A MINI PROJECT REPORT ON

AIR QUALITY INDEX PREDICTION

USING MACHINE LEARNING

SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF
BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND TECHNOLOGY

BY

Vedika Patil - 46, R A Ramya Rajeshwari - 50, Pranjali Shelke - 58

UNDER THE GUIDANCE OF

Prof.Sonali Bodekar

USHA MITTAL INSTITUTE OF TECHNOLOGY S. N. D. T.
WOMEN'S UNIVERSITY MUMBAI – 400049

April 2022

CERTIFICATE

This is to certify that Ms.Vedika Patil, Ms. R A Ramya Rajeshwari and Ms.Pranjali Shelke have completed the project report on the topic Air Quality Index Prediction Using Machine Learning satisfactorily in partial fulfilment for the Bachelor's Degree in Computer Science and Technology under the guidance of Prof. Sonali Bodekar during the year 2021-22 as prescribed by SNDT Women's University.

Guide
Prof.Sonali Bodekar

Head of department
Prof.Kumud Wasnik

Principal Dr.Shikha Nema

Dr.Shikha Nema

Examiner 1

Examiner 2

ACKNOWLEDGEMENT

We pay our deep sense of gratitude to our guide Prof.Ms.Sonali Bodekar who encouraged and guided us throughout the project. Our sincere thanks to our Principal Dr.Shikha Nema and The Head of the Computer Science and Technology Department Prof.Ms.Kumud Wasnik for setting us a platform to explore. Last but not the least, sincere thanks to all our Teaching staff ,non teaching staff, family and friends who shared their support morally. With due regards, thank you to all the students of UMIT for helping us with our survey.

(Signature)

Vedika Patil

R A Ramya Rajeshwari

Pranjali Shelke

Date : April 25,2022

INDEX

LIST OF FIGURES AND TABLES

ABSTRACT

The aim of this project is to use a Random Forest Classifier technique for Air Quality Index prediction in various Indian cities. (Cohen et al., 2020).

The main focus is to calculate the Air Quality Indexes in various urban regions and categorising them in categories such as good, poor, severe and so on based on the health impacts caused by those AQI values of air. And ultimately plotting them in a city wise manner to find out the most polluted city in India over the years 2015-2020.

A random forest algorithm is an ensemble method for regression and classification. We will determine the presence of air pollutants like carbon mono-oxide, carbon dioxide, nitrogen ,sulphur dioxide, ozone, pm2.5, pm10, Benzene, xylene etc. Our model will take new values and predict the result from them. But will also predict the air qualities of the existing cities in the dataset and will help to find out the most polluted city.

So, in our paper we will talk about Random forest Technique to predict and Measure Pollution. This algorithm is also used for data training and prediction.

PROBLEM STATEMENT

Air Quality Index is an index that is used for reporting daily air quality. In India, the Central and State Pollution Control Boards have commissioned the National Air Monitoring Program (NAMP) which covers 240 cities with 342 monitoring stations. (Cohen et al., 2020).

With the help of indexes, we know how clean or unhealthy our air is, and what associated health effects might be a concern. The Air Quality Index mainly focuses on the quality of air within our environment. Most of the popular cities in India have the worst quality index just because of the increasing population and pollution.

AQI is calculated for four major air pollutants regulated by the Clean Air Act: particle pollution, ground-level ozone sulphur dioxide, and carbon monoxide. EPA has established national air quality standards to protect public health for each of these pollutants.

Air quality index (AQI) is the concentration of all air pollutants.It's a single number and easy to understand. Based on AQI, the area can be categorised into: good, satisfying, moderately polluted, poor, very poor and severe.

Once the model is trained we can easily predict the air qualities of the various cities in India and can find out the health impacts that can be caused by such quality of air. Also the system will take new datas i.e. new values of AQI can be given to the system to predict the quality of air for unknown regions.

OBJECTIVES:

1. Comparing air quality conditions at different locations/cities. (Cohen et al., 2020).

2. It also helps in identifying faulty standards and inadequate monitoring programmes.

3. AQI helps in analysing the change in air quality (improvement or degradation).

4. AQI informs the public about environmental conditions. It is especially useful for people suffering from illnesses aggravated or caused by air pollution.

5. Create a Model to Predict the quality of air(Classification to various category like Good,moderate etc)

LITERATURE SURVEY

1] Air pollution represents the biggest environmental risk to health. Approximately 92 of the world population live in places where air quality levels exceed WHO limits. Air pollution is one of the largest causes of the top four non-communicable diseases such as stroke, lung cancer, chronic respiratory disease, and heart disease. In 2012, one out of 9 deaths was the result of air pollution-related diseases. Over half of deaths among children less than 5 years old from acute lower respiratory infections are due to particulate matter inhaled from indoor air pollution from household solid fuels. More than 660 million Indians breathe air that fails India's National Air Quality Standards.Research suggests that meeting those standards would increase life expectancy in India by 1 year. Going further and meeting the international benchmarks of the World Health Organisation is estimated to add 4.7 years to life expectancy.According to this global estimation, in Mongolia 1123 people die from air pollution related diseases each year Authors Kleine Deters, J., Zalakeviciute, R., Gonzalez, M. and Rybarczyk, Y. in their research paper have proposed the research of outdoor pollution causing millions of premature deaths due to anthropogenic fine particulate matter or PM 2.5 in the capital city of Ecuador. A machine learning approach based on six years of meteorological and pollution data analysis to predict concentration of PM 2.5. (Cohen et al., 2020).

2] A paper published in International Research Journal of Engineering Technology (IRJET) in 2019, used Data Mining techniques such as: data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation and knowledge presentation to build a system to calculate the polluted city based on: Particulate matter (PM2.5 and PM10), NO2, SO2, CO, O3 present in the air. The system would take data from the current values of: Temperature, Wind speed, relative Humidity, Traffic index, Air quality of the previous day. The system is based on Multivariate Multi Step Time Series prediction using the Random Forest algorithm. Multivariate is a popular statistical tool that uses multiple variables to forecast possible outcomes. Predicting multiple time steps into the future is called multi step time series forecasting. Time series provide better understanding of a database. In this existing system, time series decomposes into four constituent parts: Level, Trend, Seasonality, Noise. Whereas Random forest algorithm is a supervised classification algorithm which creates forest of trees, the more the trees more accurate are the results of the system

3] A Springer paper published in China in 2021 used Convolution Neural Networking technology mainly focusing on PM2.5 air pollutants based on two information: 1] Geographic location of all monitoring stations and 2] PM 2.5 concentrations recorded as a time sequence at each monitoring station. The system proposes a Deep Neural Network which is modified for a pixel-wise regression and provides a possible way to predict a pollution map accordingly. The model uses DenseNet to connect the layers with each other directly which makes flow of information and reuse of features possible and the system becomes more computationally efficient and enables the ability of feature extraction which gives better performance than traditional convolution neural networks. The system has four parts: 1] A head layer 2] A dense connection block (dense block) 3] A skip connection block 4] A pixel level regression The system uses Radial Basis Function (RBF) instead of SVR (which is a regressor which minimises a hinge loss) to increase the ability of non-linearity fit.

4] A Springer paper published in 2018 based on a case study of Delhi city used the Multivariable linear regression model of ELM (Extreme Machine Learning). The system predicted the Air quality index for PM10, PM2.5, NO2, CO, O3. The model used the previous day air quality index of pollutants and meteorological conditions for predictions. When the model was compared with the existing prediction system and actual values of the next day, the ELM based prediction had greater accuracy. Drawback: The drawback the system had was, ELM is much fast to train but cannot encode more than one layer of abstraction so it cannot form Deep network

5] A paper published in 2018, Forecasting air pollution load in Delhi using Data Analysis by Elseviere, collected data sets form Central Pollution Control Board (CPCB). The data was processed to remove any redundancy and further it followed pre-processing of data, which included: parsing of dates, noise removal, cleaning the data, training and scaling of the data. This system used the Time Series Regression method and a predictive analysis was done. Drawback: The system is using regression which is limited to the linear relationship and hence it is easily affected by the outliers. The system was basically designed to analyse the data and observing the outcomes rather than predicting the values

6] One of the notable works regarding AQI monitoring is YYang et al. implemented a mobile AQI monitoring system using the Gaussian plume model based on the neural network. developed an ImgSensingNet, a vision-based aerial-ground sensing system for AQI monitoring and forecasting by the fusion of images taken from the Unmanned Aerial-Vehicle (UAVs). in an urban city, designed a 3-Dimensional (3D) Real-time AQI monitoring using the Adaptive Gaussian Plume (AGPM) model with the help of Unmanned Aerial-Vehicle (UAVs). Z Hu developed real-time, fine-grained, power-efficient air quality sensing for the smart city and compares the ground sensing data and aerial sensing data to improve the data collected. M Khashei et al. presented the

performance analysis and comparison of both the ARIMA model and the Artificial Neural Network (ANN) model for various data sets like a sunspot, Canadian lynx, and dollar exchange to forecast future values. Neural networks are challenging to deal with because of a complex nature. Also, it is not suitable for real-time data change for a short period as per the literature survey.

7] A study using machine learning to classify the Air Quality Level in Beijing city was conducted by the author applied Random Forest, Support Vector Machine, and ranking methods to determine the top pollutants such as CO, PM2.5 and PM10. The experiment produced about 95 accuracy when using the SVM algorithm. A recent study used several machine learning algorithms to prove the ability of computing methods to determine air pollutant index. The authors used four different algorithms: Neural Network (NN), k-Nearest Neighbours algorithm (kNN), Decision Tree (DT) and SVM. The experiment was performed on a dataset collected from the state of Macedonia in 2017; they found that NN produced better accuracy of 0.92, KNN and SVM approximately 0.8, and DT 0.78. The study has shown the potential of using machine learning methods to improve air quality forecasts.

