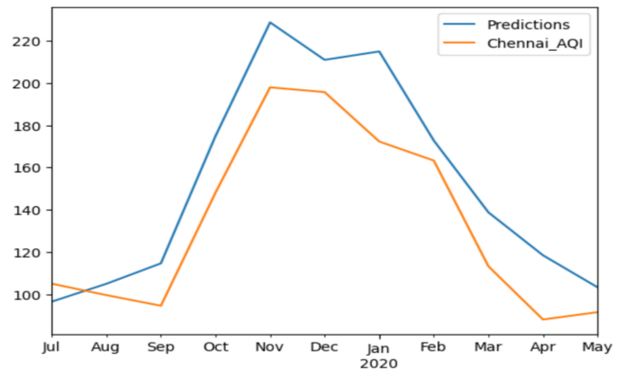


Figure 12 - Chennai AQI Prediction vs. Actual

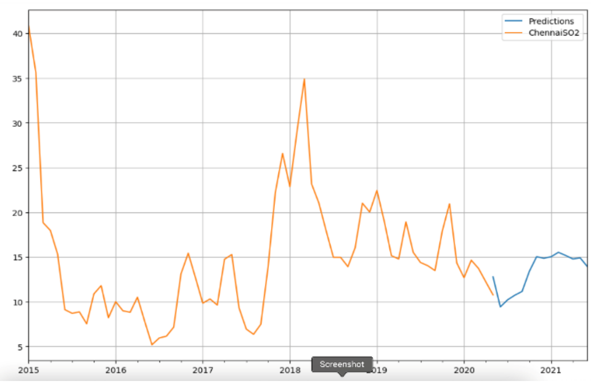


Figure 13 - Chennai SO2 Prediction

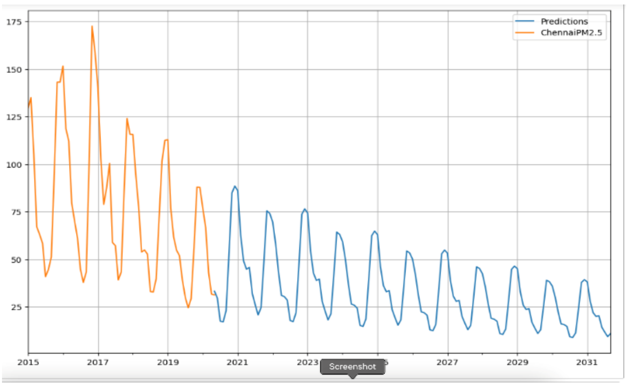


Figure 14 – Chennai CO Prediction

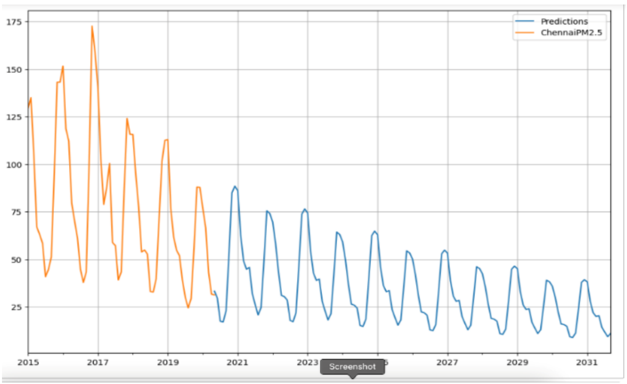


Figure 15 - Chennai PM2.5 Prediction

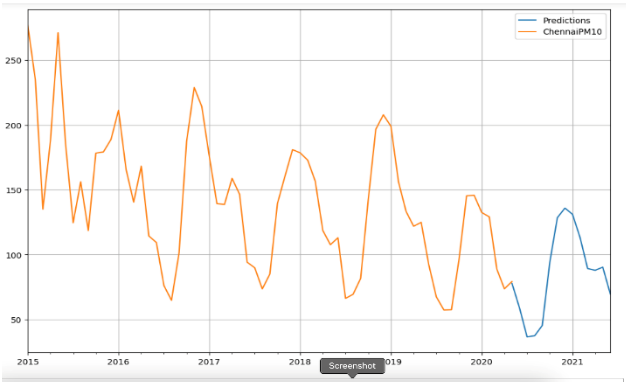


Figure 16 - Chennai PM10 Prediction

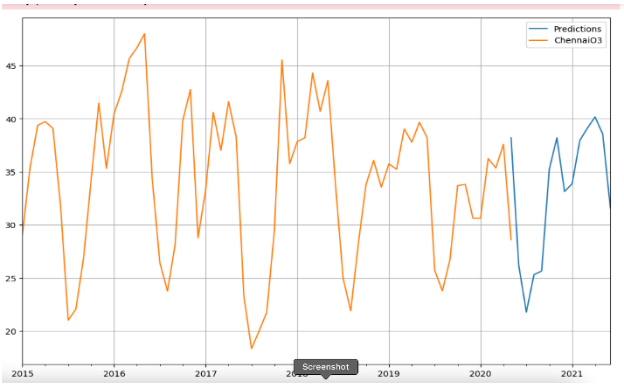


Figure 17 – Chennai O3 Prediction

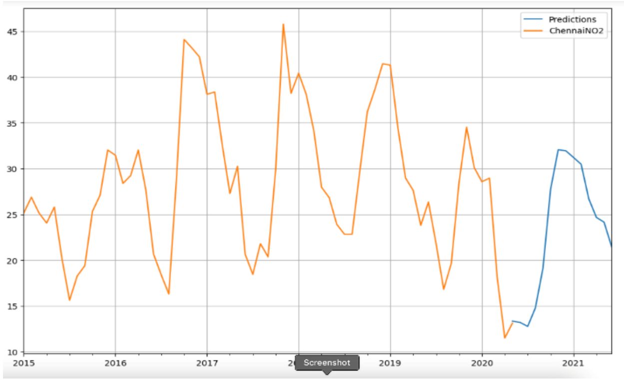


Figure 18 – Chennai NO2 Prediction

Table 3 - RMSE of Trained model.

