AIR QUALITY INDEX AUTOMATION, STATISTICAL ANALYSIS & FORECASTING

Submitted in partial fulfillment of the requirements for the degree of

Master of Science
In
Data Science

by

Palak Goel
21MDT0041

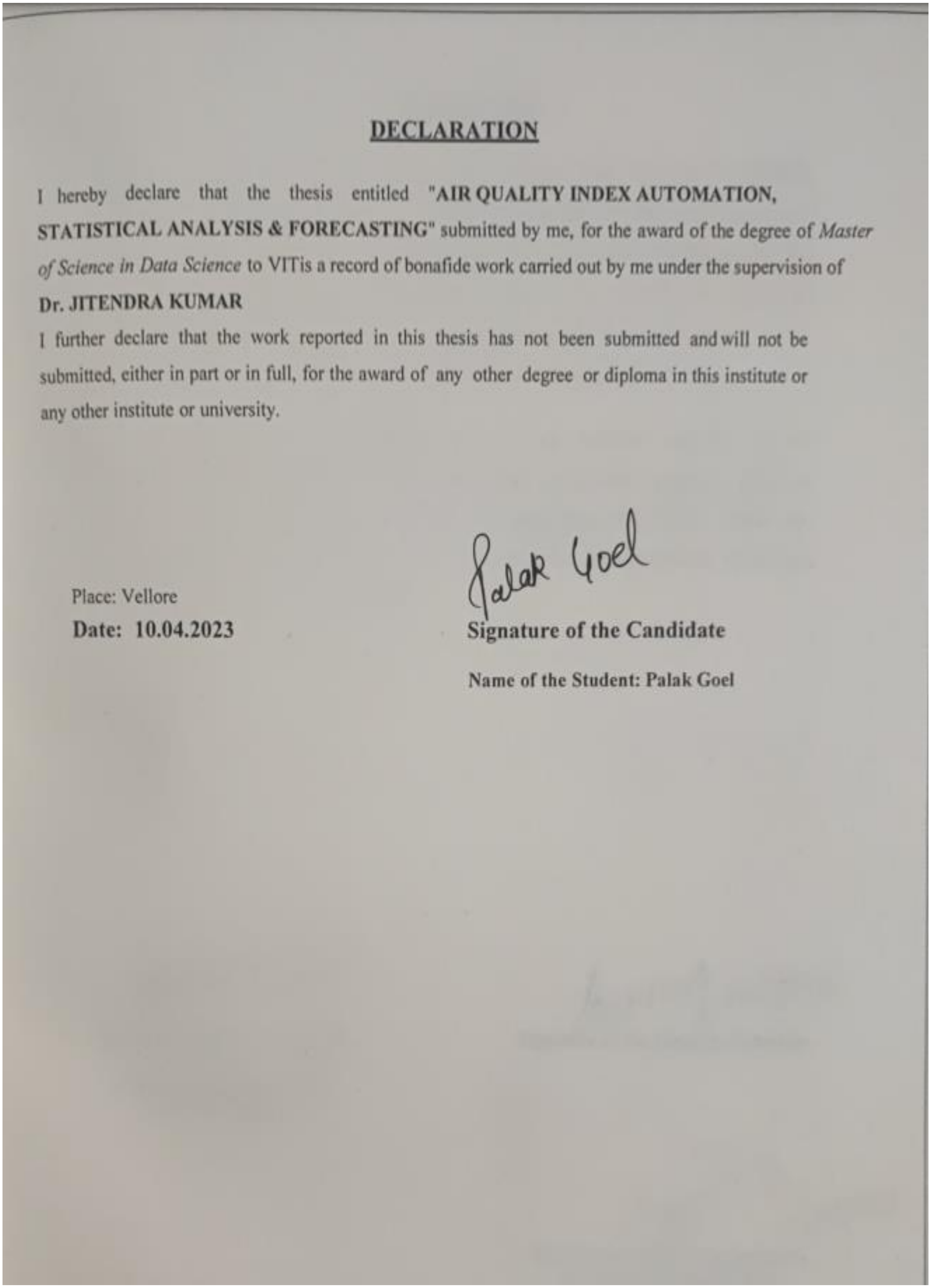
Under the guidance of

Dr. Jitendra Kumar

School of Advanced Sciences
VIT, Vellore.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

APRIL, 2023



DECLARATION

I hereby declare that the thesis entitled **"AIR QUALITY INDEX AUTOMATION, STATISTICAL ANALYSIS & FORECASTING"** submitted by me, for the award of the degree of *Master of Science in Data Science* to VITis a record of bonafide work carried out by me under the supervision of **Dr. JITENDRA KUMAR**

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

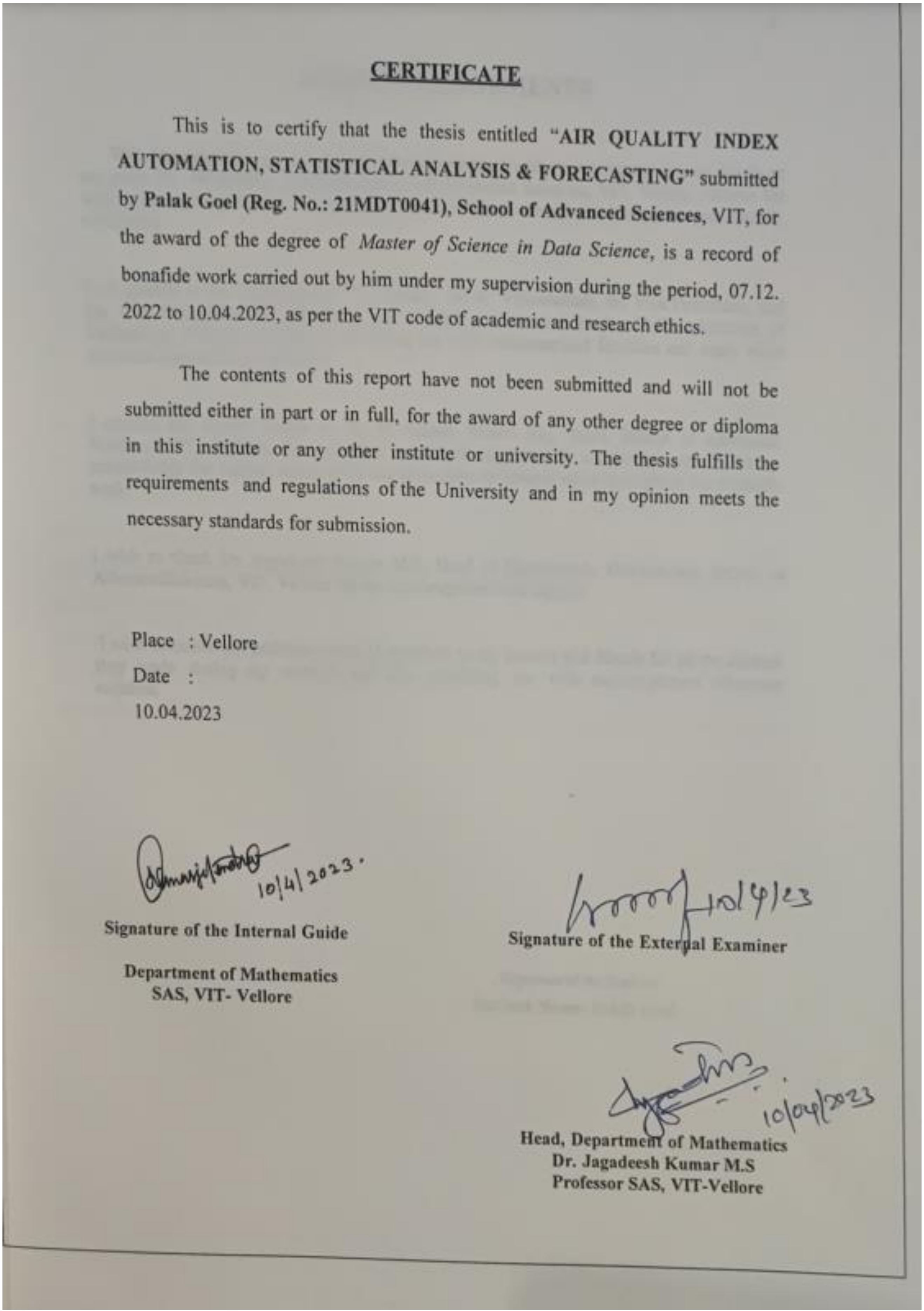
Palak Goel

Place: Vellore

Date: 10.04.2023

Signature of the Candidate

Name of the Student: Palak Goel



CERTIFICATE

This is to certify that the thesis entitled **"AIR QUALITY INDEX AUTOMATION, STATISTICAL ANALYSIS & FORECASTING"** submitted by **Palak Goel (Reg. No.: 21MDT0041), School of Advanced Sciences**, VIT, for the award of the degree of *Master of Science in Data Science*, is a record of bonafide work carried out by him under my supervision during the period, 07.12. 2022 to 10.04.2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore
Date :
10.04.2023

10/4/2023.

Signature of the Internal Guide

Department of Mathematics
SAS, VIT- Vellore

10/4/23

Signature of the External Examiner

10/04/2023

Head, Department of Mathematics
Dr. Jagadeesh Kumar M.S
Professor SAS, VIT-Vellore

ACKNOWLEDGEMENTS

With immense pleasure and deep sense of gratitude, I wish to express my sincere thanks to my guide **Dr. JITENDRA KUMAR,** School of Advanced Sciences, VIT, Vellore without his motivation and continuous encouragement, this research would not have been successfully completed.

I am grateful to the Chancellor of VIT, Vellore, Dr. G. Viswanathan, the Vice Presidents, and the Vice Chancellor for motivating me to carry out research in the Vellore Institute of Technology, Vellore and also for providing me with infrastructural facilities and many other resources needed for my research.

I express my sincere thanks to Dr. N Arunai Nambi Raj, Dean, School of Advanced Sciences, VIT, Vellore, for her kind words of support and encouragement. I like to acknowledge the support rendered by my classmates in several ways throughout my researchwork.

I wish to thank Dr. Jagadeesh Kumar M.S, Head of Department, Mathematics, School of AdvancedSciences, VIT, Vellore for his encouragement and support.

I wish to extend my profound sense of gratitude to my parents and friends for all the support they made during my research and also providing me with encouragement whenever required.

Signature of the Student

Student Name: Palak Goel

ABSTRACT

India ranks sixth in the world for pollution, and the rapid increase in pollution is having a negative influence on the environment in several large cities. In this regard, the Central Pollution Control Board's data were used to produce the polluted database. (India). These figures demonstrate the annual increases in SO2, NOx, particulate matter (PM) 2.5 and PM 10 for the year 2021. They were collected from several monitoring stations in Indian cities. In order to gain a comprehensive understanding of the harm caused by air pollution, this project will track trends in air pollution and assess the air quality index in relation to various geographic locations. One instrument for determining the current status of air quality is the air quality index. The Air Quality Index (AQI), which is based on the synergistic effects of the four pollutants PM10, PM2.5, SO2, and NO2, is calculated using the sub-index approach. According to the results, the sites under examination fell into four categories when utilised as a trained dataset for machine learning algorithms: severely polluted (CP), highly polluted (HP), moderately polluted (MP), and low polluted. Finding the cluster or hotspot of pollution particle matter using satTScan Additionally, Excel is used to carry out simulation approaches, which are then further confirmed by determining the root mean square error.

CONTENTS **Page No.**

List of Figures

List of Tables

List of Abbreviations

AQI	Air Quality Index
ANN	Artificial Neural Network
MLR	Multiple Linear Regression
CPCB	Central Pollution Control Board's
SO2	Sulphur Dioxide
NO2	Nitrogen Dioxide
PM2.5	Particulate Matter 2.5
PM10	Particulate Matter 10
RSPM	Respirable Suspended Particulate Matter
SPM	Suspended Particulate Matter
EPA	Environment Protection Agency

1. INTRODUCTION

With a population of more than 1,3 billion, India is the second most populated nation in the globe. Sadly, this big population comes at a cost, as India confronts serious air pollution issues, which have become an increasing concern in recent years.India has a severe problem with air pollution, which is caused by a number of factors. Industrial operations, construction, crop residue burning, and vehicle emissions are among the leading contributors of air pollution in the United States. According to the World Health Organization (WHO), fourteen of the twenty most polluted cities in the world are located in India, with Delhi being the most polluted capital in the world.

The Air Quality Index (AQI) is a numerical index used to reflect the air quality in a particular location. It is one method for measuring air pollution levels. Air pollutants such as particulate matter (PM2.5 and PM10), nitrogen dioxide (NO2), sulphur dioxide (SO2), and ozone are used to calculate the AQI. (O3). Understanding the pollutants briefly . Nitrogen dioxide (NO2) is primarily released during the burning of fuels used in transportation or power plants.Sulphur dioxide, or SO2, is primarily released when coal, gasoline, and sulfuric acid are burned.The deadliest type of air pollution is known as suspended particle matter, or spm. They are discovered suspended in the atmosphere of the earth and are microscopic in size.Respirable suspended particle matter (rspm). respiratory illnesses are caused by a subtype of spm. PM10: Because it has a diameter of less than 10 micrometres and may be more hazardous than PM2.5, it is categorised as spm.Particulate matter in suspension with a width of less than 2.5 micrometres is known as pm2_5. They frequently hang suspended for extended periods of time and may be very harmful.The AQI ranges from 0 to 500, with higher readings indicating greater air pollution.The current state of air pollution and AQI in India is really frightening. As of March 2023, the Central Pollution Control Board (CPCB) reports that the average AQI in India's major cities remains in the unhealthy to hazardous level. This comprises Delhi with an average AQI between 300 and 400 and Mumbai with an average AQI between 250 and 400.High AQI levels offer considerable health risks to the population, especially to vulnerable groups such as children, the elderly, and those with pre-existing diseases. Asthma, bronchitis, and lung cancer can be caused by exposure to excessive amounts of air pollution. Air pollution has also been connected to cardiovascular disorders such as stroke and heart

attack. In 2019, the National Clean Air Programme (NCAP) will be implemented as part of the Indian government's efforts to combat air pollution. The NCAP is a comprehensive plan to reduce air
pollution by 20-30% nationwide by 2024. The plan includes adoption of cleaner technology, promotion of public transportation, and reduction of industrial emissions, among other initiatives. However, despite these initiatives, air pollution in India remains a significant problem, and more has to be done to address it. This study seeks to identify the top classification and machine learning prediction models, such as logistic regression and k nearest neighbor, as well as regression and decision tree models.

1.1 OBJECTIVE

- AQI Automation.
- Studying which particulate matter (PM10, PM2.5, SO2, and NO2) effecting the air quality by correlation analysis
- Addressing the issue of multicollinearity
- Identifying most polluted state and air particulate prevailing in that region.
- Identifying state with good air quality in order to applying its practices for index improvement.
- Anova analysis of model
- Visualization over India map.
- Hotspot detection for individual air particulate matter
- Predicting air quality on the basis of PM10, PM2.5, SO2, and NO2 values.
- Model validation by using simulation technique on excel

1.2 BACKGROUND

Air pollution is a significant issue in India, one of the most polluted nations in the world. Thirteen of the world's twenty most polluted cities are located in India, according to the World Health Organization (WHO). In India, burning crop wastes, industrial emissions, and vehicle emissions are the main contributors of air pollution.

According to a review of the literature on air quality index analysis of Indian states, there are large differences in air quality between the various Indian states. The air quality improved during the 2020 COVID shutdown, when all of India was closed, leading to a GOOD air quality index. But how quickly is it expanding in different parts of India after the lockdown?

Delhi is one of the world's most polluted cities, and air pollution is a significant public health issue. In a study by Singh et al. carried out in (2020), particulate matter is the main pollutant in Delhi and the AQI is over permissible levels for the majority of the year. According to the report, urgent action is required to lower Delhi's air pollution.

Air pollution is a big issue in Maharashtra, one of India's most industrialized states. In Maharashtra, particulate matter is the main pollutant, and the air quality index (AQI) is over permissible levels for the majority of the year, In a study by More et al. carried out in (2018). According to the report, Maharashtra's air pollution can be decreased by enacting stronger laws. One of the most populous states in India is Uttar Pradesh, where air pollution is a big problem. In a study by Srivastava et al. carried out in (2021), particulate matter is the main pollutant in Uttar Pradesh and the AQI there is over permissible levels for the most of the year. According to the research, Uttar Pradesh has to take immediate action to minimize air pollution.

One of India's least polluted states, Tamil Nadu has typically acceptable air quality. According to a study by Ravindra et al. (2018), particulate matter is the main pollutant in Tamil Nadu and the AQI is below safe levels for the majority of the year. According to the report, Tamil Nadu's ability to maintain acceptable air quality is due to its efficient rules and regulations.

2. PROJECT DESCRIPTION

Air pollution is a major environmental challenge in India, with many cities regularly exceeding safe levels of air quality. The detrimental impact of poor air quality on public health, agriculture, and the environment is well documented. In recent years, there has been an increased focus on developing and implementing solutions to address this issue. Machine learning algorithms have emerged as a powerful tool for predicting air quality and providing insights into the contributing factors.