INTRODUCTION

Knowing that modernization leads to modern growth, we know all the means of transportation to date. We are constantly dependent on fossil fuels such as gasoline, diesel, gas and CNG. There are so many people who use vehicles. There is a constant increase in air pollution due to harmful emissions when driving a vehicle .It emits gases such as carbon dioxide, carbon monoxide, and nitrogen. Along with the transportation sector, we have also observed massive growth in industrial zones and rapid decline in forest occupied areas. Which has led to a great number of air pollution and the quality of air has been affected badly. (Cohen et al., 2020).

Today, the air pollution in cities is very large. Pollutants have had such a negative impact on humans that problems are occurring in all countries of the world. Air pollution can also lead to acid rain and the greenhouse effect. Diseases like lung cancer are caused by these harmful pollutants. Especially in India these days in big cities like Old Delhi and industrial cities like Allahabad, pollution is a huge disadvantage.

The government has taken steps together to take action. The government regulates these issues by taking action against automakers as a language for forming less polluted vehicles. In addition, attempts are being made by the government to ban new vehicles to release additional pollutants.Along with the numerous regulations imposed on industries on releasing pollutant gases into the air.

EXISTING SYSTEM

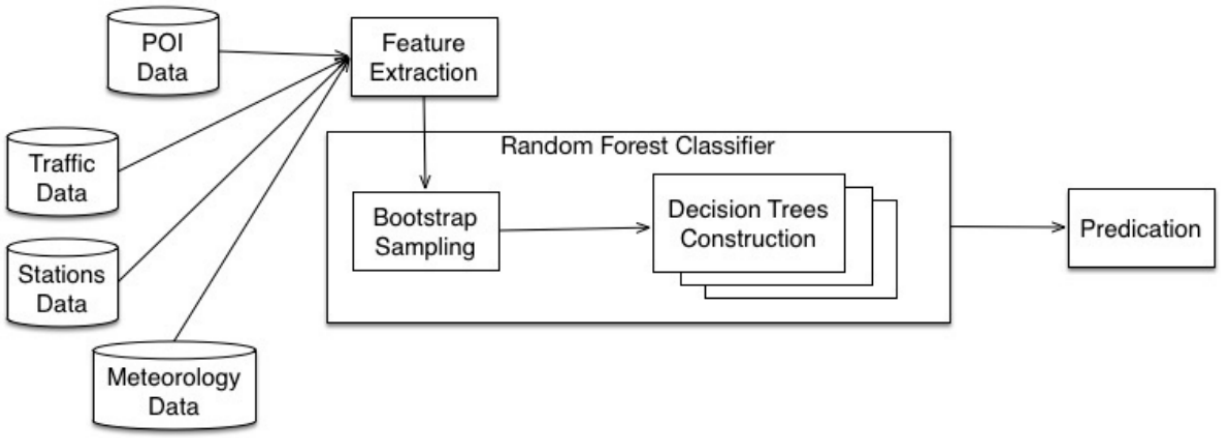


Figure 1: Existing system

The existing system is based on RAQ - a Random Forest Approach for predicting air quality in the Urban Sensing System. (Rojas, 2006).

Step 1: Data collection

The data is gathered from the Urban Sensing System. It includes: Point of Interest data (POI), traffic data and road information, Monitoring station data and lastly meteorology data.

A. Point of Interest (POI) data:

POI is a specific location at which someone finds their benefits or usually more people are interested at that location. E.g. restaurants, shopping malls etc. In the existing system this data helps to find the function of the particular region and traffic patterns at those regions. More crowded regions would cause more pollution and therefore contribute to air quality of that region.

B. Traffic and road data:

The main reason for air pollution is emission of carbon fuels due to heavy traffic in urban areas. Collecting the data such as size of the road, traffic hours, average emission from the road etc. would impact the air quality.

C. Monitoring station data:

Several monitoring stations can be set up at the industrial regions to calculate and predict the condition of air in heavy industry areas.

D. Meteorology data:

Data such as temperature, humidity and barometric pressure impact quality of air drastically. Understanding meteorological parameters is important because the atmosphere is the only medium where air pollutants directly get transported.

Step 2: Feature extraction

The data collected from various data points serve as a data set which is used to train the model which includes all the necessary features such as concentrations of important pollutants and average time of their occurrence in a particular time zone.

Step3: Bootstrap sampling

The trained data having necessary functions is divided into subsets and sampled using bootstrap technology. Bootstrap is a statistical re sampling technique that involves random sampling of a dataset with replacement. It is often used as a means of quantifying the uncertainty associated with the machine learning model.

Step 4: Decision tree construction (Tree growing and splitting)

A decision tree is constructed on each subsets created by Bootstrap sampling. Finally, the classification is done by aggregating the results generated from all the decision tree

Step 5: Prediction

The aggregated result is presented as the prediction of the model

Disadvantages of existing system :

The RAQ model in the existing system uses real time data such as POI data, Traffic data and also requires data from monitoring stations in the cities along with the knowledge of Meteorological factors. Getting all this data and gathering the physical apparatus is not possible in the current situation. The real time data will surely give better results but looking at the practical conditions, the current system is time and money consuming. Whereas the same results can be obtained by already available datasets.

PROPOSED SYSTEM

The proposed system is based on Machine Learning Technology of Random Forest Classifier for prediction of Air Quality Index. (Rojas, 2006).

The respective Workflow of project, Modules of the system, Methodology of the model and relationship with the users along with the details of machine learning algorithm used are discussed below:

Workflow:

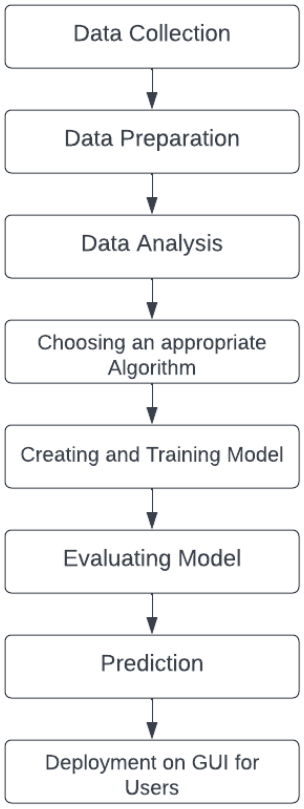


Figure 2: Workflow

The workflow of the project consists of 6 steps:

1. Data collection:

Collection of data from several resources such as available datasets.

2. Data Preparation:

Preparing data by removing the noise and extracting the features from the dataset. Also finding the missing values from the dataset.

3. Data Analysis:

Analysing and exploring the data by plotting the correlation images and graphs between the columns in the dataset.

4. Choosing an Appropriate Algorithm:

Comparing the results, accuracies, performances and accessibility of the algorithms used in previous works and choosing appropriate technology for the project.

5. Creating and Training the model:

Creating the model and training it based on available datasets and preparing it for predicting the results in the future

6. Evaluating the model:

Checking the accuracy of the system.

7. Prediction:

Predicting the results for the inputs given by the user

8. Deployment on GUI:

Finally, the model will be ready to be used by users and will have real-life application.

ARCHITECTURAL OVERVIEW

Modules of the System:

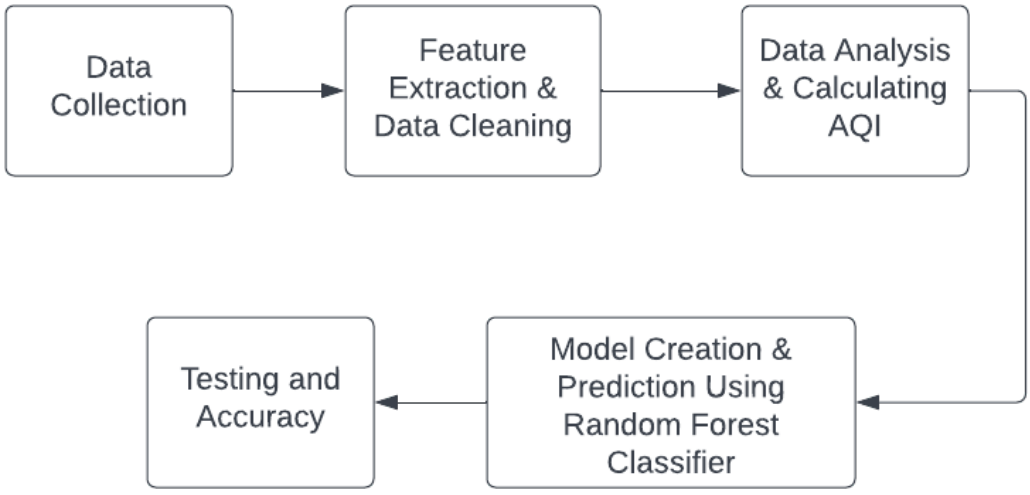


Figure 3: Modules of Existing System

The system is divided into 5 Modules:

Module 1: Data Collection:

As mentioned in the workflow this module will work on collecting the data from various resources (available dataset in our case). The data such as cities, details of pollutants and number of years taken into consideration.

Module 2: Feature extraction:

This module involves refinement and processing the data by removing noise from the dataset, finding missing values in the dataset and filling the null spaces

Module 3: Data Analysis and calculating AQI:

This module is one of the crucial steps. It includes analysing the data by displaying it pictorially by means of graphs and figures. Once the data is analysed we will calculate the AQI (Air Quality Index) and categorise them for further finding out the most polluted city.

Module 4: Model creation, training and prediction:

We will create the model and train it based on the AQI values and categories we calculated and data we have. Lastly, the model will predict the values for future inputs as well as push these values into the dataset.