MODEL	RMSE
AQI	20.06
CO	1.77
NO2	5.15
SO2	5.15
PM10	19.02
PM2.5	12.98
O3	3.37

The group make SARIMA models to predict various air quality parameters of the location Chennai includes AQI, PM2.5, PM10, O3, SO2, NO2, and CO the help of Air quality index data which collected from government of India official website .The models are baled to predict accurate values by capturing the trend and seasonality patterns of the data. To optimize/tune the models for precise predictions, a rigorous selection of model hyperparameters, such as order and seasonal order, was made during the training phase. Our key finding on AQI forecasting is, with the levels of air quality, the SARIMA model for the AQI really showed its efficiency in forecasting the Air Quality Index. The model highlighted its capacity to classify air quality conditions and supports public awareness and decision-making. The individual trained models of pollutants by using SARIMA namely PM2.5, PM10, SO2, NO2, CO, O3 gave a comprehensive understanding of air quality dynamics. The evaluation metrics such as RMSE gave insights about the performance of the model and predictive capabilities.

B. Agricutural Factor

Calculating the new pollutant levels based on the power plant involves the use of Table 1 and just simply adding the predicted value with the medication value. Users would then calculate each of the pollutant types: Particle matter 2.5 (PM2.5), Particle Matter 10(PM10), Ozone 3(O3), Carbon Monoxide (CO), Sulfur Dioxide (SO2), and Nitrogen Dioxide (NO2), which will then determine which values from Figure 4 to determine the pollutant level.

As shown in the Table 4 the new values of pollutant levels based on a power plant being added.

Table 4 – Pollutant Level Calculations Value

Pollutant	Predicted value	Equation	Update Pollutant level
PM2.5 *(ug/m³)*	98.725	98.725 + 0.02992	98.755
PM10 *(ug/m³)*	44.794	44.794 + 22	66.794
O3 *(ppm)*	21.361	21.361 + 0.11	21.471
SO2 *(ppb)*	37.575	37.575 + 42.5	80.075
NO2 *(ppb)*	33.416	33.416 + 0.0539	33.470
CO *(ppm)*	1.9256	1.9256 + 6	7.9256

Table 4 shows the new pollutant value, and one can see the pollutant levels have increased, which is what the group wanted. All the agricultural factor values are pollutant levels that the maximum number of pollutants can be released, for sure lower number of values can be produced by power plants but the group is using the concept of the power plant going full power. CO, SO2 and PM10 are the pollutants that make a big impact to the pollutants which means that power plants does not produce that much type of pollutants therefore it is not reduced by much and like what mention before some pollutants might not ever produce that much pollutant in a day.

C. Air Quality Index Calculation

When using Equation 4, the group was able to determine the AQI value of any pollutant. The only tools the group needed were Equation 4 and either Figure 3 or Figure 4 depending on the regulation the group wanted to use.

Below shows an example calculation of the AQI rating based on different pollutants.

Example:
Here shows, using Equation 4 and Figure 4 to determine the AQI rating for PM2.5.

$$I_{p-PM2.5} = \frac{60-31}{150-101} * (47.04 - 31) + 101 \qquad \text{Ex. 2}$$

Then here shows the pollutant levels and the AQI rating of each pollutant with Equation 4.

Table 5 – AQI Calculation Value

Pollutant	Pollutant level	AQI Rating
PM2.5 *(ug/m³)*	47.04	78.10
PM10 *(ug/m³)*	44.763	44.763.794
O3 *(ppm)*	21.465	21.465
SO2 *(ppb)*	73.996	91.430
NO2 *(ppb)*	30.612	38.265
CO *(ppm)*	7.2759	165.8625

From the Table 5, one can see that PM2.5 is the highest AQI value which means 222 is the AQI rating with PM2.5 being the contributor.

D. GUI

After careful consideration and to reduce confusion while displaying the results, two GUIs were developed for the project, as shown below:

1. Pollutant Forecasting and Suggestions GUI: It is divided into two parts. One is for predicting the AQI and the 6 pollutants, the other is just for determining the modified 6 pollutant values according to the equations mentioned. Here the group simply entered the desired date in the format YYYY-MM-DD.