This project aims to develop a machine learning-based model for predicting air quality in India using a dataset from the year 2021. The dataset includes various parameters such as PM2.5, PM10, NO2, SO2 The project will explore different machine learning algorithms such as linear regression, random forest, and k nearest neighbor to determine which approach produces the most accurate predictions. The data is preprocessed and transformed using various techniques such as scaling, normalization, and feature selection. The processed data is then fed into the model, which is capable of capturing temporal dependencies in the data, and predicting air quality values for the next hour, day, or week.

The study focuses on 10 major Indian cities, including Delhi, Mumbai, Bangalore, Kolkata, Chennai, Hyderabad, Pune, Ahmedabad, Jaipur, and Lucknow. These cities were selected based on their high levels of pollution and population density. The proposed model has the potential to provide valuable insights to policymakers and stakeholders in identifying the factors contributing to poor air quality and developing effective interventions to mitigate the issue. Additionally, the model can assist in predicting future air quality levels, enabling timely actions to be taken to prevent exposure to harmful pollutants.

Overall, this project can contribute to a better understanding of air quality in India and support efforts to improve public health and environmental conditions.

3. TECHNICAL SPECIFICATIONS

Hardware:

- Laptop/Computer
- 4 GB Ram min.
- 2 GB Graphics min.

Software:

- Anaconda Navigator (v3.9)
- Jupyter Notebook (v5.2)
- SaTScan (v10.1)
- Excel

Framework:

- NumPy- for calculations
- Pandas- (Series, Dataframe) for handling the data,
- Matplotlib pyplt; seaborn for Visualization
- Seaborn – for visualzation
- SkLearn- preprocessing, sklearn.metrics, and sklearn.model_selection packages must be installed for model training and finding the accuracy

4. LITERATURE SURVEY

Gopalakrishnan (2021) used machine learning to forecast air quality at several locations in Oakland, California using data from Google Street View. He concentrated on the locations where the data were missing. In order to forecast air quality for every area in city neighborhoods, the author created a web application.

Sanjeev (2021) investigated a dataset containing information on both meteorological and contaminant concentrations. According to the author, the Random Forest (RF) classifer fared the best since it is less prone to over-ftting when analysing and forecasting the air quality.

Castelli et al. (2020) used the Support Vector Regression (SVR) ML method to attempt to forecast Californian air quality in terms of contaminants and particle levels.

The authors claimed to have created a brand-new technique for simulating hourly atmospheric pollution.

Dorswamy et al. (2020) looked on machine learning predictive models for predicting PM concentration in the air. The scientists examined data from six years of Taiwanese air quality monitoring and used pre-existing models. They asserted that the actual values and projected values were fairly similar.

Liang et al. (2020) Using 11 years of data, Liang et al. (2020) examined the performances of six ML classifiers to forecast Taiwan's AQI. The best methods for predicting air quality, according to the authors, are Adaptive Boosting (AdaBoost) and Stacking Ensemble, but forecasting accuracy varies by region.

Madan et al. (2020) analysed the performances of 20 different literary works with respect to the contaminants they studied and the machine learning techniques they used. The authors discovered that numerous studies used weather information, such as temperature, wind speed, and humidity, to more precisely estimate pollution levels. They discovered that the boosting and Neural Network (NN) models outperformed the other well-known ML techniques.

Anikender Kumar and Pramila Goyal (2011) published a paper that uses a concept of Principal Component Regression (PCR) and one main machine learning algorithm that is Multiple Linear Regression Algorithm to predict the AQI value for Delhi on daily basis, India, based on historical AQI data and climatic indicators. Using historical data from the years 2000 to 2005 and various algorithms, they anticipate the daily AQI for the year 2006. Then, using the Multiple Linear Regression Technique

[1], this predicted value was compared to the observed value of the AQI for the seasons of summer, monsoon, post monsoon, and winter for the year 2006. Finding collinearity between independent variables is done using principal component analysis. In order to decrease the number of predictors and to get rid of multicollinearity among the predictor variables, the principal component analysis were used for feature selection in multiple linear regression [1]. In comparison to other seasons, the winter season shows the best performance for the Principal Component Regression in AQI prediction. The ambient air contaminants that may have harmful effects on health were not taken into account while projecting the future AQI in this study; only meteorological parameters were employed. When it comes to predicting the AQI during the winter, the Principal Component Regression performs better in comparison of any other seasons. In this study, only meteorological parameters were used to forecast the future AQI, excluding ambient air pollutants that can be detrimental to health.

Huixiang Liu and colleagues 2019:- Huixiang Liu (et al. 2019) selected Beijing and an Italian city as the subject cities for their study. They have forecasted the Air Quality Index (AQI) for the city of Beijing and also focused on the concentration of NOx in an Italian city using two datasets that are available publicaly and are completely different.The Beijing Municipal Environmental Centre [5] has made available the initial dataset, which includes 1738 public incidents, spanning the time period from 2013 in the month December to August 2018. This dataset contains information on Beijing's PM2.5, PM10, SO2, O3and NO2 concentrations as well as the hourly averaged AQI. The second dataset, which includes 9358 cases, was created using information gathered between March 2004 and February 2005 in Italian cities. Hourly averaged CO, NOx, and NO2 concentrations as well as non-methane hydrocarbons like benzene are elements of this dataset.However, they mainly focused on it because NO2 prediction is one of the important predictors for evaluating air quality. They employed Support Vector machine (SVM) and Random Forest (RF) techniques to estimate AQI and NOx concentrations. RFR outperforms SVR at forecasting AQI, whereas SVR excels at forecasting NOx concentration.

Ziyue Guan and Richard O. Sinnot (2018) In this paper they have used a range of machine learning algorithms to predict the PM2.5 concentration. Data were acquired via Airbeam, a mobile device designed to monitor PM2.5 levels, as well as from the Environment Protection

Agency's (EPA) official website for Melbourne, which provides information regarding PM2.5 air characteristics [8]. As machine learning techniques for the PM2.5 prediction, Deep learning models -Artificial Neural Networks (ANN), Long Short Term Memory (LSTM) as recurrent neural networks and machine learning algorithm Linear Regressions (LR) were used. However, LSTM performed the best and accurately forecasted high PM2.5 values.

HeidarMaleki (et al. 2019): For the stations Naderi, Havashenasi, MohiteZist, and Behdasht in Ahvaz, Iran, the most polluted city in the world, HeidarMaleki (et al. 2019) forcasted the hourly concentration values for the ambient air pollutants SO2, CO, PM10, NO2, PM2.5, and O3. They calculated and forecasted the Air Quality Index (AQI) and also evaluated the Air Quality Health Index at the four air quality monitoring stations in Ahvaz (AQHI). They used an Artificial Neural Network (ANN) machine learning technique to predict the hourly concentration of air pollutants and the two air quality indices, AQI and AQHI, from August 2009 to August 2010. Time, date, weather variables, air pollution levels, and time are some of the inputs used by ANN algorithms.

Aditya C. R. (et al.) (2018):- Based on a dataset that includes atmospheric conditions in a particular city, Aditya C. R. (et al.) (2018) employed computer algorithms to identify and predict the PM2.5 concentration level. Additionally, they predicted the day's PM2.5 concentration [10]. They first classify the air as being contaminated or not polluted using the Logistic Regression approach, and then they utilise the Auto Regression algorithm to predict the future value of PM2.5 based on historical data.

Nidhi Sharma (et al. 2018):- Nidhi Sharma (et al. 2018) completed a comprehensive analysis of air pollution data from 2009 to 2017 and provided a critical evaluation of the trend in air pollution from 2016 to 2017 in Delhi, India [14]. They have predicted future trends for a number of pollutants, including Sulfur Dioxide (SO2), Nitrogen Dioxide (NO2), Carbon Monoxide (CO), Suspended Particulate Matter (PM), Ozone (O3) and Benzene.With the use of data analytics time series regression forecasting based on historical data, the future values of the pollutants previously stated have been predicted. According to the findings of this investigation, the monitoring stations at AnandVihar and Shadipur in Delhi are being evaluated. The results show that Delhi's PM2.5 and NO2 levels have both increased, and that PM10 concentration levels have drastically increased [14]. While NO2 concentration is projected to increase by 16.77 mg/m3, ozone is projected to increase by 6.11 mg/m3, benzene is projected to drop by 1.33 mg/m3, and SO2 is projected to increase by 1.24 mg/m3.Using the

WEKA program, Mohamed Shakir and N.Rakesh (2018) calculated the effects of environmental variables on the aforementioned air pollutants, including temperature, wind speed, and humidity. They also looked at by how much amount the various air pollutants (NO, PM10, CO, NO2, and SO2) were present at different times of day and on different days of the week. Data from the Karnataka pollution control board was collected. By applying the ZeroR algorithm in the WEKA tool, the study discovered that the concentration levels of air pollutants increase throughout working days, notably during the busiest parts of the day, and decrease during weekends or holidays [15]. The study uses

straightforward K-means clustering techniques to show the relationships or correlations between environmental factors like temperature, wind speed, and humidity and air pollutants like NO, NO2, PM10, CO, and SO2.

Kazem Naddaf (et al., 2012): KazemNaddaf (et al., 2012) used the AirQ programme suggested by the World Health Organization to analyse the effects of PM10, SO2, NO2, and O3 on human health in Tehran metropolis, Iran, the nation's most populous metropolis [16]. The health implications included all-cause mortality, cardiovascular disease, and respiratory disease. According to the study's findings, PM10 caused an excess of total mortality of 2194 deaths out of 47284 in a year and had the worst health consequences on Tehran City's population of 8,700,000 [16]. There have been too many deaths caused by SO2, NO2, and ozone, with 1458, 1050, and 819 respectively. These findings show that air pollution in Tehran was a significant problem, and it is imperative that Tehran reduce its detrimental health impacts.

Yusef Omidi Khaniabadi (et al. 2016): The main objective of this study is to ascertain whether there is a correlation or association between health effects, such as mortality rates from cardiovascular diseases, and air pollutants like PM10, NO2, and O3 for the Iranian city of Kermanshah between the years of 2014 and 2015. They used the AirQ software recommended by WHO for this.There are 188 occurrences of premature mortality due to cardiovascular diseases, 33 of which are linked to NO2 and 83 to O3[17].The results of the study show that for every 10/m3 increase in PM10, NO2, and O3 concentration levels, the mortality risk will rise by 1.066, 1.012, and 1.020 respectively.

S.TikheShruti (et al. 2013) used two soft computing techniques, artificial neural network (ANN), and genetic programming, to forecast future concentration levels of air pollutants such as oxides of nitrogen (NOx), oxides of sulphur (SOx), and respirable suspended particulate matter (RSPM) over the years 2005 to 2011. (GP). The city in Maharashtra, Pune is listed as having the second-highest pollution levels in all of India. They have created a total of six models using hourly average data values of pollutants concentration spanning more than 7 years. (three of each algorithm ANN and GP). GP methods outperform ANN among these two techniques.

ArchontoulaChaloulakou (et al., 2003): This study employed Artificial Neural Network (ANN) and Multiple Linear Regression (MLR) techniques to predict the PM10 concentration over a two-year period for the city of Athens, Greece. The dataset was divided into three unequal subsets before applying input to ANN, since the training dataset only contains two thirds of the records or instances that are available, and the remaining records or cases were equally divided into the validation and test set [20]. The results of this study's comparison of ANN and MLR demonstrate that ANN outperforms MLR. This study asserts that an ANN will deliver the necessary prediction answers or outcomes if it is properly trained.

5. METHODOLOGY

AQI Formulation: AQI stands for Air Quality Index, which is a numerical scale used to report how polluted the air is in a particular location. The AQI takes into account several pollutants that are commonly found in the air, such as particulate matter, ozone, sulphur dioxide, nitrogen dioxide, and carbon monoxide.

The AQI scale ranges from 0 to 500, with higher numbers indicating more polluted air. The AQI is typically reported in real-time by government agencies and other organizations that monitor air quality. An AQI reading of 0 to 50 is considered good air quality, while an AQI reading of 51 to 100 is considered moderate air quality. An AQI reading of 101 to 150 is considered unhealthy for sensitive groups, such as people with respiratory problems. An AQI reading of 151 to 200 is considered unhealthy for everyone, and AQI readings above 200 are considered very unhealthy and can be dangerous to people's health. The AQI is a useful tool for people to determine the quality of the air they are breathing and take appropriate precautions to protect their health.

Derivation for Individual Pollutant Index and AQI The AQI is an index for reporting daily air quality. It informs you of the cleanliness and pollution levels of your breath as well as any potential health risks. The AQI concentrates on potential health impacts that may occur hours or days after breathing polluted air. EPA calculates the AQI for five major air pollutants regulated by the Clean Air Act: ground level ozone, particle pollution Air quality directly affects (also known as particulate our quality of life. matter), sulfur dioxide, carbon monoxide, and nitrogen dioxide. To safeguard the public's health, the EPA has set national air quality standards for each of these pollutants. AQI is calculated on a scale from 0 to 500, and the numbers are scaled using the following formula:

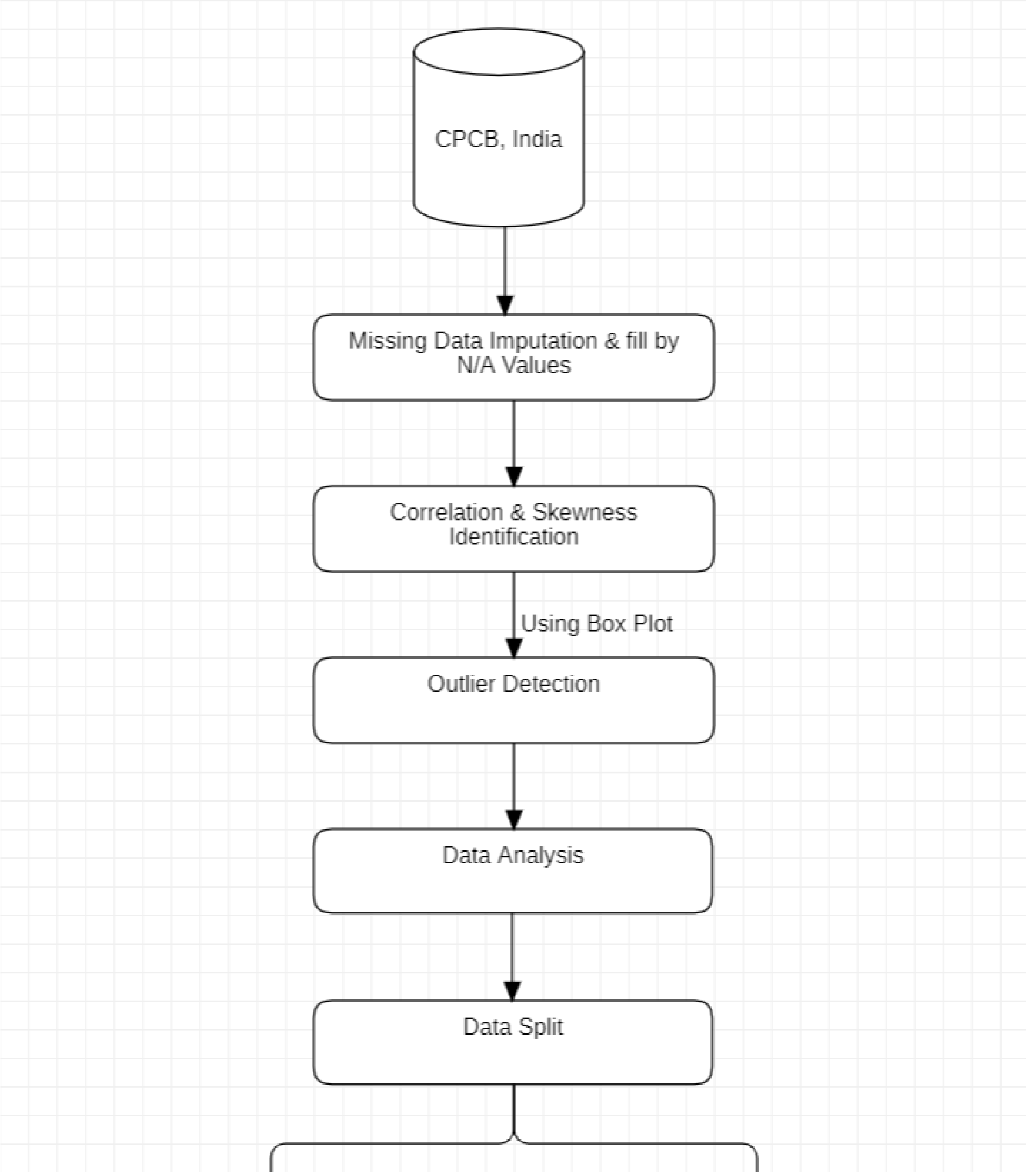
$$AQI = AQI_{min} + \frac{APM_{Ob} - APM_{Min}}{AQI_{Max} - AQI_{Min}} (APM_{Max} - APM_{Min)}$$