Module 5: Testing and Accuracy:

The model will be tested for the predictions it is making and ultimately the accuracy of the model will be calculated

Methodology of the System:

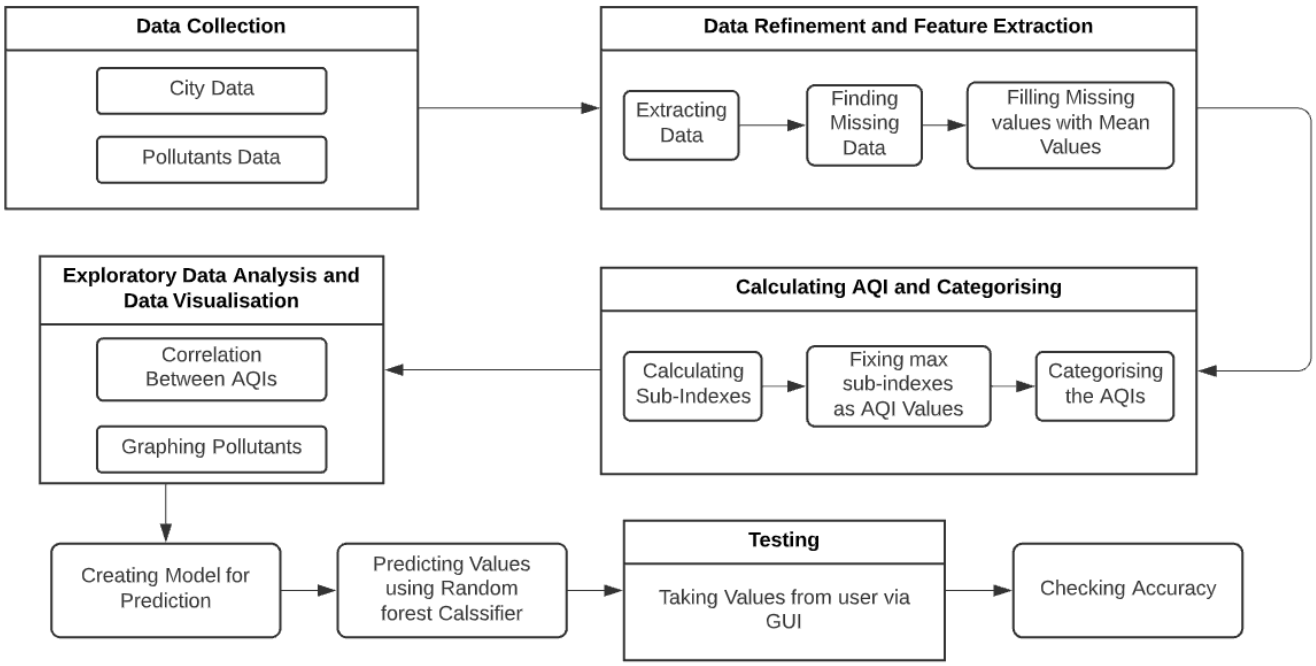


Figure 4: Methodology of Existing System

The detailed methodology of the model is divided into 8 steps:

Step 1: Data collection

In the proposed system we have collected data from the existing dataset available on Kaggle. The dataset contains following data: a. Information of pollutants for various cities in India from year 2015 to 2020 b. The concentration of more than 10 pollutants found on different dates in those cities c. It also contains two more empty columns of AQI and category from which the AQI belongs

Step 2: Data refinement and Feature extraction:

This step is further subdivided in:

a. Extracting data:

From the dataset we have extracted data of
–City

–Date
–PM2.5 (Particulate Matter 2.5-micrometer)
–PM10 (Particulate Matter 10-micrometre)
–SO2 (Sulphur Dioxide)
–NOx (Any Nitric x-oxide)
–NH3 (Ammonia)
–CO (Carbon Monoxide)
–O3 (Ozone or Trioxygen)
–Benzene
–Toluene
–Xylene
–AQI
–AQI Bucket

b. Finding missing values:

The empty spaces from the dataset will be identified

c. Filling missing values:

The missing values will be filled with mean values from the previous entries. Except the values in AQI columns.

Step 3: Calculating AQI

a. Calculating sub-indexes:

The sub-indexes of 7 pollutants will be calculated as follow:
–For PM2.5, PM10, SO2, NOx and NH3 the average value in the last 24-hrs is used with the condition of having at least 16 values.
–For CO and O3 the maximum value in the last 8-hrs is used.
–Each measure is converted into a Sub-Index based on pre-defined groups.

b. Filling AQI column:

Sometimes measures are not available due to lack of measuring or lack of required data points.
Final AQI is the maximum Sub-Index with the condition that at least one of PM2.5 and PM10

should be available and at least three out of the seven should be available.

c. Categorising AQIs:

Finally, the last column in the dataset will be filled by categorising the ranges of calculated AQI values into Good, Satisfactory, Moderate, Poor, Very Poor and Sever.

The categorization and health impact of AQI values are given in the table below:

AQI range	AQI category	Associated health impact
0-50	Good	Minimal impact
51-100	Satisfactory	May cause minor breathing discomfort to sensitive people
101-200	Moderate	May cause breathing discomfort to the people with lung disease such as asthma and discomfort to people with heart disease, children and older adults
201-300	Poor	May cause breathing discomfort to people on prolonged exposure and discomfort to people with heart disease
301-400	Very poor	May cause respiratory illness to the people on prolonged exposure. Effect may be more pronounced in people with lung and heart disease
401-500	Severe	May cause respiratory effects even on healthy people and serious health impacts on people with lung/heart disease. The health impacts may be experienced even during light physical activity

Step 4: Exploring data analysis and visualisation

Now, the dataset does not have any empty cells. So we are moving forward to visualising the data

a. A heatmap figure to see the correlation between the AQI and AQI categories
b. Plots for distribution of different pollutants separately for the 5 years
c. City wise graph of average AQI values in the 5 years

Step 5: Creating the model:

The Machine learning model will be created based on available values in the dataset. Basically, a model will be created to learn how to categorise AQI values in the respective categories.

Step 6: Training and predicting AQI categories:

As said earlier, the model will be trained for predicting good to severe categories for the given AQI values. We are using random forest classifier for that purpose:

Note on Algorithm and technology used:

For Air Quality Index Prediction we are using Random Forest Algorithm which is classification problem in Supervised Learning

Supervised Learning:

A supervised learning algorithm takes a known set of input data (the learning set) and known responses to the data (the output), and forms a model to generate reasonable predictions for the response to the new input data. Supervised learning is classified in:
1. Classification
2. Regression

There are many algorithms under supervised learning algorithms such as Linear Regression, Nearest Neighbour, SVM, kernel SVM, Naive Bayes and Random Forest. Compared to all other algorithms Random forest gives better results, so our approach selects Random Forest to predict the accurate air pollution.

Classification Problem:

A Classification algorithm aims to sort inputs into a given number of categories or classes, based on the labelled data it was trained on. Classification algorithms can be used for binary classifications and Feature recognition. Here AQI values should be understood according to the classification reported by standard agencies and based on the observed values of the pollutants and the predicted values, the AQI value per hour would be calculated for the training and validation sets.

Random Forest:

A Random Forest is ensemble classification which uses many trees to predict the final result of the specific problem. It uses recursive partitioning to generate many trees and then aggregate the results. Each tree is independently constructed using a bootstrap sample of the training data, which subdivides the parameter set first into several parts depending on one of the parameters, and subsequently repeats the process for each part.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max-samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

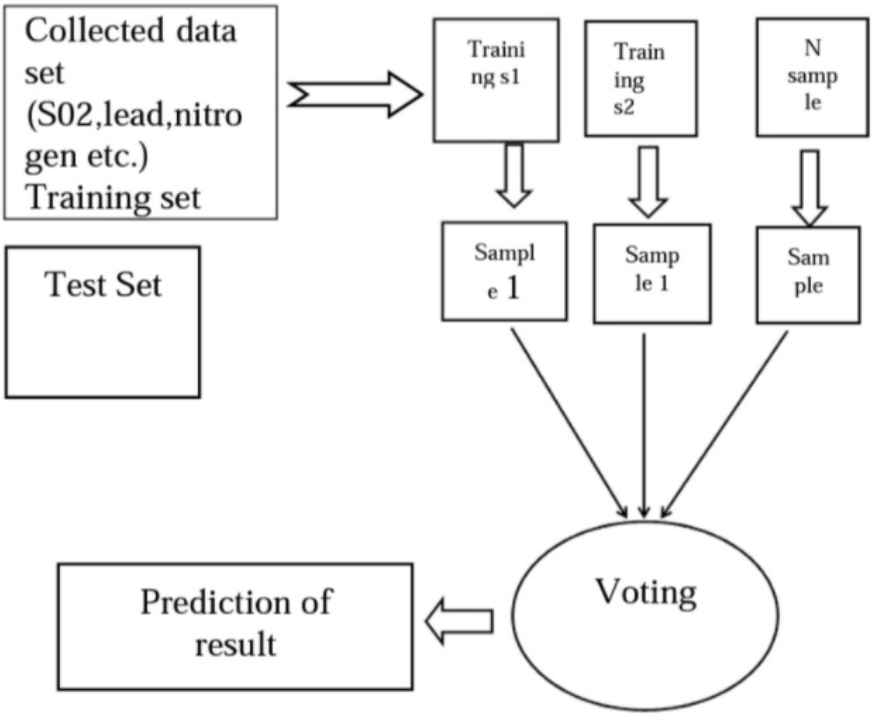


Figure 5: Working of Random Forest Classifier

Why Random Forest classifier?

1. A random forest produces good predictions that can be understood easily.
2. It can handle large datasets efficiently.
3. The random forest algorithm provides a higher level of accuracy in predicting outcomes over

the decision tree algorithm.

Step 7: Testing:

Testing if the predictions are correct by giving outside values of AQI into the code.

Step 8: Checking accuracy:

Using Confusion matrix to find out the accuracy of the code

Confusion Matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model (or ”classifier”) on a set of test data for which the true values are known.

also known as an error matrix, is a specific table layout that allows visualisation of the performance of an algorithm, typically a supervised learning one.