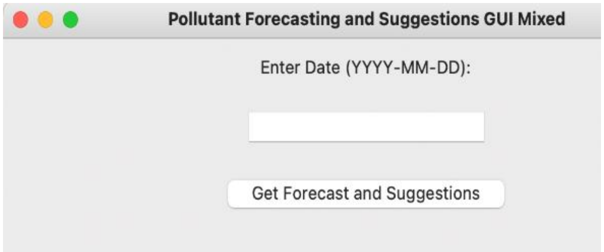


Figure 19 – GUI 1 User Input Window

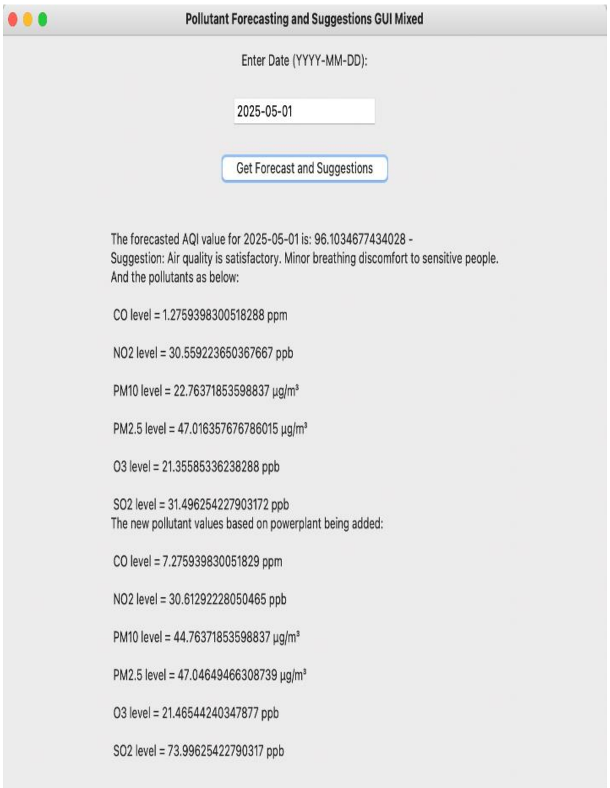


Figure 20 - GUI 1 Result Window

The above results consist of two sections for the date 2025-05-01. One contains the predictions for the AQI ~ 96.10 and the six pollutants CO, NO2, PM10, PM2.5, O3 & SO2 with the units, as shown in the Figure 20. The other section contains the modified pollutant values due to the establishment of a power plant in the area. These new modified pollutant values will be the inputs for the second GUI, which calculates the new AQI value due to the power plant.

AQI Calculator GUI: It is used to calculate the modified AQI value for the same date, but with the values of the modified pollutants from the first GUI. Here the group entered the resultant altered values of the 6 pollutants SO2, NO, CO, O3, PM2.5, PM10 from the Figure 20. And the group got the output, the AQI for the day according to the altered pollutants (highest pollutant affecting the AQI).

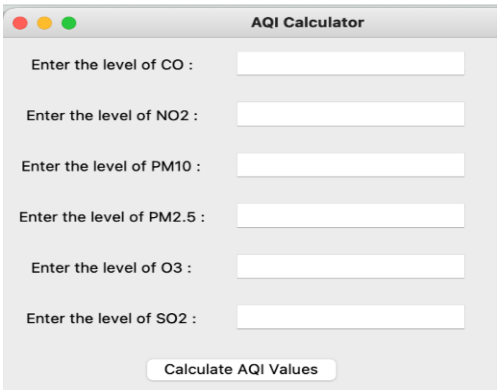


Figure 21 - GUI 2 User Input Window

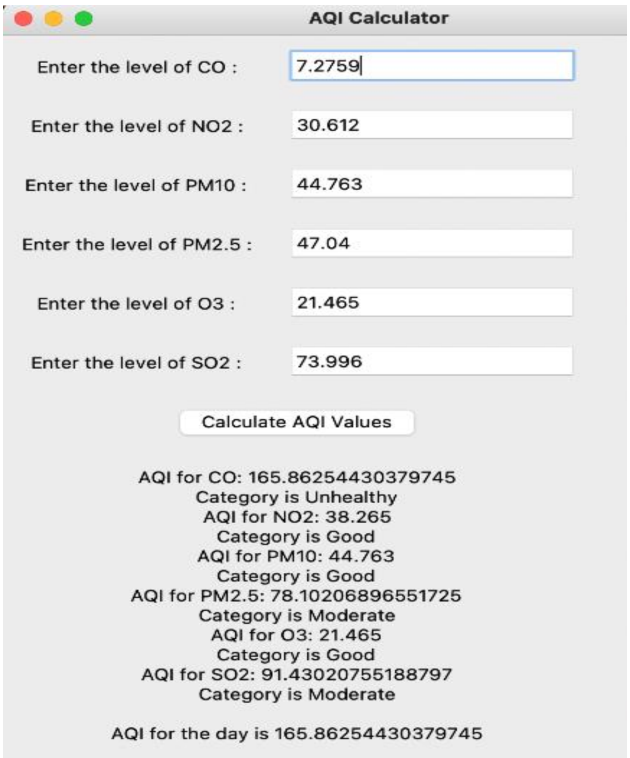


Figure 22 - GUI 2 Result Window

The AQI contribution for each pollutant is displayed in the above result, with the pollutant having the largest contribution, typically PM2.5 or PM10, indicating the AQI for the current day. The group can now compare the two AQIs from the two GUIs because the final line of the result shows the changed AQI in the figure. Here, the two values 96.10 and 165.86 clearly show how the local power plant arrangement has affected the AQI.

VII. Conclusion

Throughout the process the group encountered some blockages, which made the group change direction with some requirements. Mainly with the agricultural factor, converting the pollutant values from kg/year to either ppm, ppb, ug/m^3 was difficult to do since the group was determining unrealistic values. Converting the pollutant values to those specific units was very important because those were the units needed to acquire the AQI rating. With the quick pivot of direction, the group was still able to apply an agricultural factor tool with the predicted values based on the European power plant regulations and manipulated the concept that will allow a more realistic value. At the end, it was not the original plan and it felt simpler with just modify the values by calculation, but the group was happy to still apply some factor and can still be used as a tool to know if adding power plants with new regulations will help cities. Especially when users can also compare the different regulations from other continents.

The group using the model of SARIMAX allows the data sets to be properly trained and processed while giving the results the group is looking for. SARIMAX was a type of model that would focus on seasonal patterns which helps the group modify parameters to obtain an accurate prediction. With comparing other type of models, the group concluded with many giving the same results but not as precises as SARIMAX, the group has decided not to make the case study more complicated by adding multiple models but just focusing the groups time with SARIMAX.