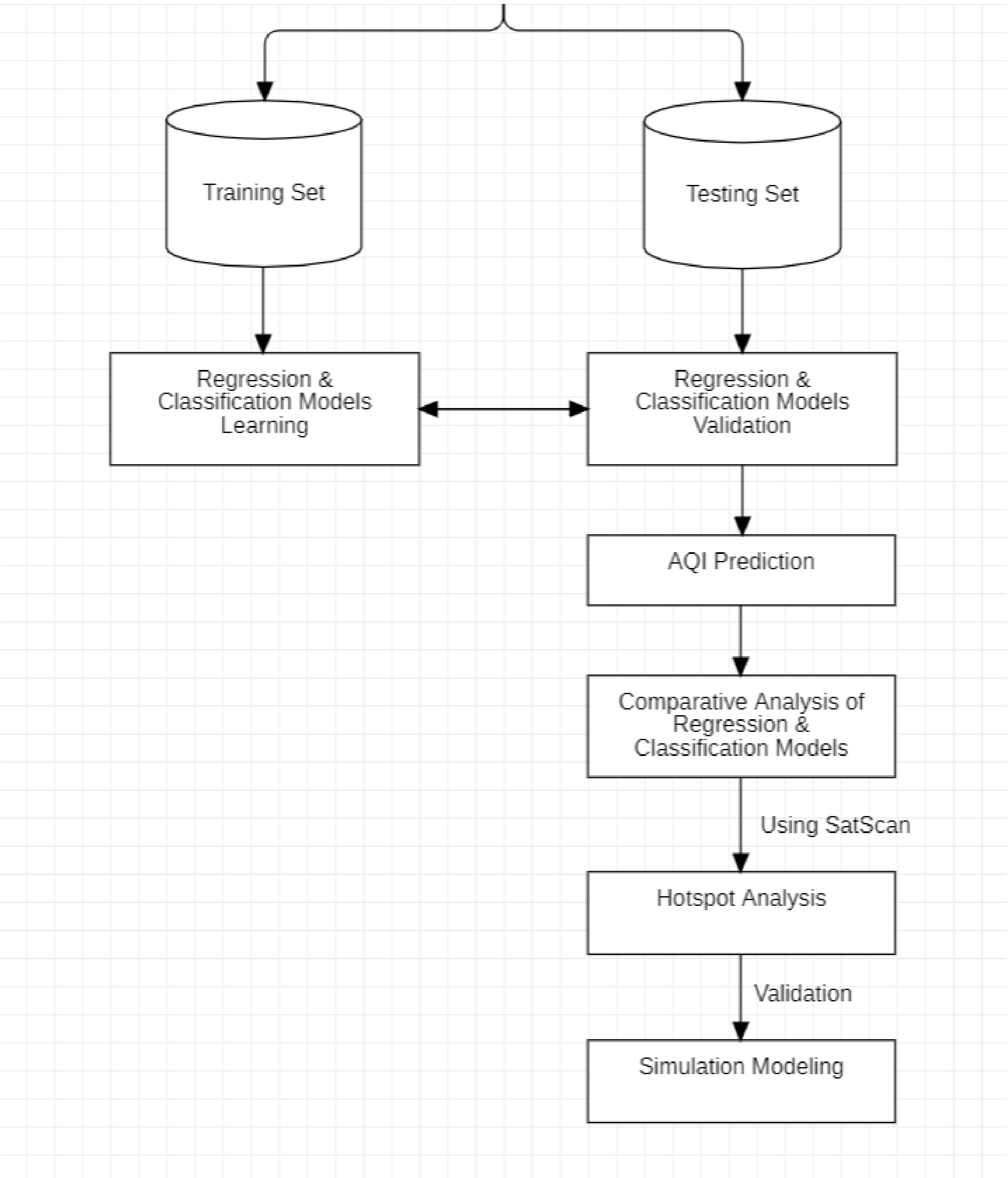
The first and most crucial need for effective visualisation and the development of effective ML models is data quality. The pre-processing procedures aid in decreasing the noise in the data, which ultimately speeds up processing and expands the applicability of ML algorithms. The two most frequent mistakes in data extraction and monitoring applications are outliers and missing data. The data preparation step involves modifying or deleting outlier data, filling out data that is not a number (NAN), and performing other operations on data.

List of operations done on data :-

- Correlation analysis
- Missing Value Treatment
- Statistical and descriptive analysis
- Outlier detection
- Feature selection
- Machine Learning Algorithm:
  - Multiple Linear Regression:
  - Logistic Regression:
  - K Nearest Neighbour:
  - Decision Tree:
  - Random forest
  - SaTScan for clustering and hotspot analysis
  - Error metrics
  - Performance metrics

5.1 MODEL BUILDING STEPS


CPCB, India
Missing Data Imputation & fill by N/A Values
Correlation & Skewness Identification
Using Box Plot
Outlier Detection
Data Analysis
Data Split


Training Set
Testing Set
Regression & Classification Models Learning
Regression & Classification Models Validation
AQI Prediction
Comparative Analysis of Regression & Classification Models
Using SatScan
Hotspot Analysis
Validation
Simulation Modeling

5.2 Data preprocessing:

Categorical conversion

Classification Conversion Our analysis necessitates the conversion of at least one independent variable into numeric data, which must be either a multi-class categorical variable or a binary categorical variable. For the same, we'll employ cat coding and one hot decoding. Cat coding, which essentially gives numbers for ordinal data, transforms categorical data into numeric form for use. Using cat coding, we can map out our sortable categories, such as ancient, new, renovated, 0, 1, and 2.

One-hot encoding (binary values from categorical data) Categorical factors are represented as binary vectors in one-hot encoding. In order to do this, the categorical numbers must first be converted to integer values. The index of the integer, which is denoted with a 1, is then used to symbolise each integer value as a binary vector with all other values being zero.

State is a multiclass variable in our dataset, whereas type is a binary category variable. So, ultimately, we are changing them. We will only use cat coding in the study that follows to convert our ordinal data. We have primarily used cat coding because it alters the target column itself, whereas one hot encoding creates new columns based on the types that the column contains, which are represented by the binary numbers 1 and 0. Consequently, the data collection becomes much more complex and redundant. Both approaches can be used to convert categories to numbers, but we favour cat coding over one hot encoding.

Missing value Treatment

Since we already know that our dataset contains missing values , and we need to fill them for our further analysis . We will be using Imputation to fill in our missing values. Imputation is the process of replacing missing data with substituted values . Because missing data can create problems for analyzing data, imputation is seen as a way to avoid pitfalls involved with listwise deletion of cases that have missing values.

Correlational Analysis

A statistical method called correlation analysis is used to look at the direction and strength of the connection between two or more variables. It is a helpful instrument for determining the strength of the correlation between changes in one variable and changes in another variable.

The statistical metric used to express the strength of correlation between two factors is the correlation coefficient. Its value falls between -1 and 1, with -1 denoting a perfect negative correlation, 0 denoting no correlation, and 1 denoting a perfect positive correlation. A correlation coefficient of 0 indicates a lack of a linear relationship rather than an absence of any connection at all between the variables.

In order to make forecasts about the course of events, correlation analysis can be used to determine relationships between variables. In order to find the main causes of issues and create effective remedies, it can also be used to pinpoint variables that might be influencing changes in other variables. It's crucial to remember that a connection does not necessarily indicate a cause. Two factors are not necessarily caused by each other just because they are correlated. Both of the variables could be changing as a result of other variables or factors, or the connection between them could be fictitious.

Correlation analysis is a potent instrument for examining the connections between variables, in conclusion. It can help with prediction and the creation of suitable solutions while also offering insightful information about the character of relationships. However, it's crucial to proceed with care and avoid assuming that a correlation equals a cause.

Multicollinearity

Regression analysis frequently encounters the issue of multicollinearity, which happens when there is a high degree of correlation between two or more predictor factors. This can cause a number of problems, including instability of the regression coefficients, trouble in deciphering the findings, and decreased prediction accuracy. A statistic called the Variance Inflation Factor (VIF) is used to gauge how multicollinear a regression model is. It gauges the extent to which multicollinearity has raised the variance of the estimated regression coefficient. High multicollinearity is indicated by a high VIF number, which can result in unstable and unreliable regression coefficients. Each predictor variable in the model is regressed against every other predictor variable in the model to compute VIF. The variance of the estimated regression coefficient for a given variable divided by the variance of the estimated regression coefficient in the absence of multicollinearity is known as the variable interaction factor, or VIF.

A VIF number of 1 denotes the absence of multicollinearity, while a value higher than 1 denotes the existence of multicollinearity to some extent. A VIF value higher than 5 typically denotes a high level of multicollinearity, according to a general rule of thumb. In conclusion, multicollinearity can pose a serious challenge to regression analysis, producing findings that are erratic and unstable. VIF is an effective instrument for assessing the degree of multicollinearity and locating predictor variables that might be a factor in the issue. We can enhance the precision and dependability of our regression models by recognising and resolving multicollinearity.As our precisions not getting very much effected and also our AQI is dependent on all the factors present as a independent variables or predictors so I am not addressing much about multicollinearity in my regression model to avoid underfitting.

Detecting Outliers

Data points known as outliers differ significantly from the remainder of the data. Measurement mistakes, experimental errors, or other data anomalies can result in outliers. The process of finding and analysing outliers in a collection is known as outlier detection. Because outliers can significantly affect statistical analyses like mean, standard deviation, and correlation, it is a crucial stage in the data analysis process. Visual examination, statistical analysis, and machine learning algorithms are just a few ways to find outliers. Plotting the data and locating any data points that are distant from the remainder of the data are steps in visual inspection. Calculated algorithms are used in statistical tests to pinpoint data points that stand out from the rest of the data, such as Z-scores or the Grubbs' test. Outliers can be found using machine learning techniques like classification and clustering. Once outliers have been found, the analyst can decide whether to remove them from the dataset or keep them. While eliminating outliers can increase the precision of statistical analyses, it can also result in data loss. In some circumstances, keeping outliers may be necessary, such as when the outliers are real data points that reflect exceptional occurrences or extreme values.

In conclusion, finding and analysing data points that significantly differ from the remainder of the data constitutes the crucial step of outlier detection in data analysis. Outliers can be found using a variety of techniques, and whether or not to keep them should depend on the particular objectives of the analysis and the characteristics of the data.

In our dataset we are detecting outliers by Swarm Plot, Violin Plot and Box Plot. In this model I am categorizing AQI values then using those categorizations for data visualization :- Swarm plot, violin plot, and box plot are all commonly used data visualization techniques in statistics and data analysis.

- A swarm plot is a type of scatter plot that displays the individual data points along with their distribution. Unlike traditional scatter plots, swarm plots ensure that the data points are not overlapping with each other by "swarming" them around the center. This makes it easier to visualize the distribution of the data points and identify any outliers or patterns in the data.

- A violin plot, on the other hand, is a type of plot that displays the distribution of the data using a kernel density estimate. The plot looks like a violin and hence, the name. The width of the plot at any point indicates the density of the data at that point. It is useful in visualizing the shape of the distribution of the data and its spread.

- Lastly, a box plot is another common plot used to visualize the distribution of data. It displays the distribution of the data using five statistical summary values - the minimum value, the maximum value, the median, and the first and third quartiles. The plot looks like a box with whiskers extending from the box. The box represents the middle 50% of the data, while the whiskers extend to the minimum and maximum values.

In conclusion, swarm plot, violin plot, and box plot are all useful data visualization techniques that provide different insights into the distribution of the data. They can help in identifying patterns, outliers, and other important features of the data, and are widely used in data analysis and statistics. Some of the machine learning algorithm that I am using for model training and then judging the best among them using error metrics and performance metrics.

Multiple Linear Regression

A dependent variable and several independent factors are modelled using the statistical technique known as multiple linear regression. It is a development of straightforward linear regression, which takes only one independent variable into account. In a multiple linear regression, each independent variable has a different coefficient, and the dependent variable is assumed to be a linear product of the independent variables. To estimate these coefficients and use them to forecast the dependent variable is the aim of multiple linear regression. Multiple linear regression has the benefit of being able to simulate intricate relationships between the dependent variable and numerous independent factors.

Logistic Regression

In logistic regression, the connection between the independent variables and the likelihood that the dependent variable will take on a particular value are modelled using a logistic function. Any value of the independent variables that the logistic function maps to a probability value between 0 and 1 can be understood as the odds that the dependent variable will take on a particular value. The ability to simulate non-linear relationships between the independent factors and the dependent variable is one of the benefits of logistic regression. This is beneficial when there is a nonlinear connection between the variables, such as when there are nonlinear decision boundaries. It is crucial to remember that logistic regression makes the assumption that the independent factors are unrelated to one another, which can cause problems with multicollinearity. Furthermore, logistic regression makes the assumption that all levels of the independent variables have the same connection.

K Nearest Neighbour:

A non-parametric machine learning algorithm called K nearest neighbours (KNN) is used for classification and regression jobs. It is a straightforward algorithm that finds the K data points in the training set that are the closest to a new observation (i.e., nearest neighbours) and uses the labels of those neighbours to forecast the label of the new observation. The K in KNN stands for the amount of closest neighbours used in the prediction process. The algorithm would locate the three neighbours who were nearest to the new observation, for instance, if K=3, and then make a prediction based on the labels of those three neighbours. The algorithm determines the distance between the new observation and the other observations in the training set using a distance measure, such as Euclidean distance, in order to identify the closest neighbours. The algorithm then chooses the K samples that are closest to the new observation in terms of distance. For classification tasks, the label that appears most frequently among the K closest neighbours is the one that is forecast for the new observation. The predicted value of the new data in regression tasks is the average of the values of the K closest neighbours.

KNN has a number of benefits, including its simplicity and convenience of use. It can be applied to jobs involving classification and regression, and it performs well when dealing with both linear and non-linear issues. The option of K can significantly affect how well the algorithm performs, but KNN can be sensitive to outliers and noise in the data.

Here are the steps involved in implementing the KNN algorithm:

- Load the data: First we need to load the dataset into memory. This could be done using a library like Pandas, or by reading the data from a file.
- Split the data: The data is typically split into training and testing sets. Build the model using training set, while the testing set is going to use for evaluating the performance of the model.
- Preprocess the data: Before the data can be used for modeling, it may need to be preprocessed. This could involve tasks like normalization, feature scaling, or

imputation of missing values.

- Choose the value of k: The value of k is a hyperparameter that must be chosen before running the algorithm. It is typically chosen using cross-validation or other model selection techniques.
- Calculate distances: Once the value of k has been chosen, the algorithm calculates the distance between the new data point and all the points in the training set.
- Find the k nearest neighbors: The k nearest neighbors are identified based on their distance from the new data point.
- Assign a label: Once the k nearest neighbors have been identified, the algorithm assigns a label to the new data point based on the majority class among its k neighbors.
- Evaluate the model: Finally, the performance of the model is evaluated using the testing set. This could involve metrics like accuracy, precision, recall, or F1 score.

By following these steps, the KNN algorithm can be implemented to solve classification and regression problems.

Decision Tree

Decision tree is a supervised machine learning algorithm . Generally used for both classification and regression jobs. In order to maximise the homogeneity (i.e., similarity) of the dependent variable within each subgroup, the data are recursively partitioned into subsets based on the values of the independent variables. The algorithm chooses the independent variable that divides the data into subsets with the most uniform dependent variable at each stage. Typically, a statistic like information gain or Gini impurity is used to quantify this. The algorithm keeps recursively partitioning the data until it either hits a maximum depth or a predetermined level of homogeneity. (i.e., number of levels). After the tree has been constructed, new data can be categorised by moving down the tree in accordance with the values of the independent variables.

Random Forest

A well-liked ensemble machine learning algorithm called random forest is used for both classification and regression jobs. It functions by constructing a number of decision trees, each of which is trained using a random subset of the input data and the independent factors.

During training, the algorithm bootstrap selects a random subset of the data with replacement and creates a decision tree from it. However, the algorithm only takes into account a random subset of the variables at each split in the tree, rather than using the optimal split based on all the available variables.

Once all the trees have been constructed, it is possible to classify or forecast new observations by running them through each tree and calculating the average (for regression) or majority vote (for classification) of the predictions. In addition to being able to manage both categorical and numerical data, missing data, and overfitting, random forest has a number of benefits. The algorithm lowers the possibility of overfitting to the training data by creating numerous trees on random subsets of the data and variables.

Performance Indicators

The percentage of variance in the target variable that is explained by the independent variables in the model is shown statistically by the R-squared number. Its values fall between 0 and 1, with a value of 1 indicating that the model fully accounts for the variance in the objective variable and a value of 0 indicating that it does not. A helpful metric to assess a model's goodness of fit is its R-squared value. It cannot assess whether a model is suitable for prediction and is sensitive to the amount of variables in the model, among other drawbacks. A measure for assessing the effectiveness of classification model performance is accuracy score. It shows how many accurate predictions the model made as a percentage of all the predictions. Its values vary from 0 to 1, with 1 denoting perfect accuracy and 0 denoting no accuracy. Although accuracy score is a helpful measure for assessing classification models, it has some drawbacks, including favouring models that predict the majority class and being unable to handle unbalanced datasets.

In conclusion, precision score and R-squared value are both valuable performance indicators with distinct advantages and disadvantages. The specific issue at hand and the characteristics of the dataset will determine which metric is most appropriate. To obtain a more thorough understanding of the model's performance, it is frequently a smart practise to use a variety of performance metrics.