Relationship with the User:

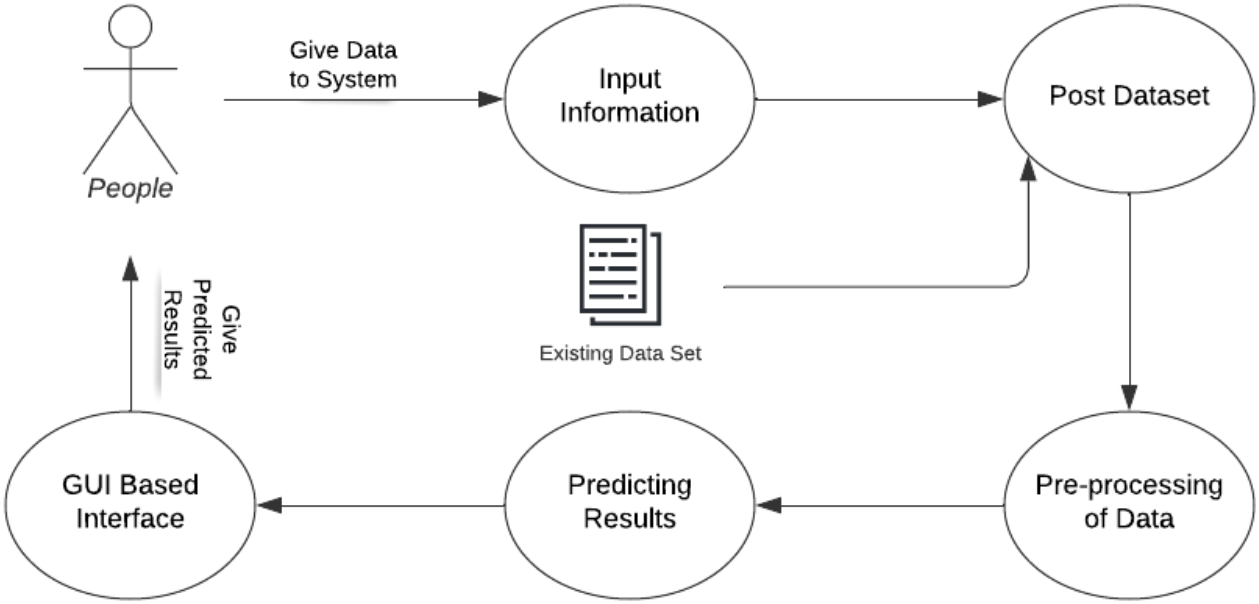


Figure 6: Use case diagram

At the end we will build a user interface in the form of a website. As shown in the figure,

1. The system will be available to people where they will give input to the system

2. Which will be pushed into the dataset. The dataset will add the input values and learn from it along with the existing values.

3. The data will be processed, in our case, the AQI will be calculated.

4. The model will predict the category of the AQI value

5. And finally, display it to the user.

HARDWARE AND SOFTWARE REQUIREMENTS

Software Requirements

1. Python -

Python is a high-level, general-purpose programming language. Its design philosophy emphasises code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects

2. Numpy -

import numpy as np

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

3. Pandas -

import pandas as pd

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD licence.

4. Matplotlib -

import matplotlib.pyplot as plt

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

5. Seaborn -

import seaborn as sns

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them

6. Sklearn -

!pip install sklearn

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

IMPLEMENTATION

Random Forest Algorithm Flowchart

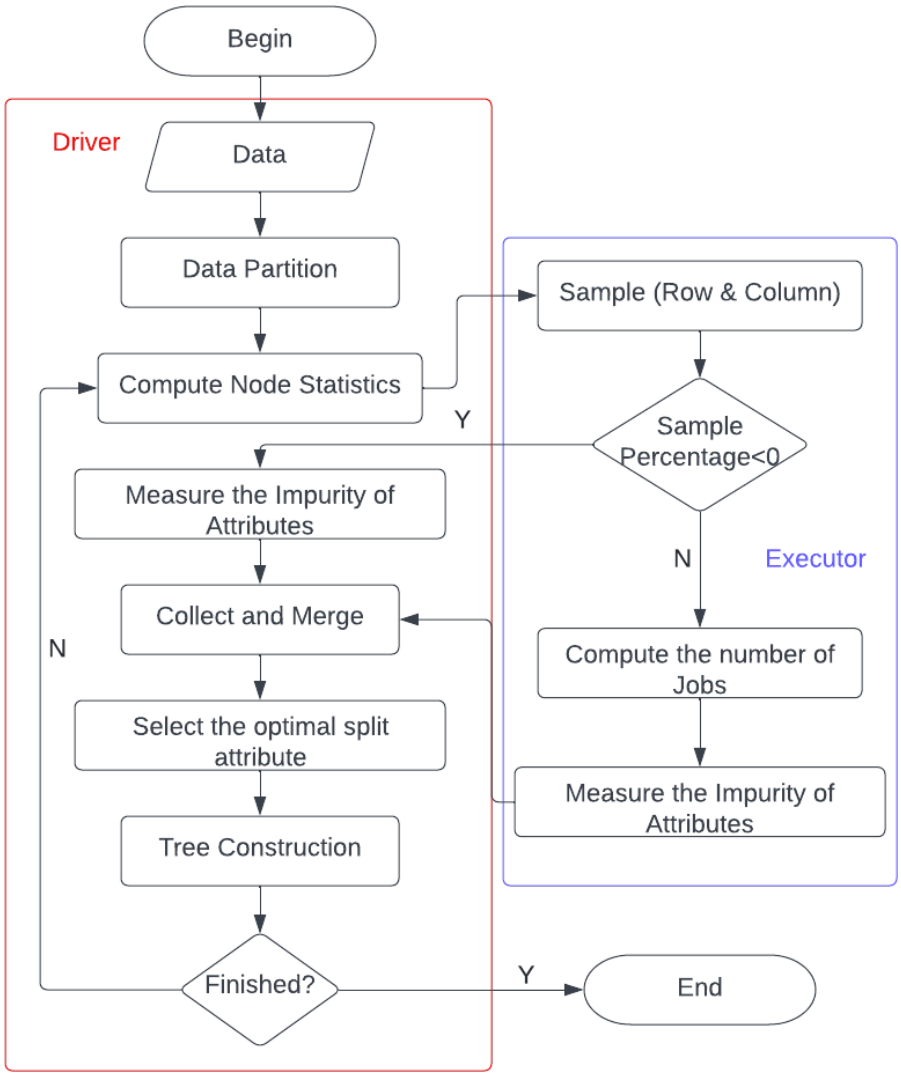
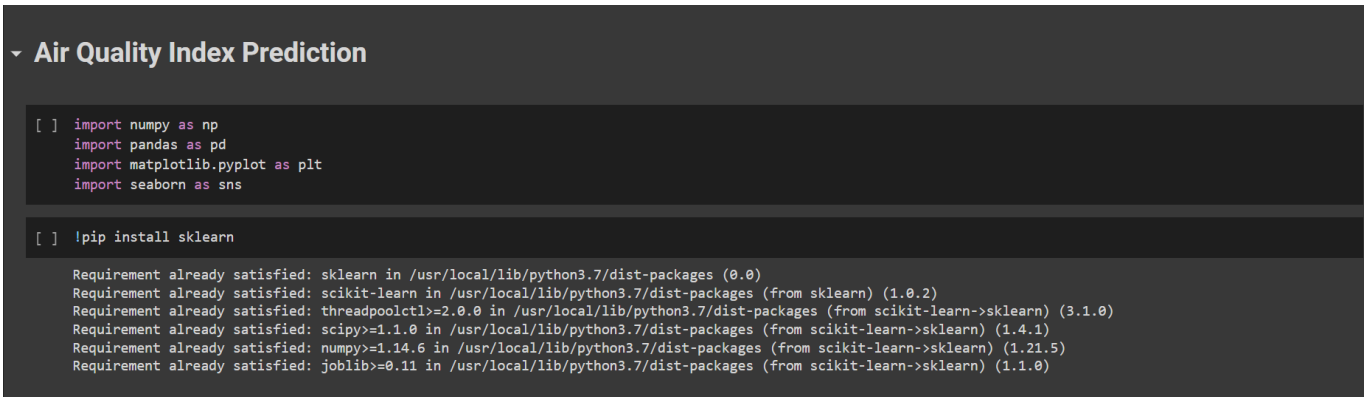


Figure 7: Random Forest Algorithm Flowchart

Importing Packages

Importing all the required libraries like numpy, pandas, matplotlib, seaborn, sklearn required for the project.



Air Quality Index Prediction

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

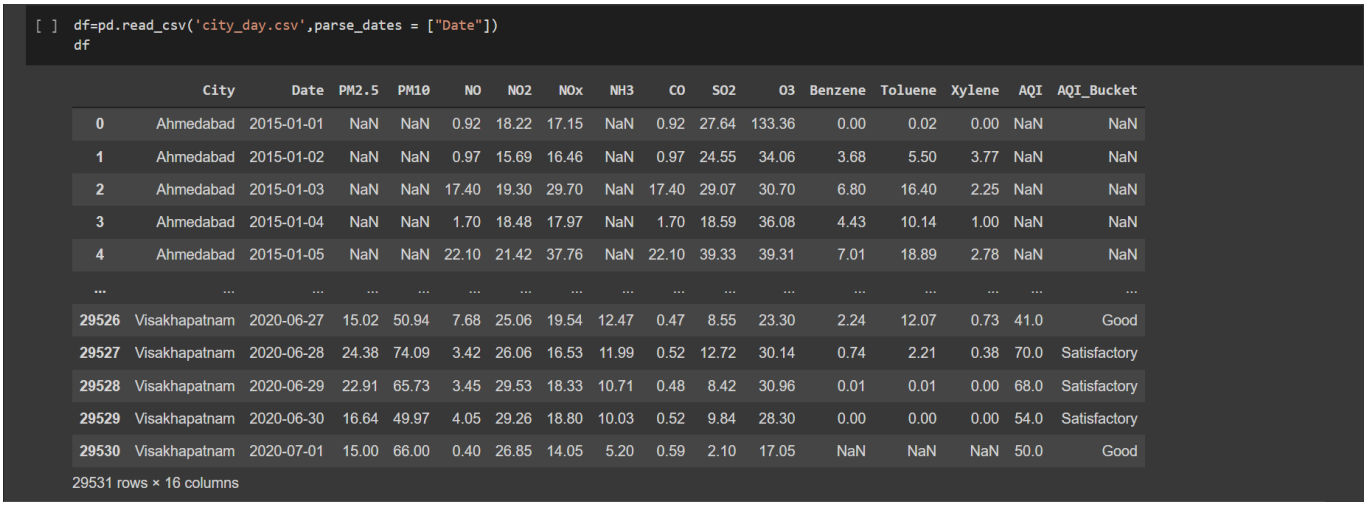
```
[ ] !pip install sklearn
```

```
Requirement already satisfied: sklearn in /usr/local/lib/python3.7/dist-packages (0.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (from sklearn) (1.0.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->sklearn) (3.1.0)
Requirement already satisfied: scipy>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->sklearn) (1.4.1)
Requirement already satisfied: numpy>=1.14.6 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->sklearn) (1.21.5)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->sklearn) (1.1.0)
```