The group has created a graphical user interface (GUI) that allows to display the predicted AQI and polluted values, modified polluted values and the newly calculated AQI ratings based on the modified pollutant levels caused by the power plant. Within the groups GUI, the group created 2 GUIs that allow one window to predict all the values and the other window to calculate the new AQI values. Based on the window that displays the predicted values, users are required to input data (Year-month-day) they desire to predict. The group was able to store the models of prediction of AQI and pollutant levels as a pkl file which will easily read the files without making the groups code pages long. plus adding simple equations that will use the results of the predicted pollutant levels and modify the values. Regarding the calculation GUI, it is not as complicated as the first window since the group had to input equations to calculate the values while the group had to manually input the desired pollutant levels the group would like to calculate.

Overall, the aim of the case study was to create a predictions tool with user alerts based on AQI ratings, apply an agricultural factor to the predicated value and apply a user-friendly interaction. By all counts and with the proven results the group was able to achieve their goals by using specific data sets, in a specific city to be able to predict a AQI rating for future dates and compare those values with new values that has an addition of agricultural factors and allow any users use the tool. With the agricultural factor issues the group has encounter, the group was still able to successfully complete the case study.

VIII. Outlook

Throughout the case study and after the research, the datasets were the main concern and will always be for this project as they play the key role. The success of a project is not solely determined by the size of the dataset but also by the quality of the data and the methods employed. The efficiency of the predictions depends on the accuracy and availability of sufficient datasets. There will always remain room for improvement in this section as it was the main source of predictions.

The model chosen was SARIMAX, but with more datasets the window can be used for other models such as LSTM or Facebook Prophet models for more accurate predictions. Evaluating and validating any alternative models to choose since they could provide the group with better results compared to the model currently in use, as their data processing capacity is greater, and they have more adjustable parameters.

More accurate data sets and the equations for agriculture and power plant construction, which influence AQI and pollutant levels, would also have been much more helpful. They play a significant role in predicting the construction of power plants or agricultural use. Modern technology upgrades also lead to changes in the equations, so a proper update of the equations is also necessary for the resulting AQI to be more exact. When implementing strategies to modify pollutants, it is important to consider the specific characteristics of the power plant, the type of pollutants emitted and the regulatory environment. In addition, the involvement of environmental experts and cooperation with regulatory authorities can help to ensure effective and sustainable pollution control measures.

The use of two GUIs can be transformed into one well-organized and efficient GUI. This makes it easier to use and takes less time and space to display the results in a single window. This is beneficial in various scenarios and provides a consistent and optimized user experience. While a single GUI has its advantages, the decision should be based on the specific needs and requirements of application and user base. Careful consideration of workflows, functionality and design principles will help determine whether a single or multiple GUIs are better suited to any project.

Finally, with the current data set, it was sufficient to obtain the predictions on a small scale. But to extend them to a large scale, it would be much easier to have a universal dataset and universal AQI standard assessments for all countries, reducing the effort and increasing the efficiency for use in any country. While universal datasets offer various advantages, it is important to consider potential limitations such as bias, data quality and the need for domain-specific nuances. In addition, ensuring proper documentation and ethical considerations are crucial to utilize the benefits of universal dataset.

IX. Acknowledgement

The group would like to express their deepest appreciation to professor Ginu Paul Alinkal who provided the group the opportunity to successfully complete the case study and provided guidance to the group. A thank you to all the group members that put their time and effort into completing every task that was assigned, worked non-stop on getting the correct result and the suggestions and encouragement each member had with each other. Furthermore, the group would also like to thank IEE Organization, World Health Organization, Environmental European Commission, Environmental European Agency, Air Now, Prana Air and all the other external recourse the group has used to gather information, data and guidance to complete the case study.

References

[1] " Air Now," [Online]. Available: https://www.airnow.gov/sites/default/files/2020-05/aqi-technical-assistance-document-sept2018.pdf. [Accessed 2024].

[2] [Online]. Available: https://www.pranaair.com/blog/what-is-air-quality-index-aqi-and-its-calculation/.

[3] "Enviroment European Commission," [Online]. Available: https://environment.ec.europa.eu/topics/air/air-quality/eu-air-quality-standards_en. [Accessed 2024].

[4] "World Health Organization," [Online]. Available: https://www.who.int/news/item/03-09-2021-who-and-un-partners-compendium-of-500-actions-aims-to-reduce-diseases-from-environmental-factors-and-save-lives. [Accessed 2024].

[5] "Righ to Clean Air," [Online]. Available: https://www.right-to-clean-air.eu/en/background/deadlines-for-compliance-with-the-limit-values/#:~:text=For%20even%20smaller%20PM2.,of%2020%20%C2%B5g%2Fm3.. [Accessed 2024].

[6] "CO2Meter," [Online]. Available: https://www.co2meter.com/en-de/blogs/news/carbon-monoxide-levels-chart. [Accessed 2024].

[7] "Real Python," [Online]. Available: https://realpython.com/account/signup/?intent=continue_reading&utm_source=rp&utm_medium=web&utm_campaign=rwn&utm_content=v1&next=%2Fpython-gui-tkinter%2F. [Accessed 2024].

[8] "Control Automation," [Online]. Available: https://control.com/forums/threads/ppm-to-kg-day-conversion.31785/. [Accessed 2024].

[9] "European Enviroment Agency," [Online]. Available: https://sdi.eea.europa.eu/data/3da7d329-beea-4a7b-89bc-d45fc1c4b8ac?path=%2FEXCEL. [Accessed 2024].

[10] "Teesing," [Online]. Available: https://teesing.com/en/tools/ppm-mg3-converter#mgm3-ppm. [Accessed 2024].