Error metrics Predictive model success is frequently assessed in machine learning and statistical modelling using error metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

- MAE calculates the average absolute difference between a target variable's real and predicted values. A more reliable measure than RMSE, it is less susceptible to outliers. Since MAE shows the average difference between predictions and actual values in the same units as the objective variable, it is simple to understand. The amount or direction of the errors, however, are not disclosed by MAE.

- The average squared difference between a goal variable's actual and predicted values is measured by RMSE, in contrast. It is more sensitive to outliers than MAE because it weighs big errors more heavily than small errors. Since RMSE displays the typical size of the errors in the same units as the goal variable, it is also easier to understand than MAE. When comparing models that predict various kinds of variables, RMSE may be deceptive because it is sensitive to the scale of the target variable.
- In summation, MAE and RMSE are both practical error metrics with unique advantages and disadvantages. The best metric to use will rely on the particular issue at hand and the characteristics of the target variable. To get a more complete picture of the model's success, it is frequently wise to use both metrics.

Anova Analysis

A statistical technique called ANOVA (Analysis of Variance) is used to examine differences between two or more categories. It is an effective method for assessing whether there is a substantial difference between means when they are compared. ANOVA measures variability by contrasting it with variability within categories. It indicates that there is a significant difference in the means of the groups if the variability between groups is noticeably higher than the variability within groups.

ANOVA has a number of benefits, one of which is the ability to handle multiple groups at once, which can save time and resources when compared to performing separate t-tests between each set of groups. ANOVA can be expanded to include more variables, such as covariates, in order to account for other potential affecting factors. The term for this is ANCOVA. (Analysis of Covariance).

ANOVA is frequently employed in a wide range of disciplines, including business, biology, and psychology. It can be used, for instance, to evaluate the performance of various marketing strategies in a business context or to compare the efficacy of various treatments in a medical study. It's essential to remember that ANOVA makes the assumptions that the data is normally distributed and that the variances between the groups are equal. Inaccurate outcomes can result from violating these presumptions. ANOVA is also a hypothesis-testing technique and does not reveal the magnitude of the effect or the usefulness of the differences that are noticed. ANOVA is a strong statistical tool for comparing means and identifying whether there is a substantial difference between two or more groups, so that's what we've got there. It can be expanded to include more variables and handle numerous groups at once. It's crucial to understand the method's presumptions and restrictions, though.

Simulation

Making a model or representation of a real-world system or phenomenon in order to study its behaviour and forecast its results under various circumstances is known as simulation. In order to build a virtual environment that accurately represents the behaviour of the real-world system, mathematical and statistical models are used. Model development, data collection and analysis, validation, and experimentation are typical stages in the simulation method. A mathematical or statistical model is developed during model development to reflect the system under study, and data is gathered to estimate the model's parameters. The model is then verified by contrasting its results with data from the actual world or with information from other trustworthy sources. After the model has been verified, experiments can be carried out to determine how various situations or conditions will turn out.

Numerous disciplines, including engineering, economics, biology, physics, and social studies, use simulation. It is especially helpful when studying a system directly is challenging or unattainable, or when conducting experiments would be costly or time-consuming. The ability to study complicated systems, play with various scenarios and conditions, and carry out experiments that might be too risky or impractical in the real world are some benefits of simulation. The need for precise data and the challenge of accurately capturing all the pertinent factors that affect how a real-world system behaves are two drawbacks of simulation, though. In summary, simulation is an effective instrument for analysing the behaviour of complex systems and forecasting their results under various circumstances. The process entails developing a mathematical or statistical model of the system under study, using it to run tests and consider various scenarios. Although simulation has many benefits, it also has drawbacks, so it is important to carefully examine the assumptions and constraints of the model being used.

.

6. CODE:

➢ Import the Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
from sklearn import metrics
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.tree import DecisionTreeRegressor
```

➢ Reading Dataset using pandas

```
data=pd.read_csv("C:\\Users\\Palak Goel\\Desktop\\AQI\\AQI_data.csv", encoding=
'unicode_escape')

data.head(40)

data.shape

data.info()

data.isnull().sum()

data.describe()

data.nunique()

df=data

df.columns

df['State_UT'].value_counts()
```

```
matrix1 = df1.corr()
# print correlation matrix
print("Correlation Matrix : ")
print(matrix1)
```

➢ Checking all null values and treating those null values

```
Null_values1 = df1.isnull().sum().sort_values(ascending=False)
Null_values_percentage1 =
((df1.isnull().sum()/df.isnull().count())*100).sort_values(ascending=False)
```

➢ Concatenating total null values and their percentage of missing values for further imputation or column deletion

```
missing_data_with_percentage1 = pd.concat([Null_values1, Null_values_percentage1],
axis=1, keys=['Total1', 'Percent1'])
```

Since we already know that our dataset contains missing values , and we need to fill them for our further analysis . We will be using Imputation to fill our missing values. Imputation is the technique of replacing missing data with substituted mean values . Because missing data can create wrong outputs for analyzing data.

```
by_State1=df.groupby('State_UT')
def impute_mean_function(series1):
    return series.fillna(series1.mean())
df['PM2.5']=by_State1['PM2.5'].transform(impute_mean_function)
df.fillna(0, inplace=True)
df.isnull().sum()
```

➢ Reading the data for states

```
df1.groupby('State_UT')[['PM2.5','PM10','SO2','NO2']].mean()
```

➢ Data Visualization

- **Visualization for the state having higher SO2 levels in the air which is Jharkhand followed by Tripura**

```
plt.figure(figsize=(40, 10))
plt.xticks(rotation=90)
sns.barplot(x='State_UT',y='SO2',data=df1);
plt.rcParams['figure.figsize']=(50,20)
df1[['SO2','State_UT']].groupby(["State_UT"]).mean().sort_values(by='SO2').plot.bar
(color='green')
plt.show()
```

- **Visualization for the state having higher NO2 levels in the air which is Delhi followed by Telangana**

```
plt.figure(figsize=(40, 10))
plt.xticks(rotation=90)
sns.barplot(x='State_UT1',y='NO2',data=df1);
```

```
df1[['NO2','State_UT1']].groupby(["State_UT1"]).mean().sort_values(by='NO2').plot.b
ar(color='purple')
plt.show()
```

- **Visualization for the state having higher PM2.5 levels in the air which is Delhi followed by Jharkhand**

```
plt.figure(figsize=(40, 10))
plt.xticks(rotation=90)
sns.barplot(x='State_UT1',y='PM2.5',data=df1);
df1[['PM2.5','State_UT1']].groupby(["State_UT1"]).mean().sort_values(by='PM2.5').pl
ot.bar(color='blue')
plt.show()
```

- **Visualization for the state having higher PM10 levels in the air which is Delhi followed by Uttar Pradesh**

```
plt.figure(figsize=(30, 10))
plt.xticks(rotation=90)
sns.barplot(x='State_UT1',y='PM10',data=df1);
df1[['PM10','State_UT1']].groupby(["State_UT1"]).mean().sort_values(by='PM10').plot
.bar(color='green')
plt.show()
```

➢ Function for calculating SO2 individual pollutant subindex(SO2i)

```
def aut_SOi(so2):
    i=0
    if (so2<=40):
     i= so2*(50/40)
    elif (so2>40 and so2<=80):
     i= 50+(so2-40)*(50/40)
    elif (so2>80 and so2<=380):
     i= 100+(so2-80)*(100/300)

    elif (so2>380 and so2<=800):
     i= 200+(so2-380)*(100/420)
    elif (so2>800 and so2<=1600):
     i= 300+(so2-800)*(100/800)
    elif (so2>1600):
     i= 400+(so2-1600)*(100/800)
    return i
df['SO2i']=df['SO2'].apply(aut_SOi)
data1= df[['SO2','SO2i']]
data1.head()
```

➢ Function for calculating NO2 individual pollutant subindex(NO2i)

```
def aut_Noi(no2):
    i=0
    if(no2<=40):
     i= no2*50/40
    elif(no2>40 and no2<=80):
     i= 50+(no2-40)*(50/40)
    elif(no2>80 and no2<=180):
     i= 100+(no2-80)*(100/100)
    elif(no2>180 and no2<=280):
     i= 200+(no2-180)*(100/100)
    elif(no2>280 and no2<=400):
     i= 300+(no2-280)*(100/120)
    else:
     i= 400+(no2-400)*(100/120)
    return i
df['No1i']=df['NO2'].apply(aut_Noi)
data1= df[['NO2','No1i']]
data1.head()
# automating the individual pollutant index for no2(nitrogen dioxide)
```

➢ Function for calculating PM2.5 individual pollutant Subindex(PM2.5i)

```
def aut_PM25(x):
    if x <= 30:
        return x * 50 / 30
    elif x <= 60:
        return 50 + (x - 30) * 50 / 30
    elif x <= 90:
        return 100 + (x - 60) * 100 / 30
    elif x <= 120:
        return 200 + (x - 90) * 100 / 30
    elif x <= 250:
        return 300 + (x - 120) * 100 / 130
    elif x > 250:
        return 400 + (x - 250) * 100 / 130
    else:
        return 0

df["PM2.5i"] = df["PM2.5"].apply(lambda x: aut_PM25(x))
data1= df[['PM2.5','P     M2.5i']]
data1.head()
```

➢ Function to calculate spm (PM10) individual pollutant subindex(PMi)

```
 def aut_PMi(pm):
     i=0
     if(pm<=50):
      i=pm*50/50
     elif(pm>50 and pm<=100):
      i=50+(spm-50)*(50/50)
     elif(pm>100 and pm<=250):
      i= 100+(spm-100)*(100/150)
     elif(pm>250 and pm<=350):
      i=200+(pm-250)*(100/100)
     elif(pm>350 and pm<=430):
      i=300+(pm-350)*(100/80)
     else:
      i=400+(pm-430)*(100/80)
     return i
```

```
df['PMi']=df['PM10'].apply(aut_PMi)
data1= df[['PM10','PMi']]
data1.head()
```

➢ Function for calculating the air quality index (AQI)

```
def aut_aqi(si,ni,pm2.5i,pmi):
    i=0
    if(si>ni and si>rspmi and si>spmi):
     i=si
    if(ni>si and ni>rspmi and ni>spmi):
     i=ni
    if(rspmi>si and rspmi>ni and rspmi>spmi):
     i=pm2.5i
    if(spmi>si and spmi>ni and spmi>rspmi):
     i=pmi
    return i

df['AQI']=df.apply(lambda x:aut_aqi(x['SOi'],x['Noi'],x['pm2.5i'],x['PMi']),axis=1)
data1= df[['State_UT','SOi','Noi','pm2.5i','PMi','AQI']]
data1.head()
```

➢ Using threshold values to classify a particular values as healthy, moderate, poor, unhealthy, very unhealthy and Hazardouss

```
def AQI_Ran(m):
    if m<=50:
        return "healthy"
    elif m>50 and m<=100:
        return "Moderatee"
    elif m>100 and m<=200:
        return "Poor"
    elif m>200 and m<=300:
        return "Unhealthy"

    elif m>300 and m<=400:
        return "Very Unhealthy"
    elif m>400:
        return "Hazardouss"

df1['AQI_Ran'] = df1['AQI'].apply(AQI_Ran)
df1.head()


dataframe1 = pd.DataFrame(df1, columns=['SO2', 'NO2', 'PM10', 'PM2.5', 'AQI'])

#Exploring air pollution state-wise

States1=df1.groupby(['State_UT1','City'],as_index=False).mean()
State1=states1.groupby(['State_UT1'],as_index=False).mean()
State1

sns.heatmap(dataframe1.corr())
```

Finding the correlation between explanatory variable and explained variable i.e AQI

```
Matrix2 = dataframe.corr()
# print correlation matrix
print("Correlation Matrix: ")
print(matrix2)
```

Checking Skewness

```
Dataframe1.skew(axis = 0, skipna = True)
```

Swarm Plot of AQI Range

```
sns.set(rc={'figure.figsize':(25,8)})
plt.xticks(fontsize=15)
sns.swarmplot(data=df1,x='AQI_Ran',y='AQI',order=['Healthy', 'Moderatee', 'Poor',
'Unhealthy', 'Very Unhealthy', 'Hazardouss'])
```

Box Plot of AQI Range

```
sns.set(rc={'figure.figsize':(15,11)})
plt.xticks(fontsize=15)
sns.boxplot(data=df1,x='AQI_Range',y='AQI',order=[' Healthy ', 'Moderatee', 'Poor',
'Unhealthy', 'Very Unhealthy', 'Hazardous'])
```

Violin plot of AQI Range

```
sns.set(rc={'figure.figsize':(15,11)})
plt.xticks(fontsize=15)
sns.violinplot(data=df,x='AQI_Range',y='AQI',order=[' Healthy, 'Moderatee', 'Poor',
'Unhealthy', 'Very Unhealthy', 'Hazardousss'])
```

AQI Values Distribution

```
sns.set(rc={'figure.figsize':(15,5)})
p.axes.set_title("AQI Distribution",fontsize=25)
plt.xticks(fontsize=15)
p=sns.distplot(df['AQI'],color='yellow')
```

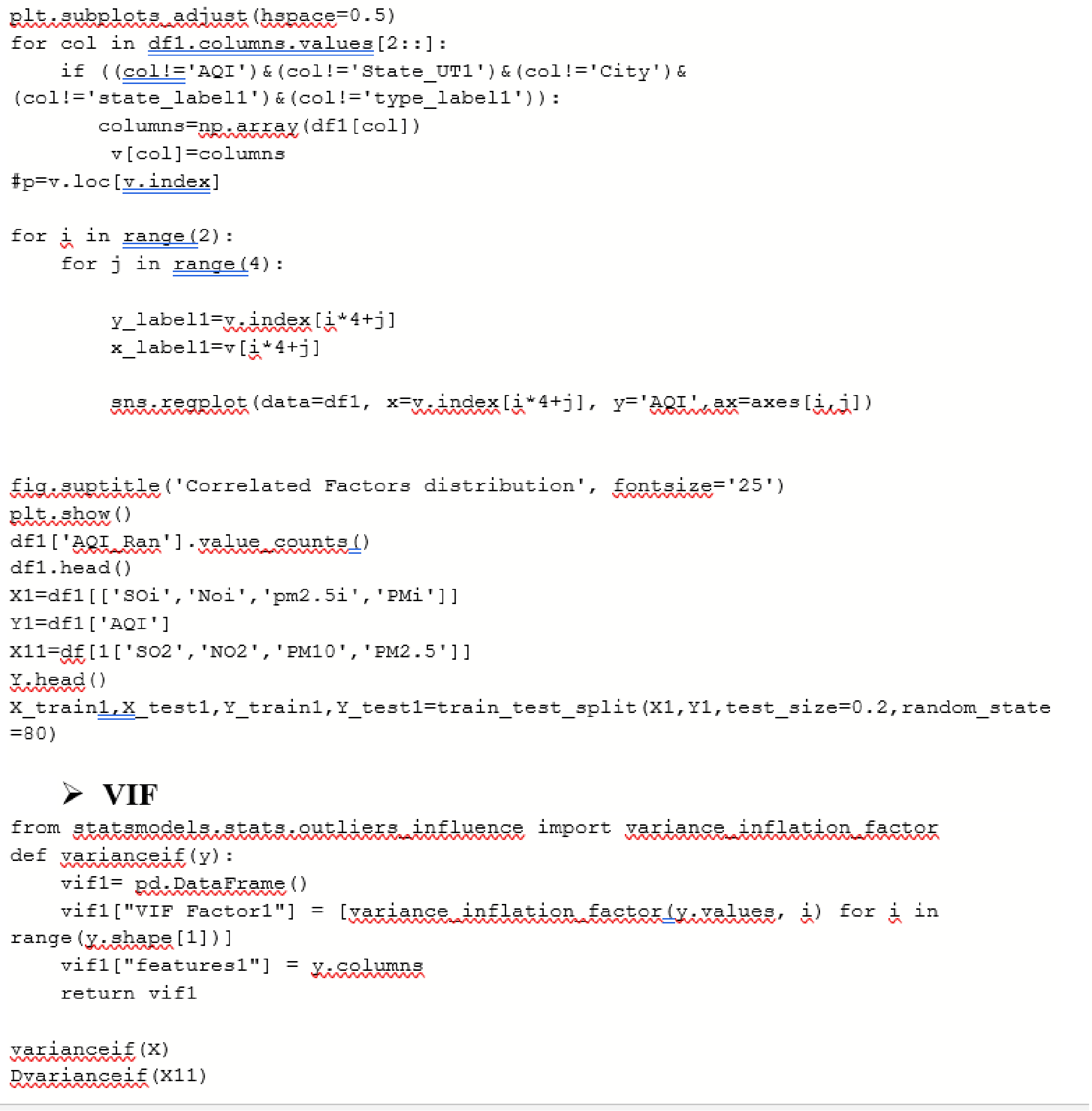
Categorical Conversion

```
df['state_label1'] = df['State_UT1'].astype('category')
cat_columns1 = df.select_dtypes(['category1']).columns
cat_columns1
```

Regplot

Distribution of important predictor variables and their relation with dependent variable