Figure 8: Importing packages

Data Frames

This is the dataset used, which is taken from Kaggle. It lists the details of Cities and the amount of Pollutants present on a daily basis.



```
[ ] df=pd.read_csv('city_day.csv',parse_dates = ["Date"])
    df
```

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29526	Visakhapatnam	2020-06-27	15.02	50.94	7.68	25.06	19.54	12.47	0.47	8.55	23.30	2.24	12.07	0.73	41.0	Good
29527	Visakhapatnam	2020-06-28	24.38	74.09	3.42	26.06	16.53	11.99	0.52	12.72	30.14	0.74	2.21	0.38	70.0	Satisfactory
29528	Visakhapatnam	2020-06-29	22.91	65.73	3.45	29.53	18.33	10.71	0.48	8.42	30.96	0.01	0.01	0.00	68.0	Satisfactory
29529	Visakhapatnam	2020-06-30	16.64	49.97	4.05	29.26	18.80	10.03	0.52	9.84	28.30	0.00	0.00	0.00	54.0	Satisfactory
29530	Visakhapatnam	2020-07-01	15.00	66.00	0.40	26.85	14.05	5.20	0.59	2.10	17.05	NaN	NaN	NaN	50.0	Good

29531 rows × 16 columns

Figure 9: Data Frames

Finding Missing Values

The missing values need to be replaced by numeric values in order to apply ML Algorithms

```
[ ] sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f611e9f0a50>
```

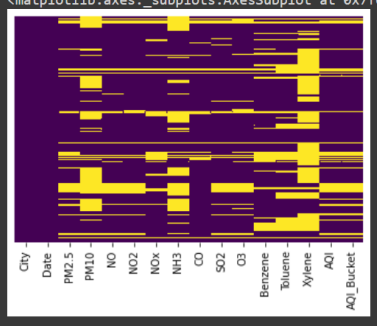


Figure 10: Finding Missing Values

```
[ ] print(df.isnull().sum())
```

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

Figure 11: Finding Missing Values

Filling Missing Values

The identified missing values should be replaced with appropriate values like mean etc.

Filling the values by taking mean of entries on that particular date i.e. mean of the rows in the dataset.

Filling the NaN values present in the pollutants with mean (city wise)

```
[ ] df.iloc[:,1:13] = df.groupby("City").transform(lambda x: x.fillna(x.mean()))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping invalid columns in DataFrameGroupBy.transform is deprecated. In a future
  """Entry point for launching an IPython kernel.
```

```
[ ] df
```

Date	City	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
2015-01-01	Ahmedabad	67.854497	114.584029	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.000000	0.020000	0.000000	NaN	NaN
2015-01-02	Ahmedabad	67.854497	114.584029	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.680000	5.500000	3.770000	NaN	NaN
2015-01-03	Ahmedabad	67.854497	114.584029	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.800000	16.400000	2.250000	NaN	NaN
2015-01-04	Ahmedabad	67.854497	114.584029	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.430000	10.140000	1.000000	NaN	NaN
2015-01-05	Ahmedabad	67.854497	114.584029	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.010000	18.890000	2.780000	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2020-06-27	Visakhapatnam	15.020000	50.940000	7.68	25.06	19.54	12.47	0.47	8.55	23.30	2.240000	12.070000	0.730000	41.0	Good
2020-06-28	Visakhapatnam	24.380000	74.090000	3.42	26.06	16.53	11.99	0.52	12.72	30.14	0.740000	2.210000	0.380000	70.0	Satisfactory
2020-06-29	Visakhapatnam	22.910000	65.730000	3.45	29.53	18.33	10.71	0.48	8.42	30.96	0.010000	0.010000	0.000000	68.0	Satisfactory

Figure 12: Filling Missing Values

```
[ ] sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f611f0db450>
```

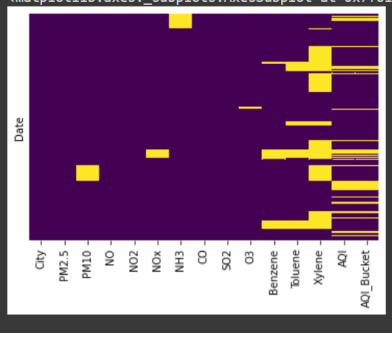


Figure 13: Filling Missing Values

Filling the values by taking mean of the columns of dataset.

```
[ ] df.iloc[:, 0:13]=df.fillna(df.mean())
    df
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a futu
  """Entry point for launching an IPython kernel.
```

Date	City	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
2015-01-01	Ahmedabad	67.854497	114.584029	0.92	18.22	17.15	23.024137	0.92	27.64	133.36	0.000000	0.020000	0.000000	NaN	NaN
2015-01-02	Ahmedabad	67.854497	114.584029	0.97	15.69	16.46	23.024137	0.97	24.55	34.06	3.680000	5.500000	3.770000	NaN	NaN
2015-01-03	Ahmedabad	67.854497	114.584029	17.40	19.30	29.70	23.024137	17.40	29.07	30.70	6.800000	16.400000	2.250000	NaN	NaN
2015-01-04	Ahmedabad	67.854497	114.584029	1.70	18.48	17.97	23.024137	1.70	18.59	36.08	4.430000	10.140000	1.000000	NaN	NaN
2015-01-05	Ahmedabad	67.854497	114.584029	22.10	21.42	37.76	23.024137	22.10	39.33	39.31	7.010000	18.890000	2.780000	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2020-06-27	Visakhapatnam	15.020000	50.940000	7.68	25.06	19.54	12.470000	0.47	8.55	23.30	2.240000	12.070000	0.730000	41.0	Good
2020-06-28	Visakhapatnam	24.380000	74.090000	3.42	26.06	16.53	11.990000	0.52	12.72	30.14	0.740000	2.210000	0.380000	70.0	Satisfactory
2020-06-29	Visakhapatnam	22.910000	65.730000	3.45	29.53	18.33	10.710000	0.48	8.42	30.96	0.010000	0.010000	0.000000	68.0	Satisfactory
2020-06-30	Visakhapatnam	16.640000	49.970000	4.05	29.26	18.80	10.030000	0.52	9.84	28.30	0.000000	0.000000	0.000000	54.0	Satisfactory
2020-07-01	Visakhapatnam	15.000000	66.000000	0.40	26.85	14.05	5.200000	0.59	2.10	17.05	3.891348	8.350983	2.790456	50.0	Good

29531 rows × 15 columns

Figure 14: Filling Missing Values

Dataset completely filled.

```
[ ] sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f611ed0e110>
```

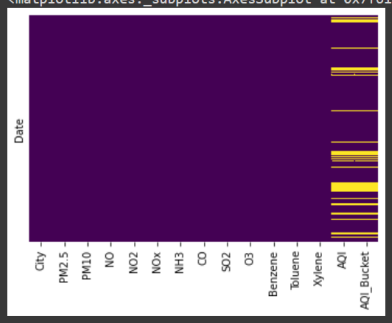


Figure 15: Filling Missing Values

Calculating AQI

Calculating the AQI for every pollutant

PM10

```python
# PM10 Sub-Index calculation
def get_PM10_subindex(x):
    if x <= 50:
        return x
    elif x > 50 and x <= 100:
        return x
    elif x > 100 and x <= 250:
        return 100 + (x - 100) * 100 / 150
    elif x > 250 and x <= 350:
        return 200 + (x - 250)
    elif x > 350 and x <= 430:
        return 300 + (x - 350) * 100 / 80
    elif x > 430:
        return 400 + (x - 430) * 100 / 80
    else:
        return 0

df["PM10_SubIndex"] = df["PM10"].astype(int).apply(lambda x: get_PM10_subindex(x))
```

Figure 16: AQI of PM10

PM2.5

```python
# PM2.5 Sub-Index calculation
def get_PM25_subindex(x):
    if x <= 30:
        return x * 50 / 30
    elif x > 30 and x <= 60:
        return 50 + (x - 30) * 50 / 30
    elif x > 60 and x <= 90:
        return 100 + (x - 60) * 100 / 30
    elif x > 90 and x <= 120:
        return 200 + (x - 90) * 100 / 30
    elif x > 120 and x <= 250:
        return 300 + (x - 120) * 100 / 130
    elif x > 250:
        return 400 + (x - 250) * 100 / 130
    else:
        return 0
```

Figure 17: AQI of PM2.5

SO2

```python
# SO2 Sub-Index calculation
def get_SO2_subindex(x):
    if x <= 40:
        return x * 50 / 40
    elif x > 40 and x <= 80:
        return 50 + (x - 40) * 50 / 40
    elif x > 80 and x <= 380:
        return 100 + (x - 80) * 100 / 300
    elif x > 380 and x <= 800:
        return 200 + (x - 380) * 100 / 420
    elif x > 800 and x <= 1600:
        return 300 + (x - 800) * 100 / 800
    elif x > 1600:
        return 400 + (x - 1600) * 100 / 800
    else:
        return 0

df["SO2_SubIndex"] = df["SO2"].astype(int).apply(lambda x: get_SO2_subindex(x))
```