[11] B. K. B. K. K.M.O.V.K. Kekulanadara, "Machine Learning Approach for Predicting Air," Sri Lanka, 2021.

[12] S. G. M. V. S. L. M. K. N. A. S. M. Parameshachari B D, "Prediction and Analysis of Air Quality Index using," 2022.

[13] G. L. B. S. Karlapudi Saikiran, "Prediction of Air Quality Index Using Supervised," India, 2021.

[14] Vopani, "Kaggle," [Online]. Available: https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india. [Accessed 2024].

APPENDIX

A. *Pollutant Forecasting and Suggestions GUI Code:*

```
import pickle
import pandas as pd
from statsmodels.tsa.statespace.sarimax import
SARIMAX
import tkinter as tk
from tkinter import ttk

class AQIForecastGUI:
    def __init__(self, root):
        self.root = root
        self.root.title("Pollutant Forecasting and
Suggestions GUI Mixed")

        # Defining pollutant names, model
filenames, and units
        pollutants = ['AQI', 'CO', 'NO2', 'PM10',
'PM2.5', 'O3','SO2']
        model_filenames = [
            'Nishanthsarima4_model.pkl',
            'Nishanthsarima_model_CO.pkl',
            'Nishanthsarima_model_NO2.pkl',
            'Nishanthsarima_model_pm10.pkl',
            'Nishanthsarima_model1_pm2_5.pkl',
            'Nishanthsarima_model_SO2.pkl',
            'Nishanthsarima_model_O3.pkl'
        ]
        units = ['-', 'ppm', 'ppb', 'µg/m³', 'µg/m³',
'ppb', 'ppb']

        # Creating a dictionary to store loaded
models and units
        self.loaded_models = {pollutant: {'model':
None, 'unit': unit, 'filename': filename} for
pollutant, filename, unit in zip(pollutants,
model_filenames, units)}

        # Loading of saved SARIMA models for
each pollutant
        for pollutant, model_info in
self.loaded_models.items():
            with open(model_info['filename'], 'rb') as
file:
                model_info['model'] = pickle.load(file)

        # Extension of time index to include the
forecast period
        self.extended_index =
pd.date_range(start='2015-01-01', end='2025-12-
31', freq='D')

        # GUI components
        self.label_date = ttk.Label(root, text="Enter
Date (YYYY-MM-DD):")
        self.label_date.pack(pady=10)

        self.date_entry = ttk.Entry(root)
        self.date_entry.pack(pady=10)

        self.forecast_button = ttk.Button(root,
text="Get Forecast and Suggestions",
command=self.get_forecast_and_suggestions)
        self.forecast_button.pack(pady=10)

        self.result_label = ttk.Label(root, text="")
        self.result_label.pack(pady=10)

    def get_forecast_and_suggestions(self):
        desired_date = self.date_entry.get()

        try:
            pd.to_datetime(desired_date)
        except ValueError:
            self.result_label.config(text="Invalid date
format. Please use YYYY-MM-DD.")
            return

        if pd.to_datetime(desired_date) not in
self.extended_index:

self.result_label.config(text="'desired_date'
should be within the range of the extended
index.")
            return

        # Initialize result text
        result_text = ""

        # Make predictions for each pollutant using
the loaded models
        for pollutant, model_info in
self.loaded_models.items():
```

```
            forecast =
model_info['model'].get_prediction(start=desired
_date, end=desired_date, dynamic=False)
            predicted_value =
forecast.predicted_mean.iloc[0]

            # Categorization and suggestions for each
pollutant
            if pollutant == 'AQI':
                res = int(predicted_value)
                if res <= 50:
                    suggestion = "Enjoy the fresh air!
Air quality is good with minimal impact."
                elif 51 <= res <= 100:
                    suggestion = "Air quality is
satisfactory. Minor breathing discomfort to
sensitive people."
                elif 101 <= res <= 200:
                    suggestion = "Air quality is
moderate. Breathing discomfort to people with
lungs, asthma, and heart diseases."
                elif 201 <= res <= 400:
                    suggestion = "Air quality is very
poor. Breathing discomfort to most people on
prolonged exposure."
                elif 401 <= res <= 500:
                    suggestion = "Air quality is severe.
Affects healthy people and seriously impacts
those with existing diseases."

                result_text += f"\nThe forecasted
{pollutant} value for {desired_date} is:
{predicted_value} {model_info['unit']}\n"
                result_text += f"Suggestion:
{suggestion}\n"
                result_text += f"And the pollutants as
below: \n"
            else:
                result_text += f"\n {pollutant} level =
{predicted_value} {model_info['unit']}\n"

        # Categorization and suggestions for each
pollutant
        for pollutant, model_info in
self.loaded_models.items():
            forecast =
model_info['model'].get_prediction(start=desired
_date, end=desired_date, dynamic=False)
            predicted_value =
forecast.predicted_mean.iloc[0]

            # Categorization and suggestions for each
pollutant
            if pollutant == 'AQI':
                #modifying the predicted value
                #predicted_value = (predicted_value +
2) / 2
                res = int(predicted_value)
                if res <= 50:
                    suggestion = "Enjoy the fresh air!
Air quality is good with minimal impact."
                elif 51 <= res <= 100:
                    suggestion = "Air quality is
satisfactory. Minor breathing discomfort to
sensitive people."
                elif 101 <= res <= 200:
                    suggestion = "Air quality is
moderate. Breathing discomfort to people with
lungs, asthma, and heart diseases."
                elif 201 <= res <= 400:
                    suggestion = "Air quality is very
poor. Breathing discomfort to most people on
prolonged exposure."
                elif 401 <= res <= 500:
                    suggestion = "Air quality is severe.
Affects healthy people and seriously impacts
those with existing diseases."

                result_text += f"The new pollutant
values based on powerplant being added: \n"
                #result_text += f"\n NEW {pollutant} :
{predicted_value} {model_info['unit']}\n"
                #result_text += f"Suggestion:
{suggestion}\n"
                #result_text += f"tThe new pollutant
values based on powerplant being added: \n"
            elif pollutant == 'CO':
                #modifying the predicted value
                predicted_value = (predicted_value +
6)
                result_text += f"\n {pollutant} level =
{predicted_value} {model_info['unit']}\n"
            elif pollutant == 'NO2':
                #modifying the predicted value
                predicted_value = (predicted_value +
((40/365) * (1-0.51)))
                result_text += f"\n {pollutant} level =
{predicted_value} {model_info['unit']}\n"
            elif pollutant == 'PM10':
                #modifying the predicted value
                predicted_value = (predicted_value +
((50) * (1-0.56)))
                result_text += f"\n {pollutant} level =
{predicted_value} {model_info['unit']}\n"
            elif pollutant == 'PM2.5':
                #modifying the predicted value
                predicted_value = (predicted_value +
(25/365) * (1-0.56))
                result_text += f"\n {pollutant} level =
{predicted_value} {model_info['unit']}\n"
            elif pollutant == 'O3':
                #modifying the predicted value
                predicted_value = (predicted_value +
(120/(3*365)))
                result_text += f"\n {pollutant} level =
{predicted_value} {model_info['unit']}\n"
            elif pollutant == 'SO2':
                #modifying the predicted value
                predicted_value = (predicted_value +
(125 * (1-0.66)))
                result_text += f"\n {pollutant} level =
{predicted_value} {model_info['unit']}\n"
```

```
        # Displaying the forecasted values and
suggestions for the specified date
        self.result_label.config(text=result_text)

if __name__ == "__main__":
    root = tk.Tk()
    app = AQIForecastGUI(root)
    root.mainloop()
```