```
fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(35,21))
v = pd.Series()
```

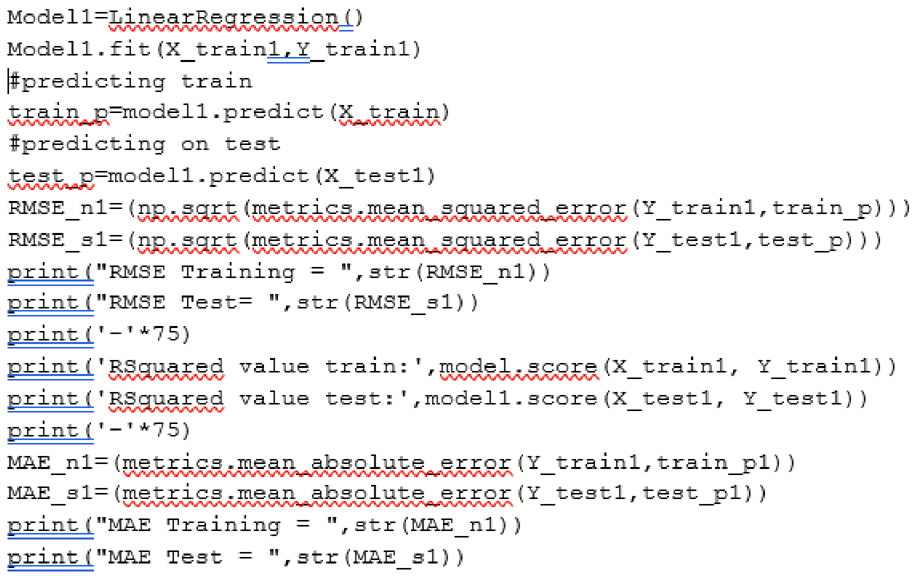


```
plt.subplots_adjust(hspace=0.5)
for col in df1.columns.values[2::]:
    if ((col!='AQI')&(col!='State_UT1')&(col!='City')&
(col!='state_label1')&(col!='type_label1')):
       columns=np.array(df1[col])
        v[col]=columns
#p=v.loc[v.index]

for i in range(2):
    for j in range(4):

        y_label1=v.index[i*4+j]
        x_label1=v[i*4+j]

        sns.regplot(data=df1, x=v.index[i*4+j], y='AQI',ax=axes[i,j])


fig.suptitle('Correlated Factors distribution', fontsize='25')
plt.show()
df1['AQI_Ran'].value_counts()
df1.head()
X1=df1[['SOi','Noi','pm2.5i','PMi']]
Y1=df1['AQI']
X11=df[1['SO2','NO2','PM10','PM2.5']]
Y.head()
X_train1,X_test1,Y_train1,Y_test1=train_test_split(X1,Y1,test_size=0.2,random_state
=80)
```

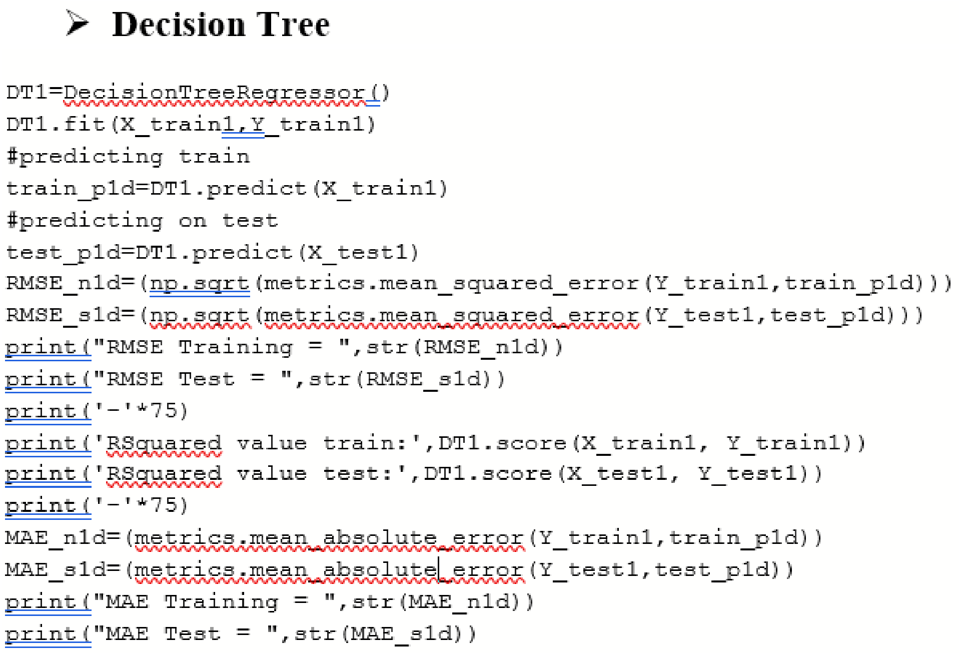
➢ VIF

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
def varianceif(y):
    vif1= pd.DataFrame()
    vif1["VIF Factor1"] = [variance_inflation_factor(y.values, i) for i in
range(y.shape[1])]
    vif1["features1"] = y.columns
    return vif1

varianceif(X)
Dvarianceif(X11)
```

➢ LINEAR REGRESSION

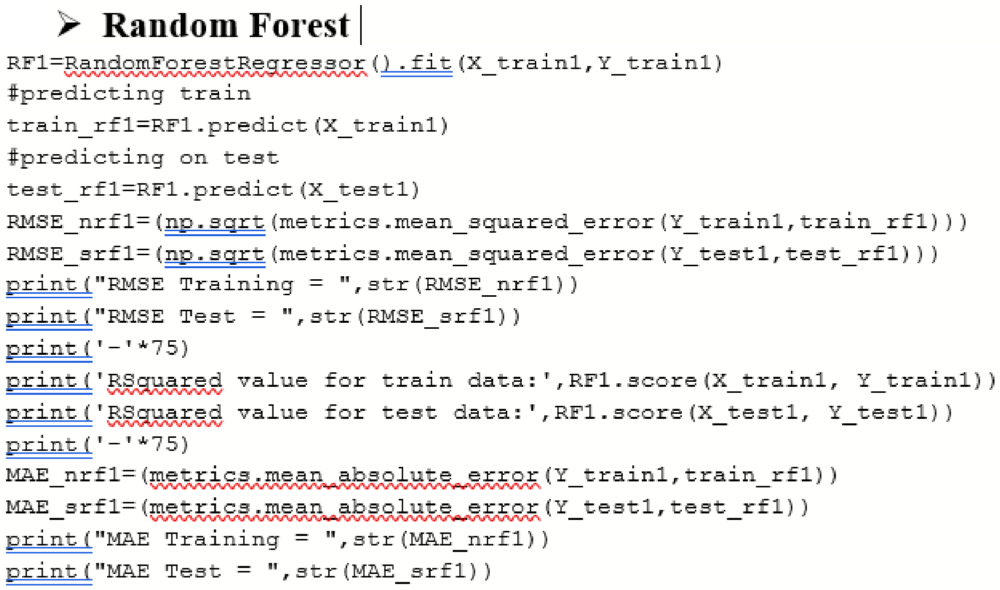


```
Model1=LinearRegression()
Model1.fit(X_train1,Y_train1)
#predicting train
train_p=model1.predict(X_train)
#predicting on test
test_p=model1.predict(X_test1)
RMSE_n1=(np.sqrt(metrics.mean_squared_error(Y_train1,train_p)))
RMSE_s1=(np.sqrt(metrics.mean_squared_error(Y_test1,test_p)))
print("RMSE Training = ",str(RMSE_n1))
print("RMSE Test= ",str(RMSE_s1))
print('-'*75)
print('RSquared value train:',model.score(X_train1, Y_train1))
print('RSquared value test:',model1.score(X_test1, Y_test1))
print('-'*75)
MAE_n1=(metrics.mean_absolute_error(Y_train1,train_p1))
MAE_s1=(metrics.mean_absolute_error(Y_test1,test_p1))
print("MAE Training = ",str(MAE_n1))
print("MAE Test = ",str(MAE_s1))
```



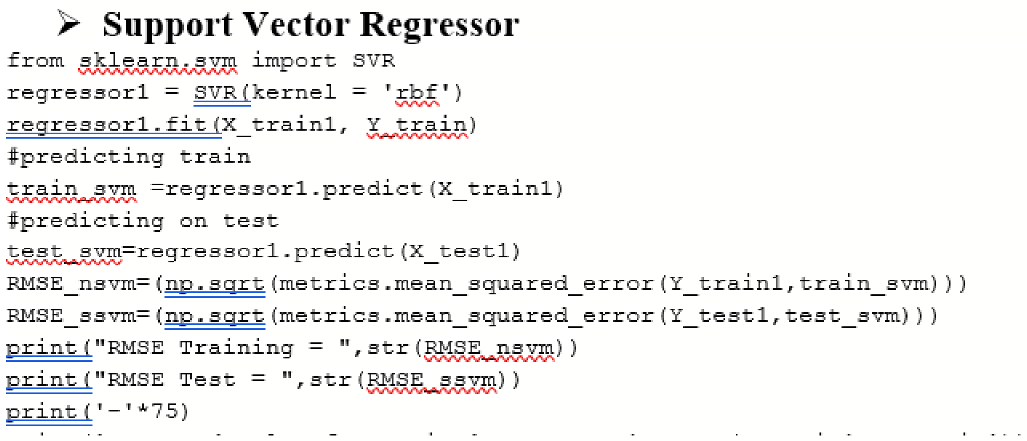
➢ Decision Tree

```
DT1=DecisionTreeRegressor()
DT1.fit(X_train1,Y_train1)
#predicting train
train_p1d=DT1.predict(X_train1)
#predicting on test
test_p1d=DT1.predict(X_test1)
RMSE_n1d=(np.sqrt(metrics.mean_squared_error(Y_train1,train_p1d)))
RMSE_s1d=(np.sqrt(metrics.mean_squared_error(Y_test1,test_p1d)))
print("RMSE Training = ",str(RMSE_n1d))
print("RMSE Test = ",str(RMSE_s1d))
print('-'*75)
print('RSquared value train:',DT1.score(X_train1, Y_train1))
print('RSquared value test:',DT1.score(X_test1, Y_test1))
print('-'*75)
MAE_n1d=(metrics.mean_absolute_error(Y_train1,train_p1d))
MAE_s1d=(metrics.mean_absolute_error(Y_test1,test_p1d))
print("MAE Training = ",str(MAE_n1d))
print("MAE Test = ",str(MAE_s1d))
```



➢ Random Forest

```
RF1=RandomForestRegressor().fit(X_train1,Y_train1)
#predicting train
train_rf1=RF1.predict(X_train1)
#predicting on test
test_rf1=RF1.predict(X_test1)
RMSE_nrf1=(np.sqrt(metrics.mean_squared_error(Y_train1,train_rf1)))
RMSE_srf1=(np.sqrt(metrics.mean_squared_error(Y_test1,test_rf1)))
print("RMSE Training = ",str(RMSE_nrf1))
print("RMSE Test = ",str(RMSE_srf1))
print('-'*75)
print('RSquared value for train data:',RF1.score(X_train1, Y_train1))
print('RSquared value for test data:',RF1.score(X_test1, Y_test1))
print('-'*75)
MAE_nrf1=(metrics.mean_absolute_error(Y_train1,train_rf1))
MAE_srf1=(metrics.mean_absolute_error(Y_test1,test_rf1))
print("MAE Training = ",str(MAE_nrf1))
print("MAE Test = ",str(MAE_srf1))
```



➢ Support Vector Regressor

```
from sklearn.svm import SVR
regressor1 = SVR(kernel = 'rbf')
regressor1.fit(X_train1, Y_train)
#predicting train
train_svm =regressor1.predict(X_train1)
#predicting on test
test_svm=regressor1.predict(X_test1)
RMSE_nsvm=(np.sqrt(metrics.mean_squared_error(Y_train1,train_svm)))
RMSE_ssvm=(np.sqrt(metrics.mean_squared_error(Y_test1,test_svm)))
print("RMSE Training = ",str(RMSE_nsvm))
print("RMSE Test = ",str(RMSE_ssvm))
print('-'*75)
print('RSquared value for train:',regressor1.score(X_train1, Y_train1))
print('RSquared value for test:',regressor1.score(X_test1, Y_test1))
print('-'*75)
MAE_nsvm=(metrics.mean_absolute_error(Y_train1,train_svm))
MAE_ssvm=(metrics.mean_absolute_error(Y_test1,test_svm))
print("MAE Training = ",str(MAE_nsvm))
print("MAE Test = ",str(MAE_ssvm))
```

➢ CLASSIFICATION ALGORITHM

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

df11['AQI_label'] = df1['AQI_Ran'].astype('category')
cat_columns1 = df1.select_dtypes(['category']).columns
df[cat_columns1] = df[cat_columns1].apply(lambda x: x.cat.codes)
df.head()

X2 = df[['SOi','Noi','pm2.5i','PMi']]
Y2 = df['AQI_Ran']
# Splitting the data into predictor and dependent columns for classification
X_train4, X_test4, Y_train4, Y_test4 = train_test_split(X2, Y2, test_size=0.33,
random_state=75)
# Splitting the data into training and testing data
```

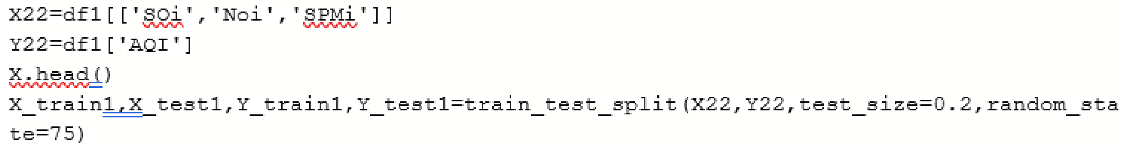
➢ LOGISTIC REGRESSION