Figure 18: AQI of SO2

NOx

```python
# NOx Sub-Index calculation
def get_NOx_subindex(x):
    if x <= 40:
        return x * 50 / 40
    elif x > 40 and x <= 80:
        return 50 + (x - 40) * 50 / 40
    elif x > 80 and x <= 180:
        return 100 + (x - 80) * 100 / 100
    elif x > 180 and x <= 280:
        return 200 + (x - 180) * 100 / 100
    elif x > 280 and x <= 400:
        return 300 + (x - 280) * 100 / 120
    elif x > 400:
        return 400 + (x - 400) * 100 / 120
    else:
        return 0

df["NOx_SubIndex"] = df["NOx"].astype(int).apply(lambda x: get_NOx_subindex(x))
```

Figure 19: AQI of NOx

NH3

```python
# NH3 Sub-Index calculation
def get_NH3_subindex(x):
    if x <= 200:
        return x * 50 / 200
    elif x > 200 and x <= 400:
        return 50 + (x - 200) * 50 / 200
    elif x > 400 and x <= 800:
        return 100 + (x - 400) * 100 / 400
    elif x > 800 and x <= 1200:
        return 200 + (x - 800) * 100 / 400
    elif x > 1200 and x <= 1800:
        return 300 + (x - 1200) * 100 / 600
    elif x > 1800:
        return 400 + (x - 1800) * 100 / 600
    else:
        return 0

df["NH3_SubIndex"] = df["NH3"].astype(int).apply(lambda x: get_NH3_subindex(x))
```

Figure 20: AQI of NH3

CO

```python
# CO Sub-Index calculation
def get_CO_subindex(x):
    if x <= 1:
        return x * 50 / 1
    elif x > 1 and x <= 2:
        return 50 + (x - 1) * 50 / 1
    elif x > 2 and x <= 10:
        return 100 + (x - 2) * 100 / 8
    elif x > 10 and x <= 17:
        return 200 + (x - 10) * 100 / 7
    elif x > 17 and x <= 34:
        return 300 + (x - 17) * 100 / 17
    elif x > 34:
        return 400 + (x - 34) * 100 / 17
    else:
        return 0
```

Figure 21: AQI of CO

O3

```
# O3 Sub-Index calculation
def get_O3_subindex(x):
    if x <= 50:
        return x * 50 / 50
    elif x > 50 and x <= 100:
        return 50 + (x - 50) * 50 / 50
    elif x > 100 and x <= 168:
        return 100 + (x - 100) * 100 / 68
    elif x > 168 and x <= 208:
        return 200 + (x - 168) * 100 / 40
    elif x > 208 and x <= 748:
        return 300 + (x - 208) * 100 / 539
    elif x > 748:
        return 400 + (x - 400) * 100 / 539
    else:
        return 0

df["O3_SubIndex"] = df["O3"].astype(int).apply(lambda x: get_O3_subindex(x))
```

Figure 22: AQI of O3

Filling calculated AQI values into the dataset

Filling the NaN values of AQI column by taking maximum values out of Sub-Indexes

```
[ ] df["AQI"] = df["AQI"].fillna(round(df[["PM2.5_SubIndex", "PM10_SubIndex", "SO2_SubIndex", "NOx_SubIndex","NH3_SubIndex", "CO_SubIndex", "O3_SubIndex"]].max(axis
```

```
[ ] df
```

Date	City	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	...	Xylene	AQI	AQI_Bucket	PM10_SubIndex	PM2.5_SubIndex	SO2_S
2015-01-01	Ahmedabad	67.854497	114.584029	0.92	18.22	17.15	23.024137	0.92	27.64	133.36	...	0.000000	149.0	NaN	109.333333	123.333333	
2015-01-02	Ahmedabad	67.854497	114.584029	0.97	15.69	16.46	23.024137	0.97	24.55	34.06	...	3.770000	123.0	NaN	109.333333	123.333333	
2015-01-03	Ahmedabad	67.854497	114.584029	17.40	19.30	29.70	23.024137	17.40	29.07	30.70	...	2.250000	300.0	NaN	109.333333	123.333333	
2015-01-04	Ahmedabad	67.854497	114.584029	1.70	18.48	17.97	23.024137	1.70	18.59	36.08	...	1.000000	123.0	NaN	109.333333	123.333333	
2015-01-05	Ahmedabad	67.854497	114.584029	22.10	21.42	37.76	23.024137	22.10	39.33	39.31	...	2.780000	329.0	NaN	109.333333	123.333333	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2020-06-27	Visakhapatnam	15.020000	50.940000	7.68	25.06	19.54	12.470000	0.47	8.55	23.30	...	0.730000	41.0	Good	50.000000	25.000000	

Figure 23: Filling AQI values

```
[ ]  sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f611e89c090>
```

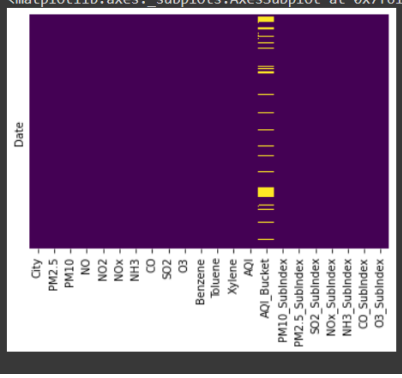


Figure 24: Filling AQI values

Filing AQI Categories

Calculating AQI bucket and Filling the NAN values present

```
[ ]  ## AQI bucketing
     def get_AQI_bucket(x):
         if x <= 50:
             return "Good"
         elif x > 50 and x <= 100:
             return "Satisfactory"
         elif x > 100 and x <= 200:
             return "Moderate"
         elif x > 200 and x <= 300:
             return "Poor"
         elif x > 300 and x <= 400:
             return "Very Poor"
         elif x > 400:
             return "Severe"
         else:
             return '0'

     df["AQI_Bucket"] = df["AQI_Bucket"].fillna(df["AQI"].apply(lambda x: get_AQI_bucket(x)))
```

Figure 25: Filling AQI categories

```
df
```

Date	City	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	...	Xylene	AQI	AQI_Bucket	PM10_SubIndex	PM2.5_SubIndex	SO2_S
2015-01-01	Ahmedabad	67.854497	114.584029	0.92	18.22	17.15	23.024137	0.92	27.64	133.36	...	0.000000	149.0	Moderate	109.333333	123.333333	
2015-01-02	Ahmedabad	67.854497	114.584029	0.97	15.69	16.46	23.024137	0.97	24.55	34.06	...	3.770000	123.0	Moderate	109.333333	123.333333	
2015-01-03	Ahmedabad	67.854497	114.584029	17.40	19.30	29.70	23.024137	17.40	29.07	30.70	...	2.250000	300.0	Poor	109.333333	123.333333	
2015-01-04	Ahmedabad	67.854497	114.584029	1.70	18.48	17.97	23.024137	1.70	18.59	36.08	...	1.000000	123.0	Moderate	109.333333	123.333333	
2015-01-05	Ahmedabad	67.854497	114.584029	22.10	21.42	37.76	23.024137	22.10	39.33	39.31	...	2.780000	329.0	Very Poor	109.333333	123.333333	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2020-06-27	Visakhapatnam	15.020000	50.940000	7.68	25.06	19.54	12.470000	0.47	8.55	23.30	...	0.730000	41.0	Good	50.000000	25.000000	
2020-06-28	Visakhapatnam	24.380000	74.090000	3.42	26.06	16.53	11.990000	0.52	12.72	30.14	...	0.380000	70.0	Satisfactory	74.000000	40.000000	

Figure 26: Filling AQI categories

```
[ ] sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f611eb06250>
```

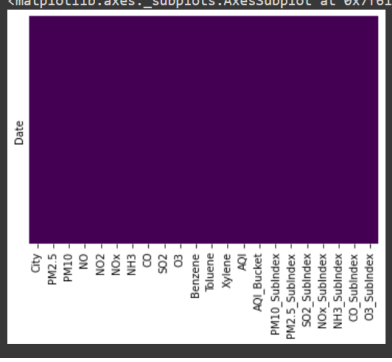


Figure 27: Filling AQI categories

Visualizing data

Visualising the data using Heat Maps and Graphs to identify patterns, distribution of different pollutants in the last 5 years and thus finding the most polluted city.