B. AQI Calculator GUI code:

```
import tkinter as tk

def calculate_expression (I_HI, I_LO, BP_HI,
BP_LO, C_p):
    result = ((((I_HI-I_LO) / (BP_HI -
BP_LO))*(C_p-BP_LO))+I_LO)
    return result

def pollutant_values(C_p, category):
    if category == "PM2.5":
        if 0 <= C_p <= 30:
            return 50, 0, 30, 0
        elif 31 <= C_p <= 60:
            return 100, 51, 60, 31
        elif 61 <= C_p <= 90:
            return 200, 101, 90, 62
        elif 91 <= C_p <= 120:
            return 300, 201, 120.0, 91
        elif 121 <= C_p <= 250:
            return 400, 301, 250, 121
        elif C_p <= 251:
            return 500, 401, 251, C_p

    elif category == "PM10":
        if 0 <= C_p <= 50:
            return 50, 0, 50, 0
        elif 51 <= C_p <= 100:
            return 100, 51, 100, 51
        elif 101 <= C_p <= 250:
            return 200, 101, 250, 101
        elif 251 <= C_p <= 350:
            return 300, 201, 350, 251
        elif 351 <= C_p <= 430:
            return 400, 301, 430, 351
        elif C_p <= 431:
            return 500, 401, 431, C_p

    elif category == "O3":
        if 0 <= C_p <= 50:
            return 50, 0, 50, 0
        elif 51 <= C_p <= 100:
            return 100, 51, 100, 51
        elif 101<= C_p <= 168:
            return 200, 101, 168, 101
        elif 169 <= C_p <= 208:
            return 400, 301, 208, 169
        elif 209 <= C_p <= 748:
            return 748, 209, 0.200, 0.106
        elif C_p <= 749:
            return 500, 401, 749, C_p

    elif category == "CO":
        if 0 <= C_p <= 1.0:
            return 50, 0, 1.0, 0
        elif 1.1 <= C_p <= 2.0:
            return 100, 51, 2.0, 1.1
        elif 2.1 <= C_p <= 10:
            return 200, 101, 10, 2.1
        elif 10.1 <= C_p <= 17:
            return 300, 101, 17, 10.1
        elif 17.1 <= C_p <= 34:
            return 400, 301, 34, 17.1
        elif C_p <= 34.1:
            return 500, 401, 34.1, C_p

    elif category == "SO2":
        if 0 <= C_p <= 40:
            return 50, 0, 40, 0
        elif 41 <= C_p <= 80.99:
            return 100, 51, 80.99, 41
        elif 81 <= C_p <= 380:
            return 200, 101, 380, 81
        elif 381 <= C_p <= 800:
            return 300, 201, 800, 381
        elif 801 <= C_p <= 1600:
            return 400, 301, 1600, 801
        elif C_p <= 1601:
            return 500, 401, 1601, C_p

    elif category == "NO2":
        if 0 <= C_p <= 40:
            return 50, 0, 40, 0
        elif 41 <= C_p <= 80:
            return 100, 51, 80, 41
        elif 81 <= C_p <= 180:
            return 200, 101, 180, 81
        elif 181 <= C_p <= 280:
            return 300, 201, 280, 181
        elif 281 <= C_p <= 400:
            return 400, 301, 400,281
        elif C_p <= 401:
            return 500, 401, 1649, C_p

        return None
def calculate_aqi(category_entry, cp_entries,
result_label):
    result_text = ""
    max_AQI = 0

    for i, entry in enumerate(cp_entries):
        category = category_entry[i].get().upper()
        C1_p = entry.get()

        if not C1_p:
            result_text += f"Enter a level for
{category}\n"
            continue

        try:
            C_p = float(C1_p)
        except ValueError:
            result_text += f"Invalid level for
{category}: {C1_p}\n"
```

```
                continue

            input_values = pollutant_values(C_p, category)

            if input_values is not None:
                I_HI, I_LO, BP_HI, BP_LO = input_values

                result = calculate_expression(I_HI, I_LO, BP_HI, BP_LO, C_p)

                if result > max_AQI:
                    max_AQI = result

                if 0 <= result <= 50:
                    comment = "Good"
                elif 51 <= result <= 100:
                    comment = "Moderate"
                elif 101 <= result <= 150:
                    comment = "Unhealthy for sensitive groups"
                elif 151 <= result <= 200:
                    comment = "Unhealthy
                elif 201 <= result <= 300:
                    comment = "Very unhealthy"
                else:
                    comment = "Hazardous"

                result_text += f"AQI for {category}: {result}\nCategory is {comment}\n"
            else:
                result_text += "Unknown Pollutant\n"

    result_text += f"\nAQI for the day is {max_AQI}\n"
    result_label.config(text=result_text)

def run():
        window.mainloop()

#AQIcalculatorGUI:
if __name__ == '__main__':
    window = tk.Tk()
    window.title("AQI Calculator")
    window.geometry ('500x700')

    pollutants = ["CO", "NO2","PM10", "PM2.5", "O3", "SO2"]

    category_entries = []
    cp_entries = []

    for i, pollutant in enumerate(pollutants):
        label = tk.Label(window, text=f"Enter the level of {pollutant} :")
        label.grid(row=i, column=0, padx=10, pady=10)

        textentry_value = tk.Entry(window)
        textentry_value.grid(row=i, column=1, padx=10, pady=10)
        category_var = tk.StringVar(value=pollutant)
        category_entries.append(category_var)
        cp_entries.append(textentry_value)

    button = tk.Button (window, text = f"Calculate AQI Values", command = lambda: calculate_aqi(category_entries, cp_entries, result_label))
    button.grid(row =len(pollutants),column = 0, columnspan =2, padx = 10, pady=10)

    result_label = tk.Label(window, text = "")
    result_label.grid(row = len(pollutants) +1 ,column = 0, columnspan =2,padx = 10, pady = 10)

    run()
```