```
#fit model on train data
Log_r = LogisticRegression().fit(X_train4, Y_train4)
#predict for training
train_lgr = log_r.predict(X_train4)
#predict for testing
test_lgr = log_r.predict(X_test4)
print("Model accuracy(train): ", accuracy_score(Y_train4, train_lgr))
print('-'*75)
print("Model accuracy (test) : ", accuracy_score(Y_test4, test_lgr))
# Kappa Score.
print('KappaScore: ', metrics.cohen_kappa_score(Y_test4,test_lgr))
```

➢ K NEAREST NEIGHBOUR

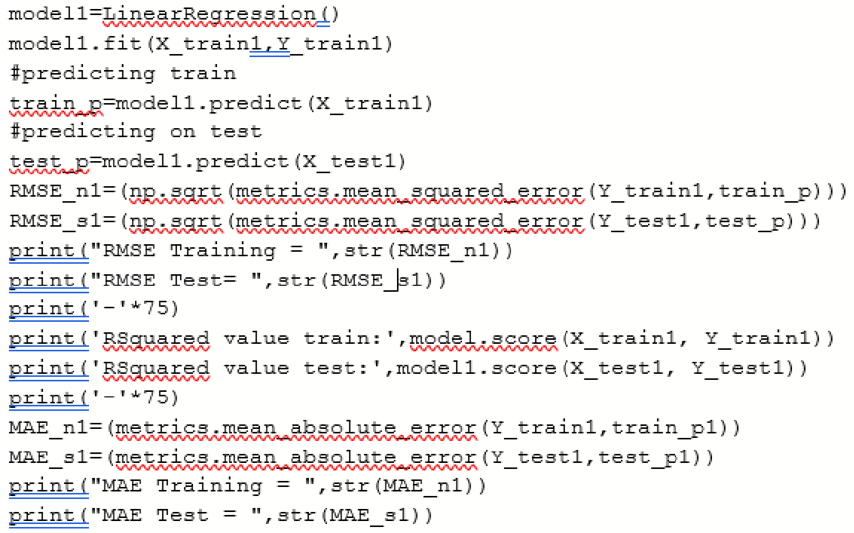
```
#fit the model on train data
KN1 = KNeighborsClassifier().fit(X_train4,Y_train4)
#predict on train
train_KN1 = KN1.predict(X_train4)
#predict on test
test_KN1 = KN1.predict(X_test4)
print("Model accuracy(train): ", accuracy_score(Y_train4, train_KN1))
print('-'*75)
print("Model accuracy (test) : ", accuracy_score(Y_test4, test_KN1))
# Kappa Score.
print('KappaScore: ', metrics.cohen_kappa_score(Y_test4,test_KN1))
```

➢ After Removing PM2.5 as a independent variable in model training



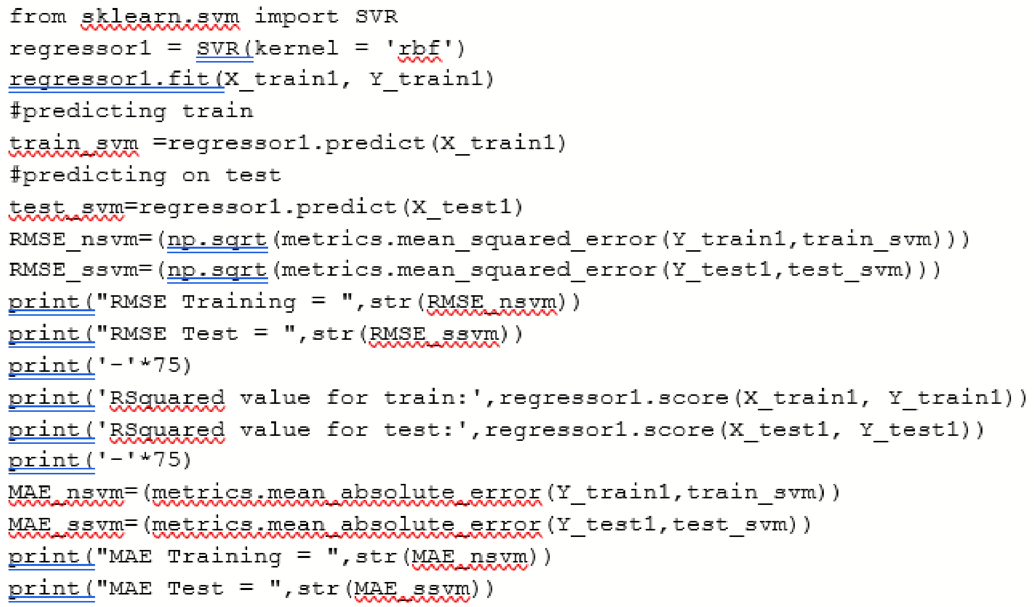
```
X22=df1[['SOi','Noi','SPMi']]
Y22=df1['AQI']
X.head()
X_train1,X_test1,Y_train1,Y_test1=train_test_split(X22,Y22,test_size=0.2,random_sta
te=75)
```

➢ Linear regression



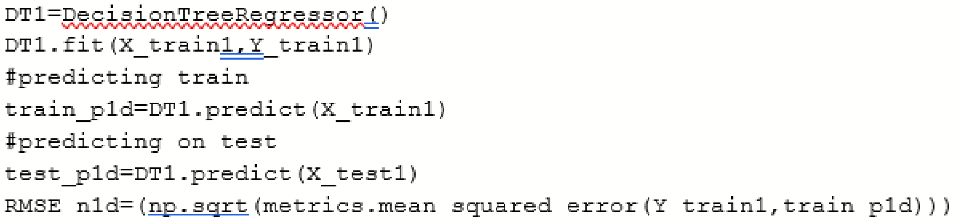
```
model1=LinearRegression()
model1.fit(X_train1,Y_train1)
#predicting train
train_p=model1.predict(X_train1)
#predicting on test
test_p=model1.predict(X_test1)
RMSE_n1=(np.sqrt(metrics.mean_squared_error(Y_train1,train_p)))
RMSE_s1=(np.sqrt(metrics.mean_squared_error(Y_test1,test_p)))
print("RMSE Training = ",str(RMSE_n1))
print("RMSE Test= ",str(RMSE_s1))
print('-'*75)
print('RSquared value train:',model.score(X_train1, Y_train1))
print('RSquared value test:',model1.score(X_test1, Y_test1))
print('-'*75)
MAE_n1=(metrics.mean_absolute_error(Y_train1,train_p1))
MAE_s1=(metrics.mean_absolute_error(Y_test1,test_p1))
print("MAE Training = ",str(MAE_n1))
print("MAE Test = ",str(MAE_s1))
```

➢ Support vector Regressor



```
from sklearn.svm import SVR
regressor1 = SVR(kernel = 'rbf')
regressor1.fit(X_train1, Y_train1)
#predicting train
train_svm =regressor1.predict(X_train1)
#predicting on test
test_svm=regressor1.predict(X_test1)
RMSE_nsvm=(np.sqrt(metrics.mean_squared_error(Y_train1,train_svm)))
RMSE_ssvm=(np.sqrt(metrics.mean_squared_error(Y_test1,test_svm)))
print("RMSE Training = ",str(RMSE_nsvm))
print("RMSE Test = ",str(RMSE_ssvm))
print('-'*75)
print('RSquared value for train:',regressor1.score(X_train1, Y_train1))
print('RSquared value for test:',regressor1.score(X_test1, Y_test1))
print('-'*75)
MAE_nsvm=(metrics.mean_absolute_error(Y_train1,train_svm))
MAE_ssvm=(metrics.mean_absolute_error(Y_test1,test_svm))
print("MAE Training = ",str(MAE_nsvm))
print("MAE Test = ",str(MAE_ssvm))
```

➢ Decision Tree



```
DT1=DecisionTreeRegressor()
DT1.fit(X_train1,Y_train1)
#predicting train
train_p1d=DT1.predict(X_train1)
#predicting on test
test_p1d=DT1.predict(X_test1)
RMSE n1d=(np.sqrt(metrics.mean squared error(Y train1,train p1d)))
```

```
RMSE_s1d=(np.sqrt(metrics.mean_squared_error(Y_test1,test_p1d)))
print("RMSE Training = ",str(RMSE_n1d))
print("RMSE Test = ",str(RMSE_s1d))
print('-'*75)
print('RSquared value train:',DT1.score(X_train1, Y_train1))
print('RSquared value test:',DT1.score(X_test1, Y_test1))
print('-'*75)
MAE_n1d=(metrics.mean_absolute_error(Y_train1,train_p1d))
MAE_s1d=(metrics.mean_absolute_error(Y_test1,test_p1d))
print("MAE Training = ",str(MAE_n1d))
print("MAE Test = ",str(MAE_s1d))
```

➢ Random Forest

```
RF1=RandomForestRegressor().fit(X_train1,Y_train1)
#predicting train
train_rf1=RF1.predict(X_train1)
#predicting on test
test_rf1=RF1.predict(X_test1)
RMSE_nrf1=(np.sqrt(metrics.mean_squared_error(Y_train1,train_rf1)))
RMSE_srf1=(np.sqrt(metrics.mean_squared_error(Y_test1,test_rf1)))
print("RMSE Training = ",str(RMSE_nrf1))
print("RMSE Test = ",str(RMSE_srf1))
print('-'*75)
print('RSquared value for train data:',RF1.score(X_train1, Y_train1))
print('RSquared value for test data:',RF1.score(X_test1, Y_test1))
print('-'*75)
MAE_nrf1=(metrics.mean_absolute_error(Y_train1,train_rf1))
MAE_srf1=(metrics.mean_absolute_error(Y_test1,test_rf1))
print("MAE Training = ",str(MAE_nrf1))
print("MAE Test = ",str(MAE_srf1))
```

➢ Classification Algorithm

```
X23 = df[['SOi','Noi','SPMi']]
Y23 = df['AQI_Ran']
X_train4, X_test4, Y_train4, Y_test4 = train_test_split(X23, Y23, test_size=0.33,
random_state=75)
```

➢ Logistic Regression

```
#fit the model on train data
logr = LogisticRegression().fit(X_train4, Y_train4)
#predicting for train
train_lgr = log_r.predict(X_train4)
#predicting for  test
test_lgr = log_r.predict(X_test4)

print("Model accuracy(train): ", accuracy_score(Y_train4, train_lgr))
print('-'*75)
print("Model accuracy (test) : ", accuracy_score(Y_test4, test_lgr))
```

➢ K Nearest Neighbour

```
#fit the model on train data
```

```
KN1 = KNeighborsClassifier().fit(X_train4,Y_train4)
#predict on train
train_KN1 = KN1.predict(X_train4)
#predict on test
test_KN1 = KN1.predict(X_test4)
print("Model accuracy(train): ", accuracy_score(Y_train4, train_KN1))
print('-'*75)
print("Model accuracy (test) : ", accuracy_score(Y_test4, test_KN1))
# Kappa Score.
print('KappaScore: ', metrics.cohen_kappa_score(Y_test4,test_KN1))
```

➢ ANOVA ANALYSIS

```
!pip install statsmodels

import statsmodels.api as sm
from statsmodels.formula.api import ols

mod1 = ols('Y1 ~ X1', data=df1).fit()
print(mod1.summary().tables[1])
aov_table1 = sm.stats.anova_lm(mod1, typ=2)
print(aov_table1)
print("R square value = ", mod1.rsquared)
```

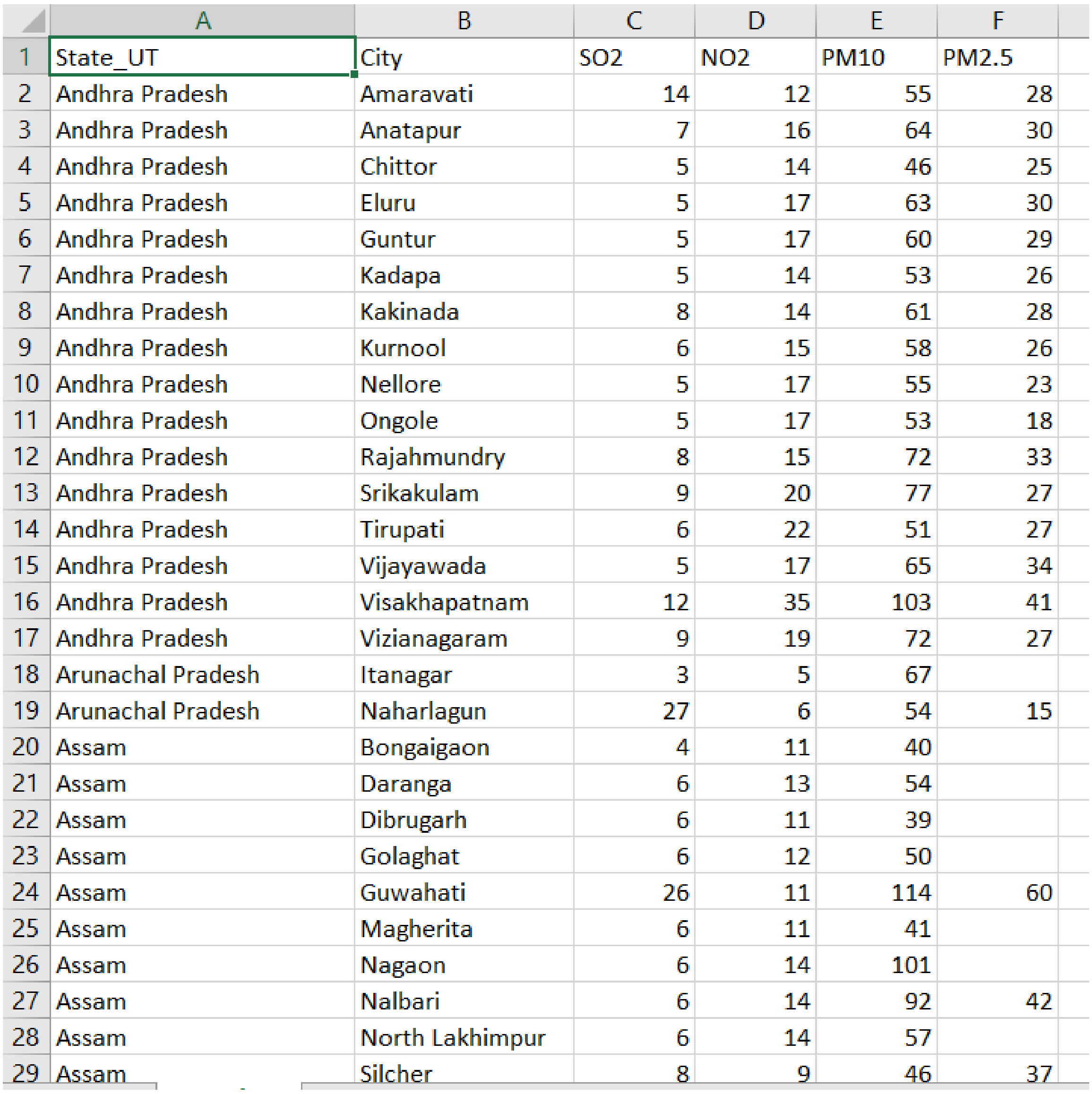
➢ Anova Analysis after removing PM2.5 as a independent variable