Heatmap for correlation between AQI and AQI Bucket column

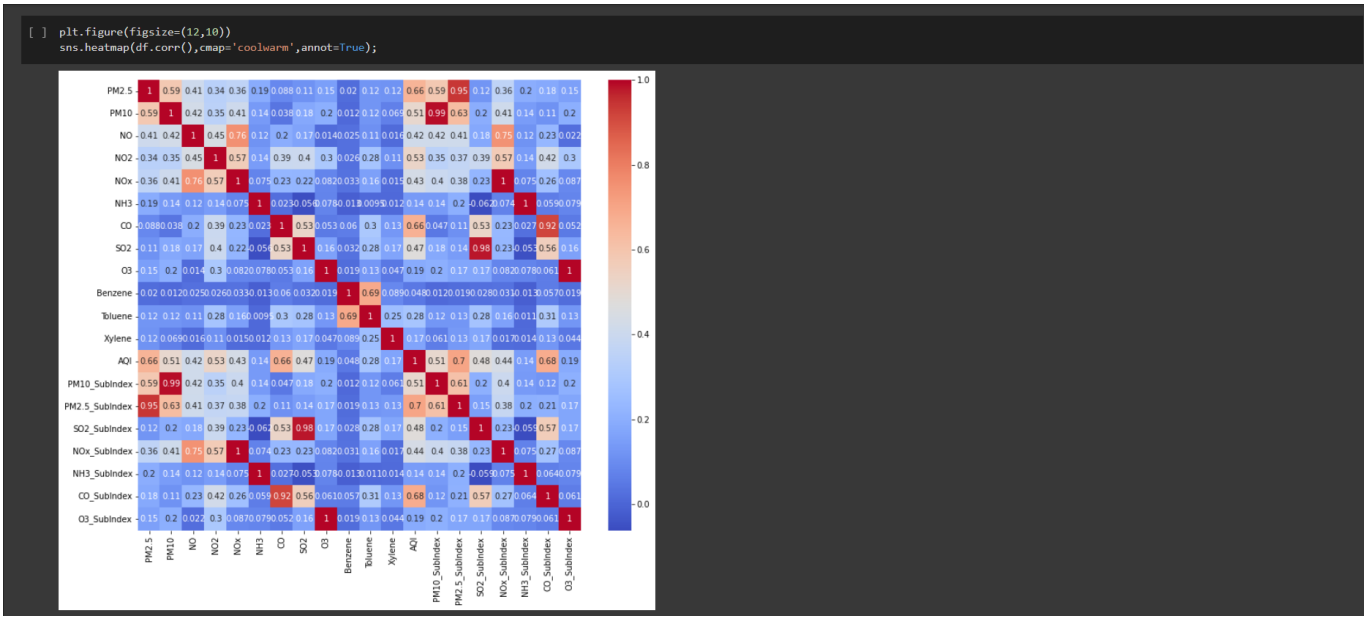


Figure 28: Heatmap for correlation between AQI and AQI Bucket column

Graphs for Distribution of pollutants

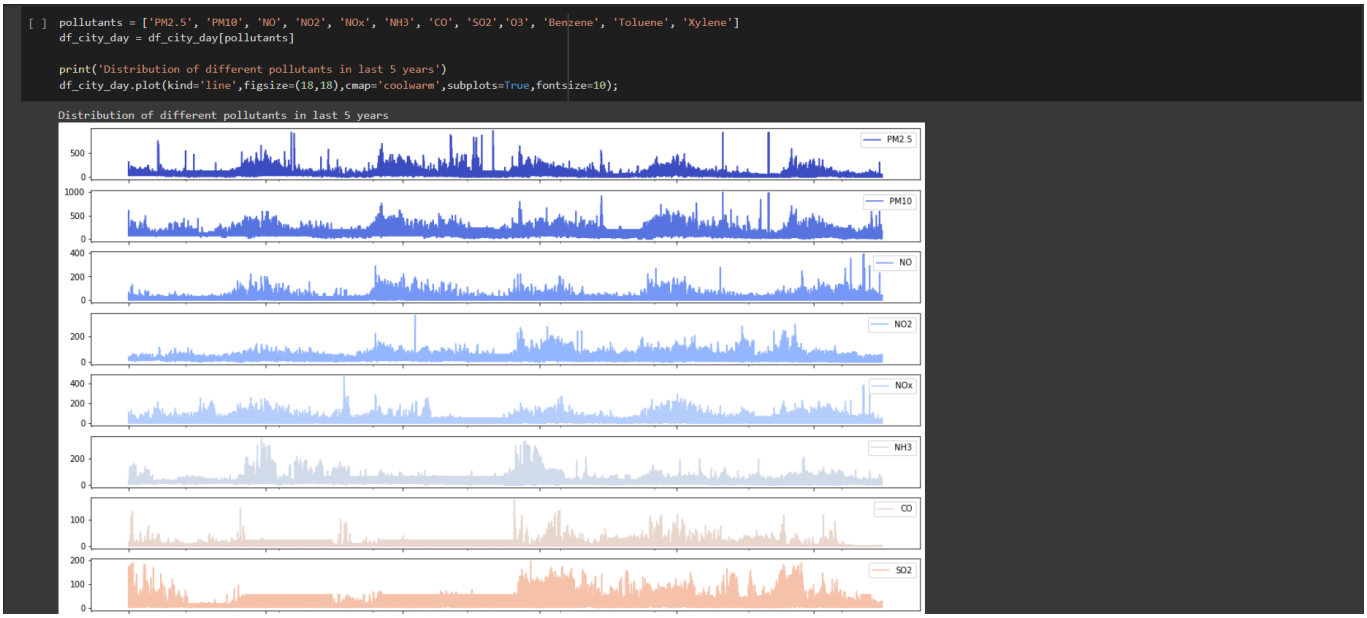


Figure 29: Graphs for Distribution of pollutants

Graphs for Average AQI

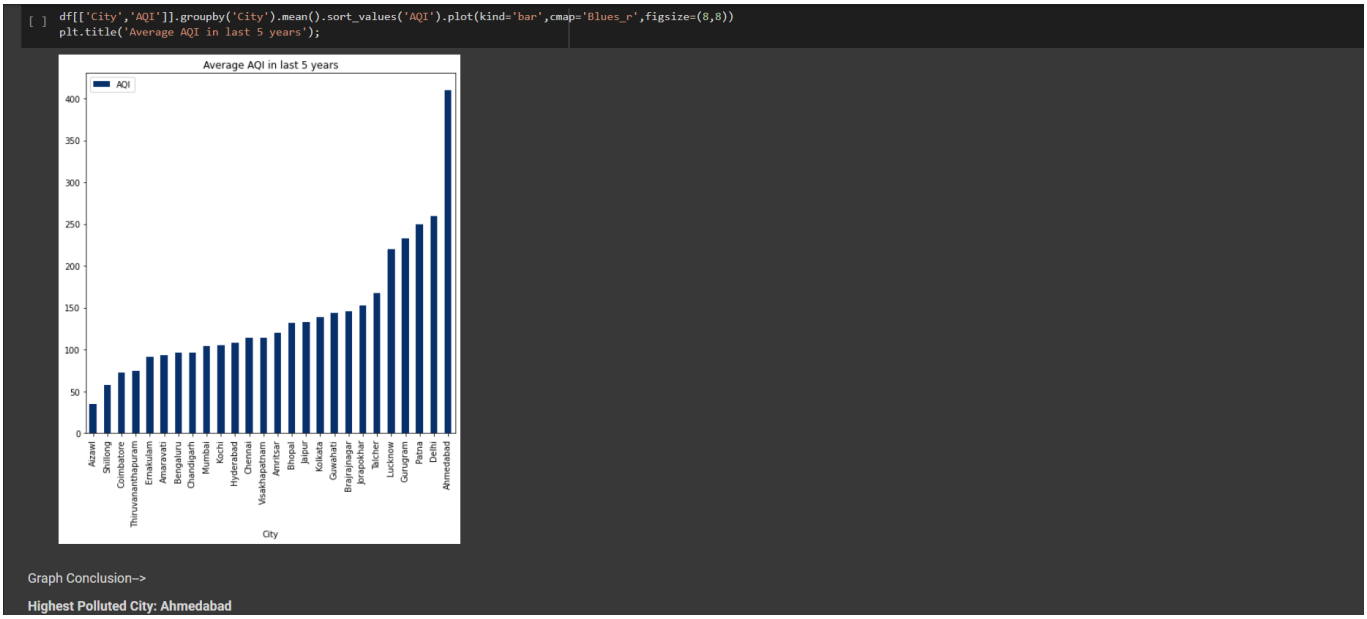
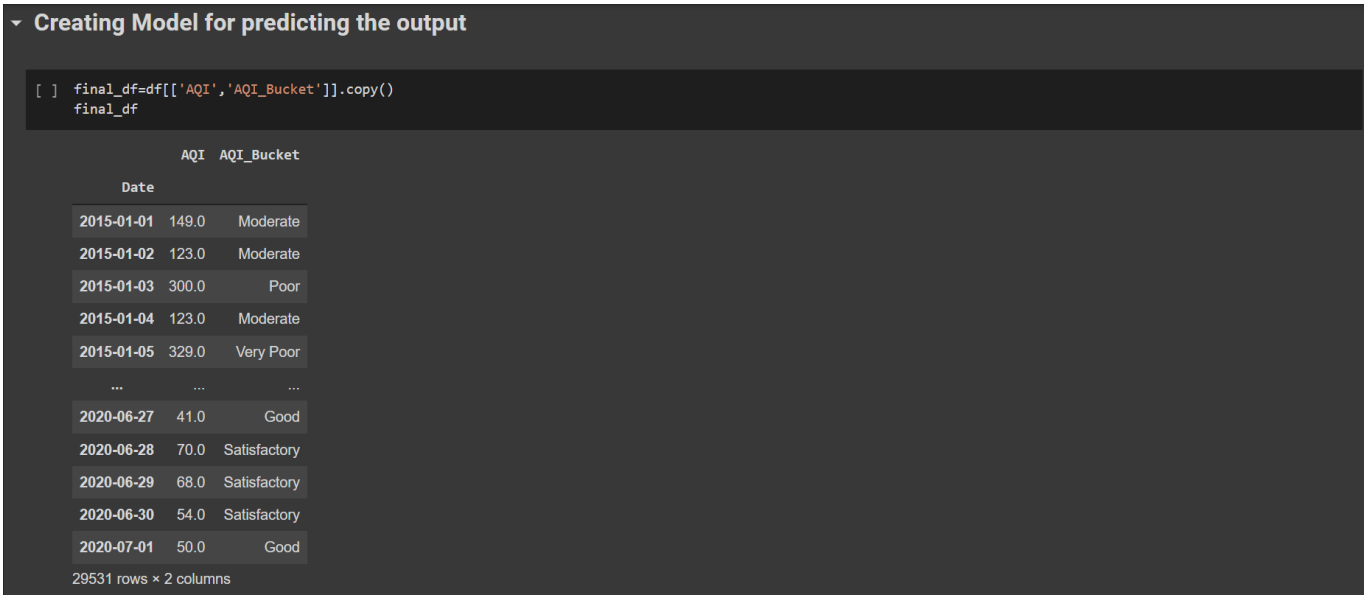


Figure 30: Graphs for Average AQI

Creating Model for prediction



Creating Model for predicting the output

```
[ ] final_df=df[['AQI','AQI_Bucket']].copy()
    final_df
```

Date	AQI	AQI_Bucket
2015-01-01	149.0	Moderate
2015-01-02	123.0	Moderate
2015-01-03	300.0	Poor
2015-01-04	123.0	Moderate
2015-01-05	329.0	Very Poor
...	...	...
2020-06-27	41.0	Good
2020-06-28	70.0	Satisfactory
2020-06-29	68.0	Satisfactory
2020-06-30	54.0	Satisfactory
2020-07-01	50.0	Good

29531 rows × 2 columns

Figure 31: Creating Model for prediction

```
[ ] final_df['AQI_Bucket'].unique()

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