C. Calcualtions

1 metric ton CO_2 gas = Sphere with diameter 10.07 m

$$\text{Volume of Sphere} = \frac{4}{3}\pi r^3$$

$$\text{Vol. of sphere} = \frac{4}{3}\pi\left(\frac{10.07}{2}\right)^3 m^3$$

$$= \frac{4}{3}\pi(5.035)^3$$

$$= 534.67\ m^3$$

1 Ton of CO_2 = 1000kg

Molecular Weight CO_2 = 44.01 g/mol

$$\text{Moles of } CO_2 = \frac{\text{mass of } CO_2}{\text{molecular wt. of } CO_2}$$

$$= \frac{1000\ kg}{44.01\ g/mol} \times 1000$$

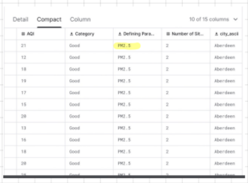
$$= 22722.10\ mol$$

$$ppm = \frac{\text{moles of } CO_2}{\text{moles of air}} \times 10^6$$

$$= \frac{22722.10}{1} \times 10^6$$

$$= 22.72 \times 10^9\ mol$$

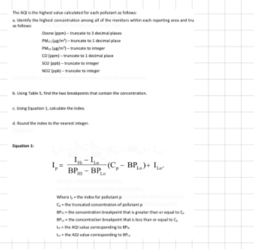
$$CO_2\ Conc = \frac{\text{Annual } CO_2 \text{ emission in } \mu g}{\text{Vol. of Air in } m^3}$$



# AQI	Category	Defining Para...	# Number of Sit...	city_ascii
21	Good	PM2.5	2	Aberdeen
12	Good	PM2.5	2	Aberdeen
18	Good	PM2.5	2	Aberdeen
19	Good	PM2.5	2	Aberdeen
17	Good	PM2.5	2	Aberdeen
15	Good	PM2.5	2	Aberdeen
20	Good	PM2.5	2	Aberdeen
13	Good	PM2.5	2	Aberdeen
16	Good	PM2.5	2	Aberdeen
18	Good	PM2.5	2	Aberdeen
20	Good	PM2.5	2	Aberdeen

PM2.5 = 21 AQI

What is 21 AQI to PM2.5 concentration?



The AQI is the highest value calculated for each pollutant as follows:

a. Identify the highest concentration among all of the monitors within each reporting area and truncate as follows:

Ozone (ppm) – truncate to 3 decimal places
$PM_{2.5}$ (μg/m³) – truncate to 1 decimal place
PM_{10} (μg/m³) – truncate to integer
CO (ppm) – truncate to 1 decimal place
SO2 (ppb) – truncate to integer
NO2 (ppb) – truncate to integer

b. Using Table 5, find the two breakpoints that contain the concentration.

c. Using Equation 1, calculate the index.

d. Round the index to the nearest integer.