```
X = df[['SOi','Noi','SPMi']]
Mod2 = ols('Y1 ~ X', data=df1).fit()
print(mod2.summary().tables[1])
aov_table2 = sm.stats.anova_lm(mod2, typ=2)
print(aov_table2)
print("R square = ", mod.rsquared)
```

➢ Anova Analysis after removing PM2.5 and PM10 as a independent variable

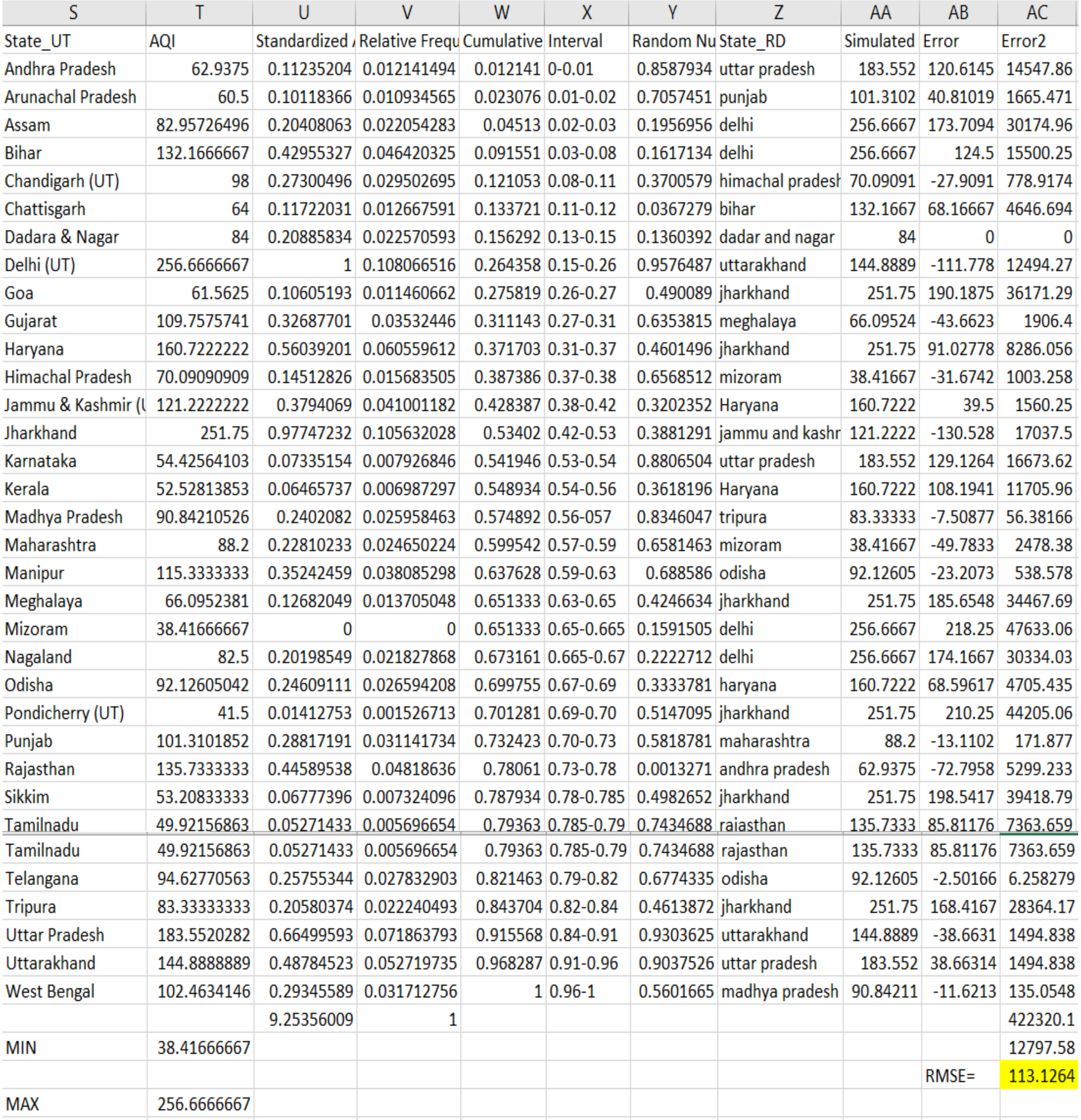
```
X1=df[['SOi','Noi']]
Mod3 = ols('Y1 ~ X1', data=df1).fit()
print(mod3.summary().tables[1])
aov_table3 = sm.stats.anova_lm(mod3, typ=2)
print(aov_table3)
print("R square = ", mod.rsquared)
```

DATASET:- Consist of 400 rows and 6 columns



	A	B	C	D	E	F
1	State_UT	City	SO2	NO2	PM10	PM2.5
2	Andhra Pradesh	Amaravati	14	12	55	28
3	Andhra Pradesh	Anatapur	7	16	64	30
4	Andhra Pradesh	Chittor	5	14	46	25
5	Andhra Pradesh	Eluru	5	17	63	30
6	Andhra Pradesh	Guntur	5	17	60	29
7	Andhra Pradesh	Kadapa	5	14	53	26
8	Andhra Pradesh	Kakinada	8	14	61	28
9	Andhra Pradesh	Kurnool	6	15	58	26
10	Andhra Pradesh	Nellore	5	17	55	23
11	Andhra Pradesh	Ongole	5	17	53	18
12	Andhra Pradesh	Rajahmundry	8	15	72	33
13	Andhra Pradesh	Srikakulam	9	20	77	27
14	Andhra Pradesh	Tirupati	6	22	51	27
15	Andhra Pradesh	Vijayawada	5	17	65	34
16	Andhra Pradesh	Visakhapatnam	12	35	103	41
17	Andhra Pradesh	Vizianagaram	9	19	72	27
18	Arunachal Pradesh	Itanagar	3	5	67	
19	Arunachal Pradesh	Naharlagun	27	6	54	15
20	Assam	Bongaigaon	4	11	40	
21	Assam	Daranga	6	13	54	
22	Assam	Dibrugarh	6	11	39	
23	Assam	Golaghat	6	12	50	
24	Assam	Guwahati	26	11	114	60
25	Assam	Magherita	6	11	41	
26	Assam	Nagaon	6	14	101	
27	Assam	Nalbari	6	14	92	42
28	Assam	North Lakhimpur	6	14	57	
29	Assam	Silcher	8	9	46	37

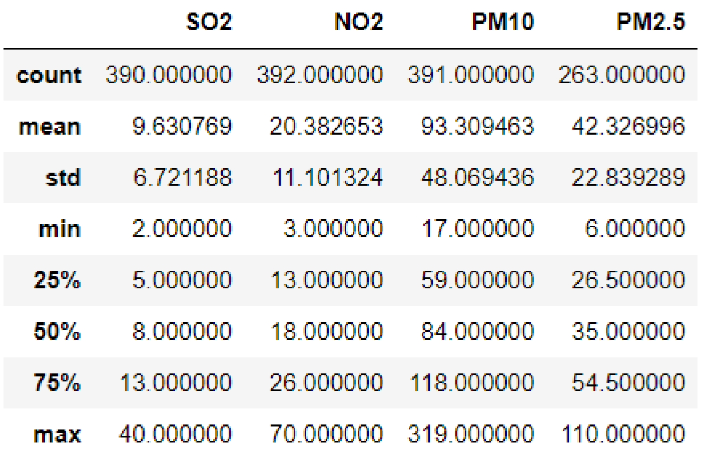
Simulation



S	T	U	V	W	X	Y	Z	AA	AB	AC
State_UT	AQI	Standardized /	Relative Frequ	Cumulative	Interval	Random Nu	State_RD	Simulated	Error	Error2
Andhra Pradesh	62.9375	0.11235204	0.012141494	0.012141	0-0.01	0.8587934	uttar pradesh	183.552	120.6145	14547.86
Arunachal Pradesh	60.5	0.10118366	0.010934565	0.023076	0.01-0.02	0.7057451	punjab	101.3102	40.81019	1665.471
Assam	82.95726496	0.20408063	0.022054283	0.04513	0.02-0.03	0.1956956	delhi	256.6667	173.7094	30174.96
Bihar	132.1666667	0.42955327	0.046420325	0.091551	0.03-0.08	0.1617134	delhi	256.6667	124.5	15500.25
Chandigarh (UT)	98	0.27300496	0.029502695	0.121053	0.08-0.11	0.3700579	himachal pradesh	70.09091	-27.9091	778.9174
Chattisgarh	64	0.11722031	0.012667591	0.133721	0.11-0.12	0.0367279	bihar	132.1667	68.16667	4646.694
Dadara & Nagar	84	0.20885834	0.022570593	0.156292	0.13-0.15	0.1360392	dadar and nagar	84	0	0
Delhi (UT)	256.6666667	1	0.108066516	0.264358	0.15-0.26	0.9576487	uttarakhand	144.8889	-111.778	12494.27
Goa	61.5625	0.10605193	0.011460662	0.275819	0.26-0.27	0.490089	jharkhand	251.75	190.1875	36171.29
Gujarat	109.7575741	0.32687701	0.03532446	0.311143	0.27-0.31	0.6353815	meghalaya	66.09524	-43.6623	1906.4
Haryana	160.7222222	0.56039201	0.060559612	0.371703	0.31-0.37	0.4601496	jharkhand	251.75	91.02778	8286.056
Himachal Pradesh	70.09090909	0.14512826	0.015683505	0.387386	0.37-0.38	0.6568512	mizoram	38.41667	-31.6742	1003.258
Jammu & Kashmir (U	121.2222222	0.3794069	0.041001182	0.428387	0.38-0.42	0.3202352	Haryana	160.7222	39.5	1560.25
Jharkhand	251.75	0.97747232	0.105632028	0.53402	0.42-0.53	0.3881291	jammu and kashm	121.2222	-130.528	17037.5
Karnataka	54.42564103	0.07335154	0.007926846	0.541946	0.53-0.54	0.8806504	uttar pradesh	183.552	129.1264	16673.62
Kerala	52.52813853	0.06465737	0.006987297	0.548934	0.54-0.56	0.3618196	Haryana	160.7222	108.1941	11705.96
Madhya Pradesh	90.84210526	0.2402082	0.025958463	0.574892	0.56-057	0.8346047	tripura	83.33333	-7.50877	56.38166
Maharashtra	88.2	0.22810233	0.024650224	0.599542	0.57-0.59	0.6581463	mizoram	38.41667	-49.7833	2478.38
Manipur	115.3333333	0.35242459	0.038085298	0.637628	0.59-0.63	0.688586	odisha	92.12605	-23.2073	538.578
Meghalaya	66.0952381	0.12682049	0.013705048	0.651333	0.63-0.65	0.4246634	jharkhand	251.75	185.6548	34467.69
Mizoram	38.41666667	0	0	0.651333	0.65-0.665	0.1591505	delhi	256.6667	218.25	47633.06
Nagaland	82.5	0.20198549	0.021827868	0.673161	0.665-0.67	0.2222712	delhi	256.6667	174.1667	30334.03
Odisha	92.12605042	0.24609111	0.026594208	0.699755	0.67-0.69	0.3333781	haryana	160.7222	68.59617	4705.435
Pondicherry (UT)	41.5	0.01412753	0.001526713	0.701281	0.69-0.70	0.5147095	jharkhand	251.75	210.25	44205.06
Punjab	101.3101852	0.28817191	0.031141734	0.732423	0.70-0.73	0.5818781	maharashtra	88.2	-13.1102	171.877
Rajasthan	135.7333333	0.44589538	0.04818636	0.78061	0.73-0.78	0.0013271	andhra pradesh	62.9375	-72.7958	5299.233
Sikkim	53.20833333	0.06777396	0.007324096	0.787934	0.78-0.785	0.4982652	jharkhand	251.75	198.5417	39418.79
Tamilnadu	49.92156863	0.05271433	0.005696654	0.79363	0.785-0.79	0.7434688	rajasthan	135.7333	85.81176	7363.659
Tamilnadu	49.92156863	0.05271433	0.005696654	0.79363	0.785-0.79	0.7434688	rajasthan	135.7333	85.81176	7363.659
Telangana	94.62770563	0.25755344	0.027832903	0.821463	0.79-0.82	0.6774335	odisha	92.12605	-2.50166	6.258279
Tripura	83.33333333	0.20580374	0.022240493	0.843704	0.82-0.84	0.4613872	jharkhand	251.75	168.4167	28364.17
Uttar Pradesh	183.5520282	0.66499593	0.071863793	0.915568	0.84-0.91	0.9303625	uttarakhand	144.8889	-38.6631	1494.838
Uttarakhand	144.8888889	0.48784523	0.052719735	0.968287	0.91-0.96	0.9037526	uttar pradesh	183.552	38.66314	1494.838
West Bengal	102.4634146	0.29345589	0.031712756	1	0.96-1	0.5601665	madhya pradesh	90.84211	-11.6213	135.0548
		9.25356009	1							422320.1
MIN	38.41666667									12797.58
									RMSE=	113.1264
MAX	256.6666667									

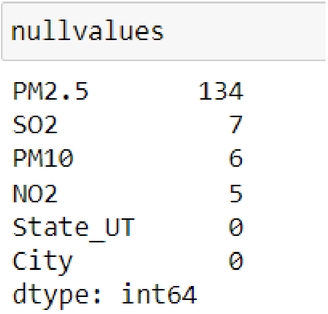
7. PROJECT OUTPUTS

TABLE 7.1 Descriptives



	SO2	NO2	PM10	PM2.5
count	390.000000	392.000000	391.000000	263.000000
mean	9.630769	20.382653	93.309463	42.326996
std	6.721188	11.101324	48.069436	22.839289
min	2.000000	3.000000	17.000000	6.000000
25%	5.000000	13.000000	59.000000	26.500000
50%	8.000000	18.000000	84.000000	35.000000
75%	13.000000	26.000000	118.000000	54.500000
max	40.000000	70.000000	319.000000	110.000000

TABLE 7.2 Missing Values Count