```
#final_df = pd.get_dummies(final_df)
final_df['AQI_Bucket'] = final_df['AQI_Bucket'].map({'Good' :0, 'Satisfactory' :1, 'Moderate' :2, 'Poor' :3, 'Very Poor' :4, 'Severe' :5}).astype(int) #mapping
final_df.head()
```

Date	AQI	AQI_Bucket
2015-01-01	149.0	2
2015-01-02	123.0	2
2015-01-03	300.0	3
2015-01-04	123.0	2
2015-01-05	329.0	4

Figure 32: Creating Model for prediction

Training the model using Random Forest

Testing the model using Random Forest Classifier algorithm with the help of sklearn

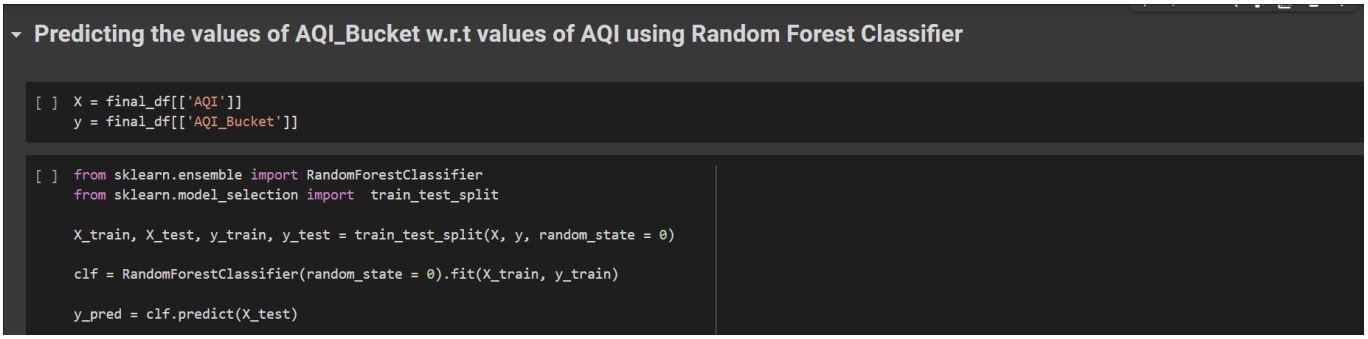


Figure 33: Training the model using Random Forest

User Input Testing

Taking input values from the user to test the model

Figure 34: User Input Testing

Accuracy

Checking the accuracy of the model using Confusion Matrix

```
[ ] from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
    print(accuracy_score(y_test, y_pred))
    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))
```

```
1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       388
           1       1.00      1.00      1.00      2397
           2       1.00      1.00      1.00      2608
           3       1.00      1.00      1.00       805
           4       1.00      1.00      1.00       861
           5       1.00      1.00      1.00       324

    accuracy                           1.00      7383
   macro avg       1.00      1.00      1.00      7383
weighted avg       1.00      1.00      1.00      7383

[[ 388    0    0    0    0    0]
 [   0 2397    0    0    0    0]
 [   0    0 2608    0    0    0]
 [   0    0    0  805    0    0]
 [   0    0    0    0  861    0]
 [   0    0    0    0    0  324]]
```

Figure 35: Accuracy

```
[113] import tensorflow as tf
      import seaborn as sn
      cm = tf.math.confusion_matrix(labels=y_test, predictions=y_pred)

      plt.figure(figsize = (10,7))
      sn.heatmap(cm, annot=True, fmt='d')
      plt.xlabel('predicted')
      plt.ylabel('Truth')
```

```
Text(69.0, 0.5, 'Truth')
```

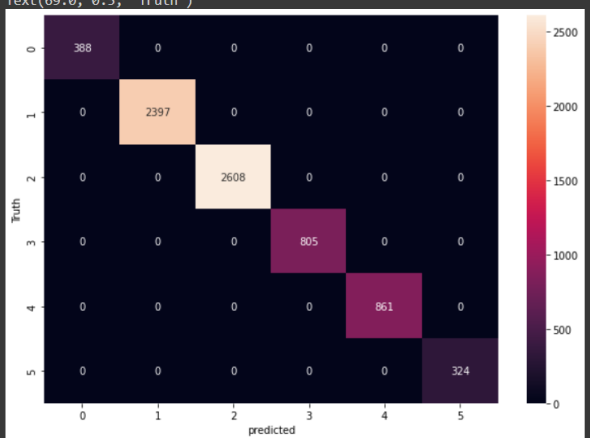


Figure 36: Accuracy

APPLICATIONS

1] In some areas, AQI is used to predict air quality. Future air quality is modelled, using current levels, to produce reasonable predictions for air quality a few days ahead. If you have access to forecast AQI, then you can assess the suitability of your planned activity given the air quality expected on that day.

2] By using outdoor air quality monitoring systems, companies can track the air quality index around their manufacturing units and subsequently control their emission rates.

3] The indoor air quality monitoring system thus helps companies to build a healthier working environment to keep the AQI under control. By comparing the real-time air quality data with ideal conditions, companies can facilitate adequate ventilation, control the production of pollutants in their facility, and keep temperature humidity level in a comfortable range.

4] With the AQI prediction system Governments can monitor the pollution levels in different cities of the country and accordingly make policies and plans to control the risk of air pollution.

5] AQI predictions of heavy traffic and heavy industrial regions can help in creating awareness about the damage happening to the quality of the sir and appropriate actions can be taken to reduce the emission

6] Doctors can also use AQI in predicting when they identify an increase in cases with respiratory distress, as they decipher to be susceptible when the AQI is high.

SCOPE AND FUTURE WORK

AQI is all the more important in developing nations like India where the common man is not quite familiar with the technical terminologies and measuring units (like ppm /ppb / or µg/mg3). Hence the AQI simplifies the understanding of their air quality by decoding the quality in terms of unitless numbers and colour, with each figure and colour representing a different category of associated health risks.The Future work includes:

1] Adding New Features

2] Real time deployment of the proposed model.

3] The historical air quality data ,meteorology data, historical traffic and road status as well as POI distribution information will be collected and the AQI across multiple years can be compared.

4] It can be extended to support online learning so daily data can be used to improve the performance of the air prediction algorithm.

5] An IoT enabled air quality monitoring and visualisation system can be set up to read real time values and use those values for prediction.

CONCLUSION

As we all know we are living in a highly progressive age. Where everything is undergoing massive development. But sadly, the development is at the cost of pollution and environmental damage.

Air pollution is one of the most concerning factors for humanity. We all know the adverse effects of air pollution not only on human health but also on our surroundings. The major step to reduce air pollution is to know the factors affecting the quality of air and the regions where the damage is the most, so that we can prevent the deterioration happening to quality of air.

In this project, we tried to solve this situation with the help of AQI. Air Quality Index(AQI), is used to measure the quality of air. The proposed work is a supervised learning approach using the Random Forest Algorithm. The results show that AQI predictions obtained through RF are promising, which are analysed with results.

By knowing the air quality we can prevent the further damage happening to the environment and our health, rather than trying to find the ways to treat it once the damage is done. As we all know, Prevention is better than cure!!

REFERENCES

[1]Tanisha Madan BPIT(GGSIPU); Shrddha Sagar; Deepali Virmani, Air Quality Prediction using Machine Learning Algorithms –A Review,01 March 2021.[Online].Available : https://ieeexplore.ieee.org/document/9362912

[2] Abdellatif Bekkar, Badr Hssina, Samira Douzi Khadija Douzi ,Air-pollution prediction in smart city, deep learning approach,22 December 2021.[Online].Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00548-1

[3]Radhika M.Patil,Dr. H.T. Dinde,Sonali. K. Powar,A Literature Review on Prediction of Air Quality Index and Forecasting Ambient Air Pollutants using Machine Learning Algorithms,Volume 5, Issue 8, August – 2020 .[Online].Available: https://ijisrt.com/assets/upload/files/IJISRT20AUG683.pdf

[4]Air Pollution: Current and Future challenges on EPA official website.[Online].Available: https://www.epa.gov/clean-air-act-overview/air-pollution-current-and-future-challenges

[5] Miss. Sampada Bharat Deshmukh, Miss. Komal Prakash Shirsat ,Miss. Sakshi Prashant Dhotre, Miss. Pooja Ravsaheb Jejurkar ,A Survey on Machine Learning Prediction of Air Quality Index,Vol-7 Issue-6 2021.[Online].Available: https://ijariie.com/AdminUploadPdf/A$_{S}urvey_{o}n_{M}achine_{L}earning_{B}ased_{P}redictio$ $n_{o}f_{A}ir_{Q}uality_{I}ndex_{i}jariie15775.pdf$

[6]Khalid Nahar,Mohammad Ashraf Ottam,Fayha Alshibli,Mohammed Abu Shquier,Air Quality Index Using Machine Learning: A Jordan Case Study,September 2020,[Online].Available : https://www.researchgate.net/publication/344438674$_{A}IR_{Q}UALITY_{I}NDEX_{U}SING_{M}AC$ $HINE_{L}EARNING\_A_{J}ORDAN_{C}ASE_{S}TUDY$

[7]Sahil Singh, Ayush Yadav and Akhilesh Kumar,Prediction of Air Pollution Using Random Forest,May 27, 2021.[Online].Available: EasyChair Preprint Prediction of Air Pollution Using Random Forest

[8]Mauro Castelli,Fabiana Martins Clemente,Ales Popovic,Sara Silva and Leonardo Vanneschi,Guest Editor : Felix Chan, A Machine Learning Approach to Predict Air Quality in California,4 August

2020.[Online].Available:
https://www.hindawi.com/journals/complexity/2020/8049504/

[9]Andrew Knox,Natalia Mykhaylova,Greg J Evans,Colin J.Lee,The expanding scope of air pollution monitoring can facilitate sustainable development,March 2013.[Online].Available:
https://www.researchgate.net/publication/232064889$_{T}he_{e}xpanding_{s}cope_{o}f_{a}ir_{p}ollution_{m}onitoring_{c}an_{f}acilitate_{s}ustainable_{d}evelopmen$