Equation 1:

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

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}

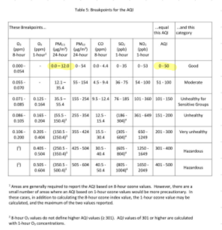


Table 5: Breakpoints for the AQI

These Breakpoints...							...equal this AQI	...and this category
O_3 (ppm) 8-hour	O_3 (ppm) 1-hour[1]	$PM_{2.5}$ (μg/m³) 24-hour	PM_{10} (μg/m³) 24-hour	CO (ppm) 8-hour	SO_2 (ppb) 1-hour	NO_2 (ppb) 1-hour	AQI	
0.000 - 0.054	-	0.0 – 12.0	0 - 54	0.0 - 4.4	0 - 35	0 - 53	0 - 50	Good
0.055 - 0.070	-	12.1 – 35.4	55 - 154	4.5 - 9.4	36 - 75	54 - 100	51 - 100	Moderate
0.071 - 0.085	0.125 - 0.164	35.5 – 55.4	155 - 254	9.5 - 12.4	76 - 185	101 - 360	101 - 150	Unhealthy for Sensitive Groups
0.086 - 0.105	0.165 - 0.204	(55.5 - 150.4)[3]	255 - 354	12.5 - 15.4	(186 - 304)[4]	361 - 649	151 - 200	Unhealthy
0.106 - 0.200	0.205 - 0.404	(150.5 - 250.4)[3]	355 - 424	15.5 - 30.4	(305 - 604)[4]	650 - 1249	201 - 300	Very unhealthy
(2)	0.405 - 0.504	(250.5 - 350.4)[3]	425 - 504	30.5 - 40.4	(605 - 804)[4]	1250 - 1649	301 - 400	Hazardous
(2)	0.505 - 0.604	(350.5 - 500.4)[3]	505 - 604	40.5 - 50.4	(805 - 1004)[4]	1650 - 2049	401 - 500	Hazardous

[1] Areas are generally required to report the AQI based on 8-hour ozone values. However, there are a small number of areas where an AQI based on 1-hour ozone values would be more precautionary. In these cases, in addition to calculating the 8-hour ozone index value, the 1-hour ozone value may be calculated, and the maximum of the two values reported.

[2] 8-hour O_3 values do not define higher AQI values (≥ 301). AQI values of 301 or higher are calculated with 1-hour O_3 concentrations.

- use eq 1 to determine C_p since we know I_p

$$C_p = \left((I_p - I_{Lo}) \div \frac{I_{hi} - I_{Lo}}{BP_{HI} - BP_{Lo}}\right) + BP_{Lo}$$

$$C_{PM2.5} = \left((21 - 0) \div \left(\frac{50 - 0}{12 - 0}\right)\right) + 0$$

CO = 1 150 000   NO = 30 500 000   SO2 = 239000   (kg/year)

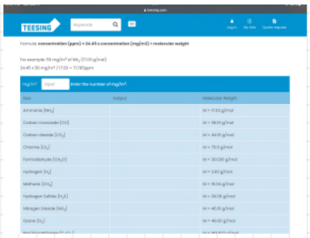
MW = 28.01 g/mol   MW = 48.01 g/mol   MW: 64.06 g/mol

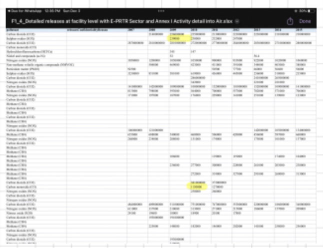
$V = 22.41 * (25 + 273.15) \times \frac{101.325}{101.325}$ T = 25°C, P = 101.325 kPa

$\therefore V = 6681.5415$ Air flow = 100 ft/min → 20 ft/min

$= 20 \frac{ft}{min} \times$

NCMH = V × A





convert tons to mg   1 tons = 9.072×10^8

$SO_x = 239000 \frac{tons}{year} = 2.168 \times 10^{14} \frac{mg}{year}$ $CO = 1\,150\,000 \frac{tons}{year} = 1.043 \times 10^{15} \frac{mg}{year}$

Need to convert SO_x from years to hours & CO from years to 8 hours

$SO_x = 2.168 \times 10^{14} \div (365 \times 24)$ $CO = 1.043 \times 10^{15} \div (365 \times 3)$

$= 2.474 \times 10^{10}$ mg / hour $= 9.527 \times 10^{11}$ mg / 8 hour

then divide it by volume to get mg/m³ so I can find ppm

$SO_x = 2.474 \times 10^{10} \div (4/3\, \pi (5.035)^3$ CO = 1 781 841 753 mg/m³

= 46 271 402.29 mg/m³

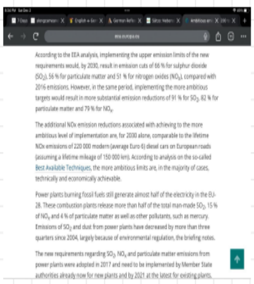
$PPm = (24.45 \times SO_x) \div 64.06$ $PPm = (24.45 \times CO) \div 28.01$

PPm = 17 660 565 $PPm = 1.55 \times 10^6$

I dont know if we have to divid that value by 10^6 cause of it be parts per million

$$\text{Moles of } SO_2 = \frac{27.283}{64.06} \times$$

= 425 897 moles



We follow the new standards of a power plant 56% of the standard

$20\ \mu g/m^3 \times (1.0 - 0.56) = 8.8\ \mu g/m^3$ year (÷ by 365 to get daily)

$0.0241\ \mu g/m^3$ per day   we add this value to the predicted values and get a new AQI

$5.04\ \mu g/m^3$ per day $+ 0.0241\ \mu g/m^3$ per day $= 5.0641\ \mu g/m^3$

Now convert new concentration to AQI value

$$I_p = \left(\frac{50 - 0}{12 - 0}\right)(5.0641 - 0) + 0$$

$I_p = 21.10$ ∴ New AQI rating will be 21.1 rather than 21 after adding a new power plant in Washington

D. *Link for other files:*
https://drive.google.com/drive/folders/10FUB6K4tSjzNQlmahnY6enndA-ZeZEqo?usp=sharing