```
nullvalues
```

```
PM2.5       134
SO2           7
PM10          6
NO2           5
State_UT      0
City          0
dtype: int64
```

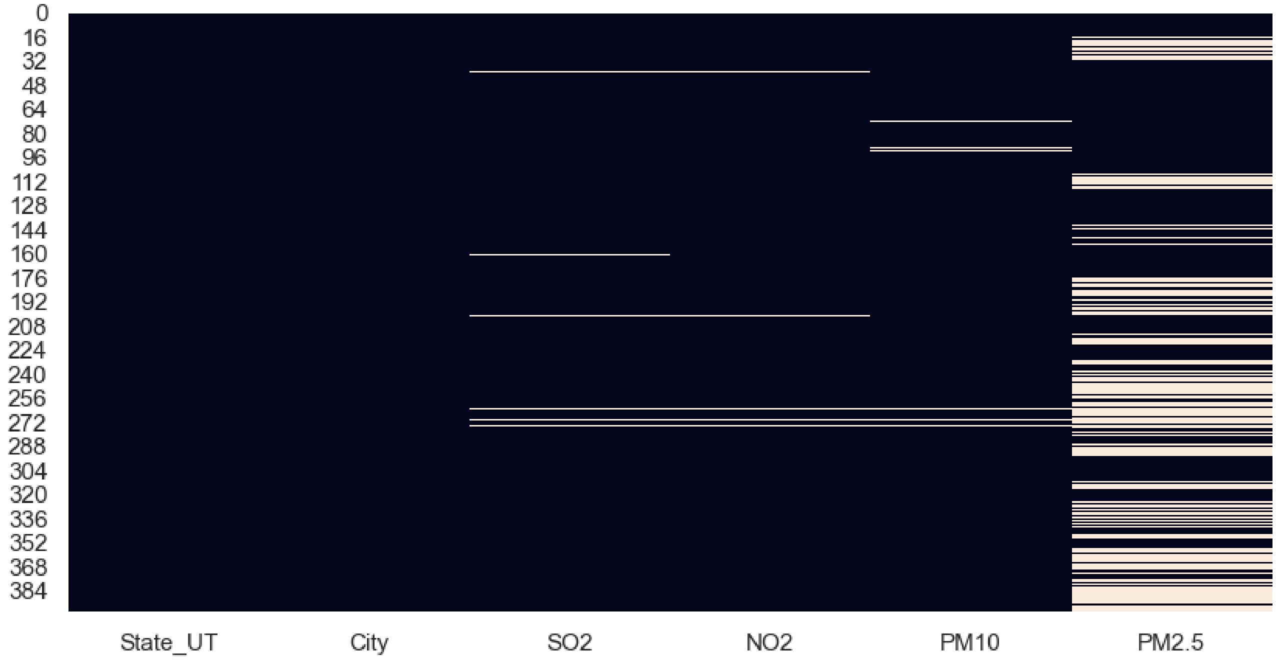


Fig7.1 Heat Map for Missing Values

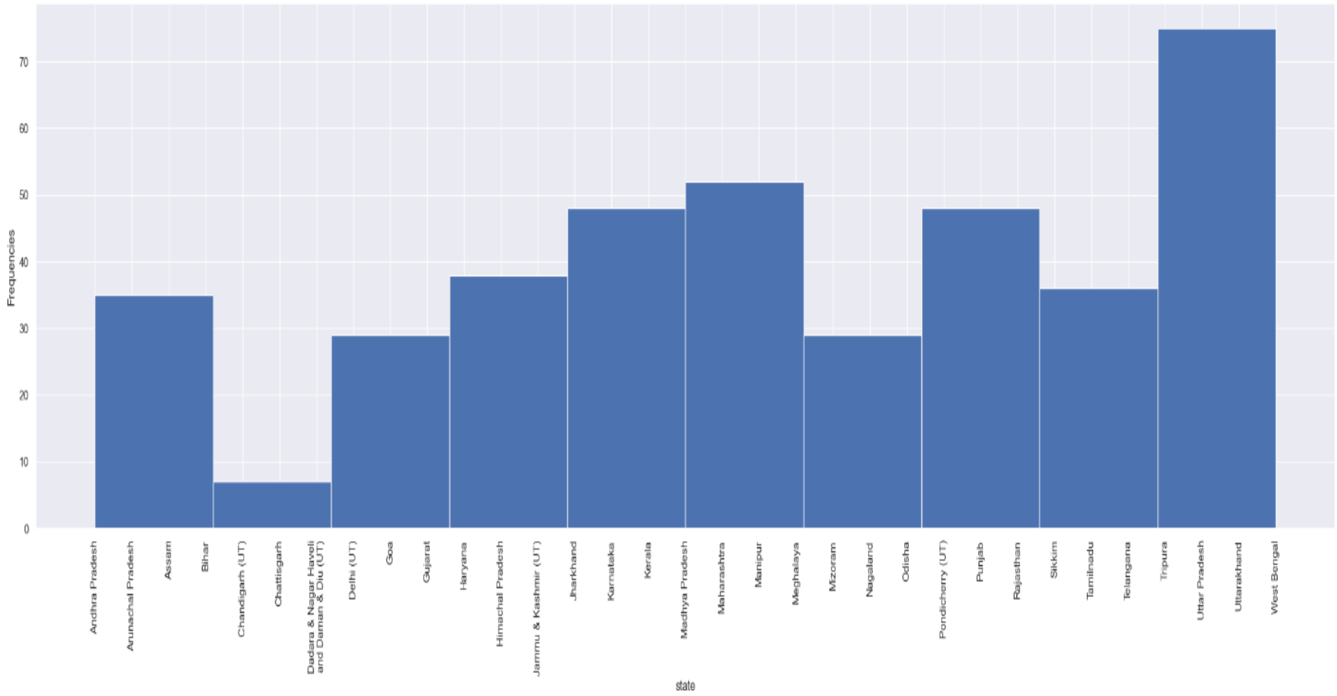


Fig 7.2 The visualization shows us the count of states present in the dataset

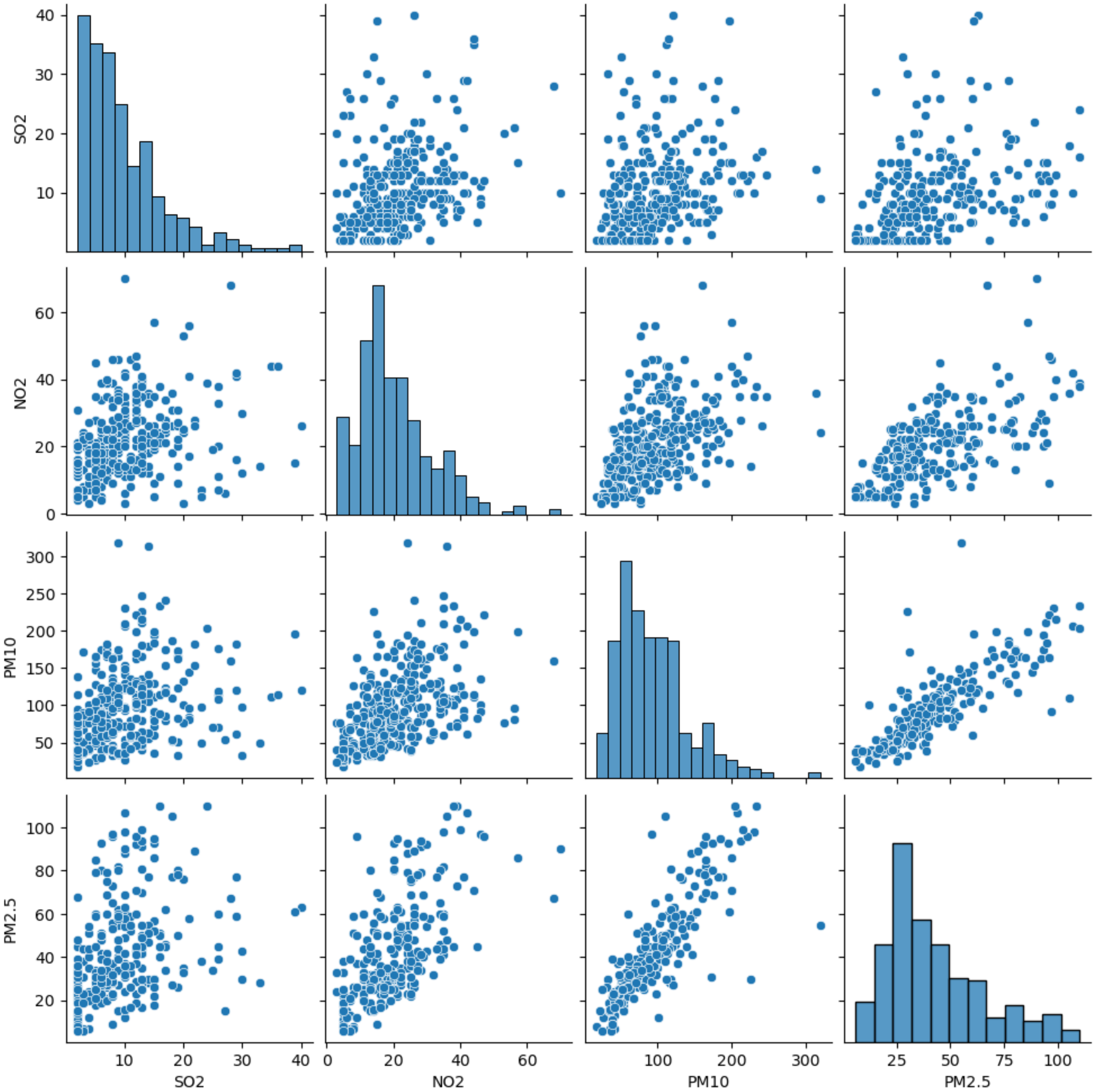


Fig 7.3 Correlation Analysis

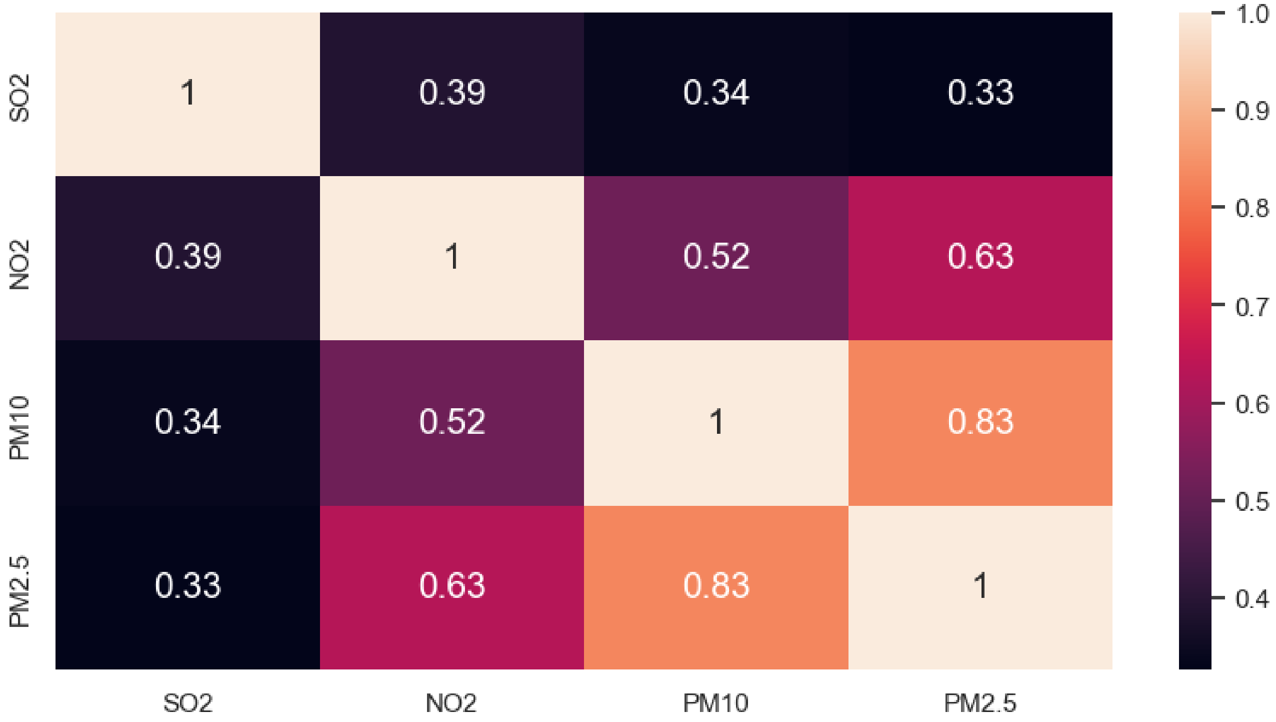


Fig 7.4 Correlation Heat Map(with in predictors)

This shows that PM10 and PM2.5 are Correlated as their Correclation Value is greater than 0.5

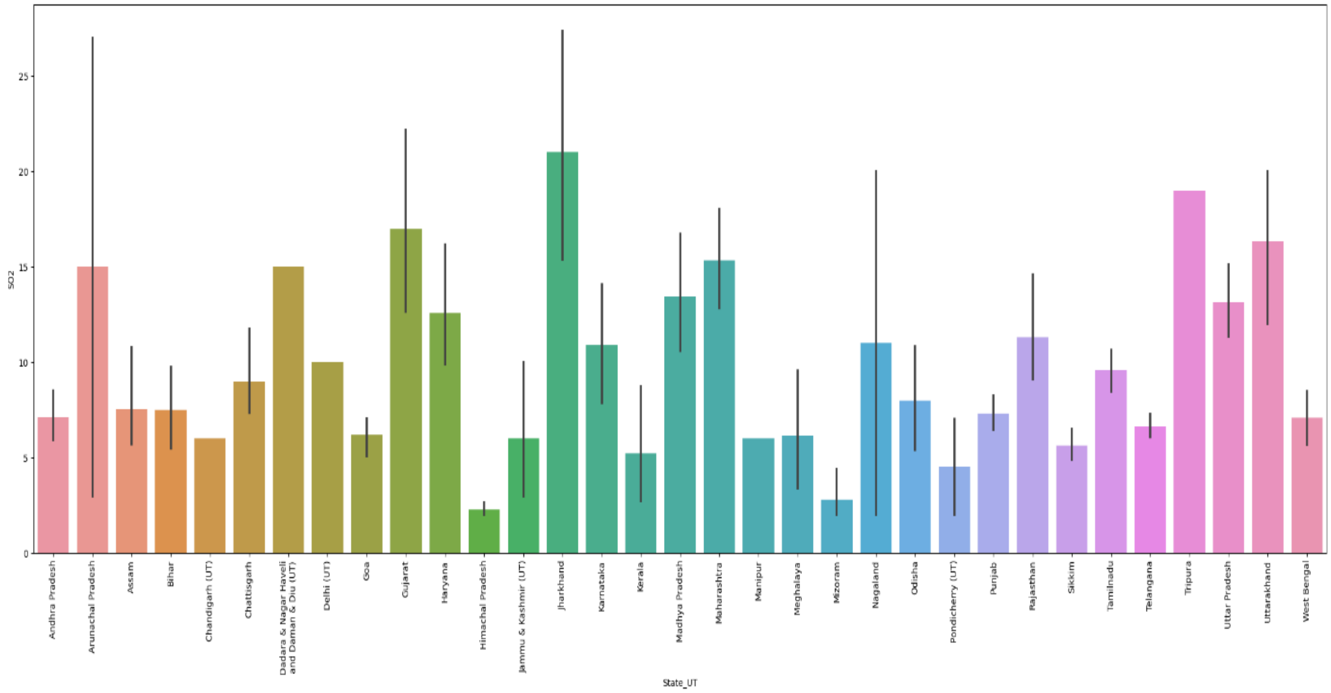


Fig 7.5 This visualization shows the name of the state having higher SO2 levels in the air which is Jharkhand followed by Tripura

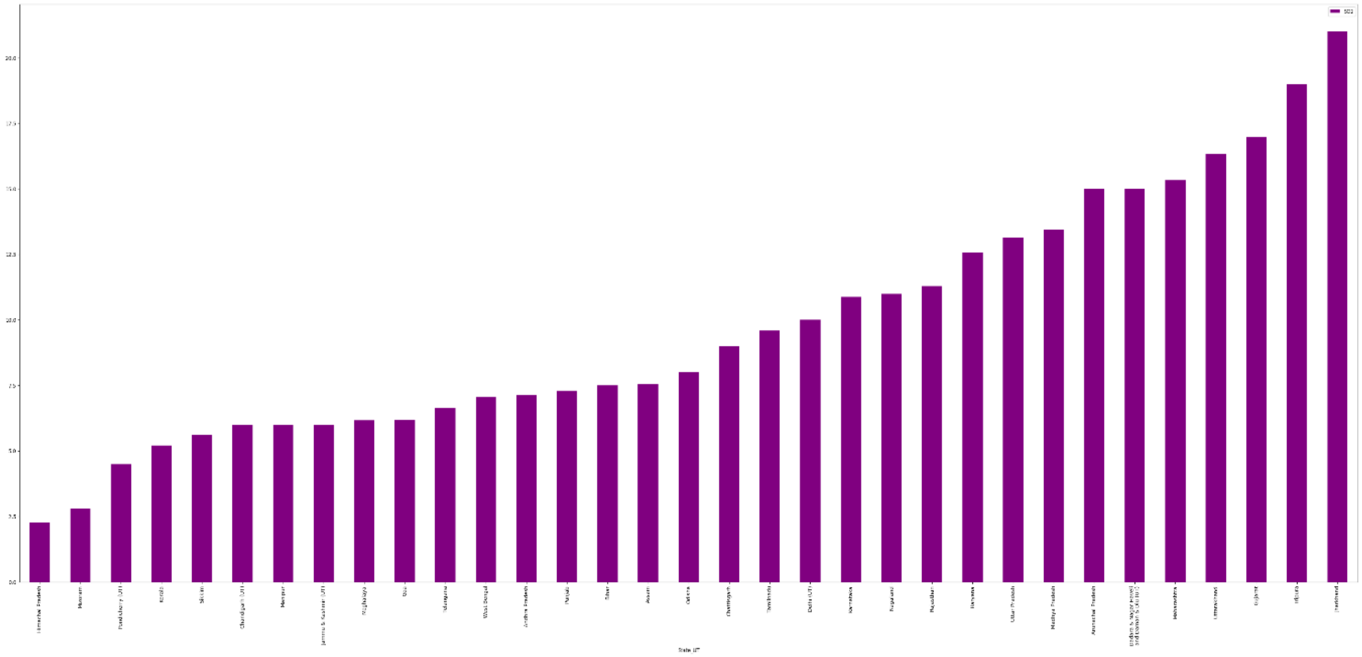


Fig7.6 states in an increasing order based on their SO2 levels.

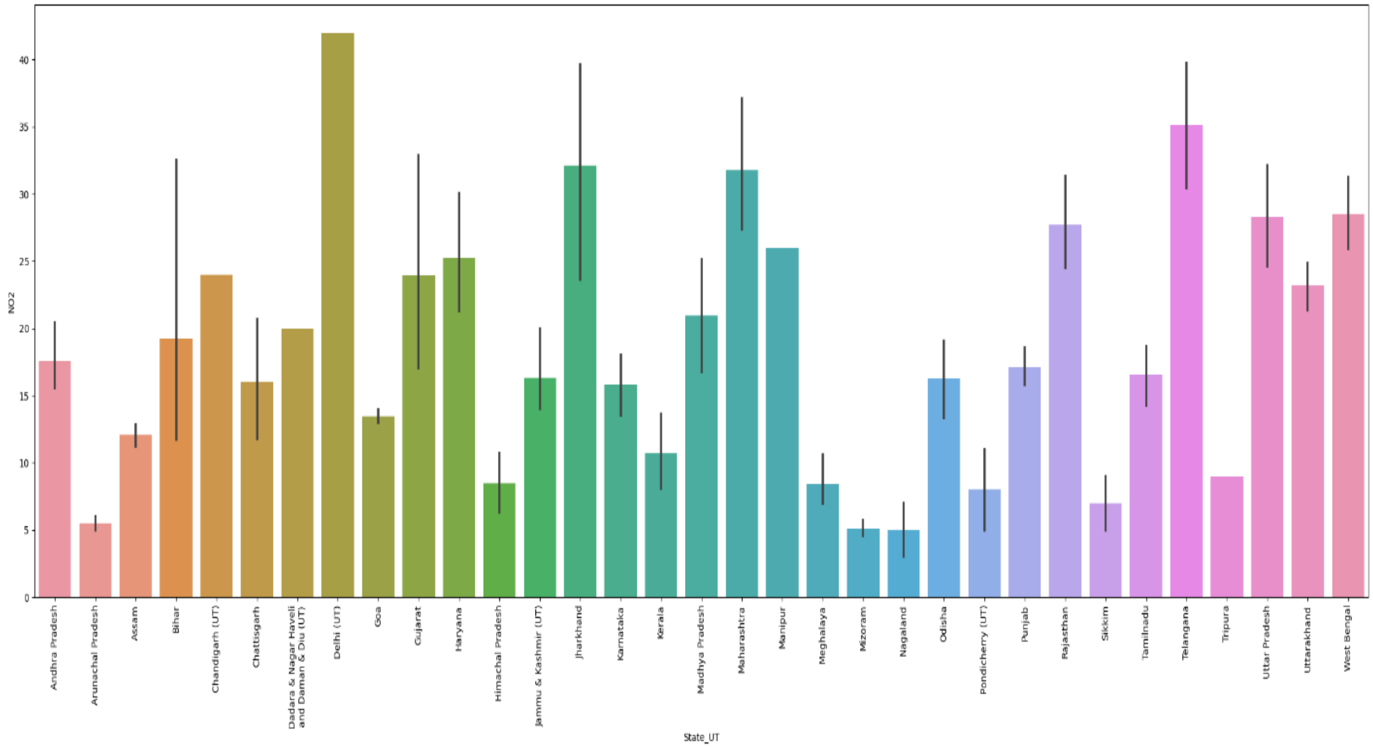


Fig 7.7 This visualization shows the name of the state having higher NO2 levels in the air which is Delhi followed by Telangana

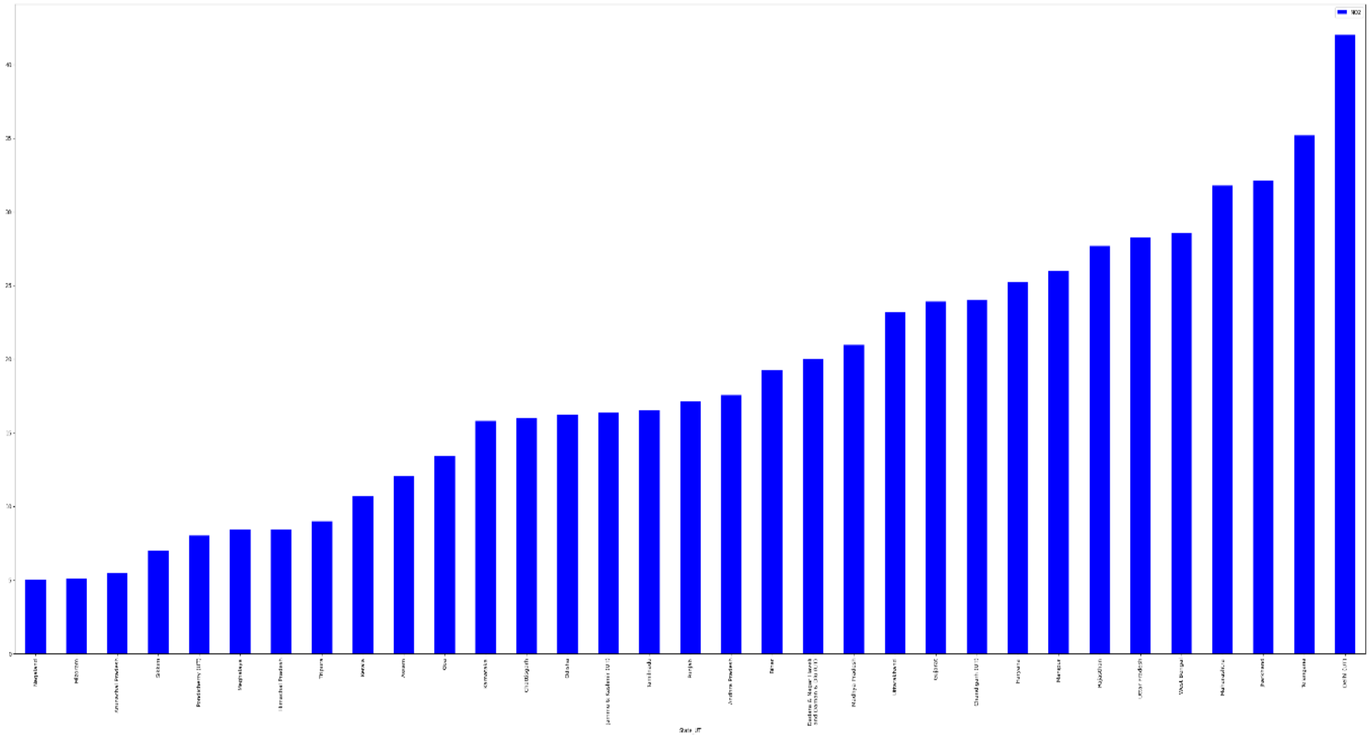


Fig7.8 states in an increasing order based on their NO2 levels.

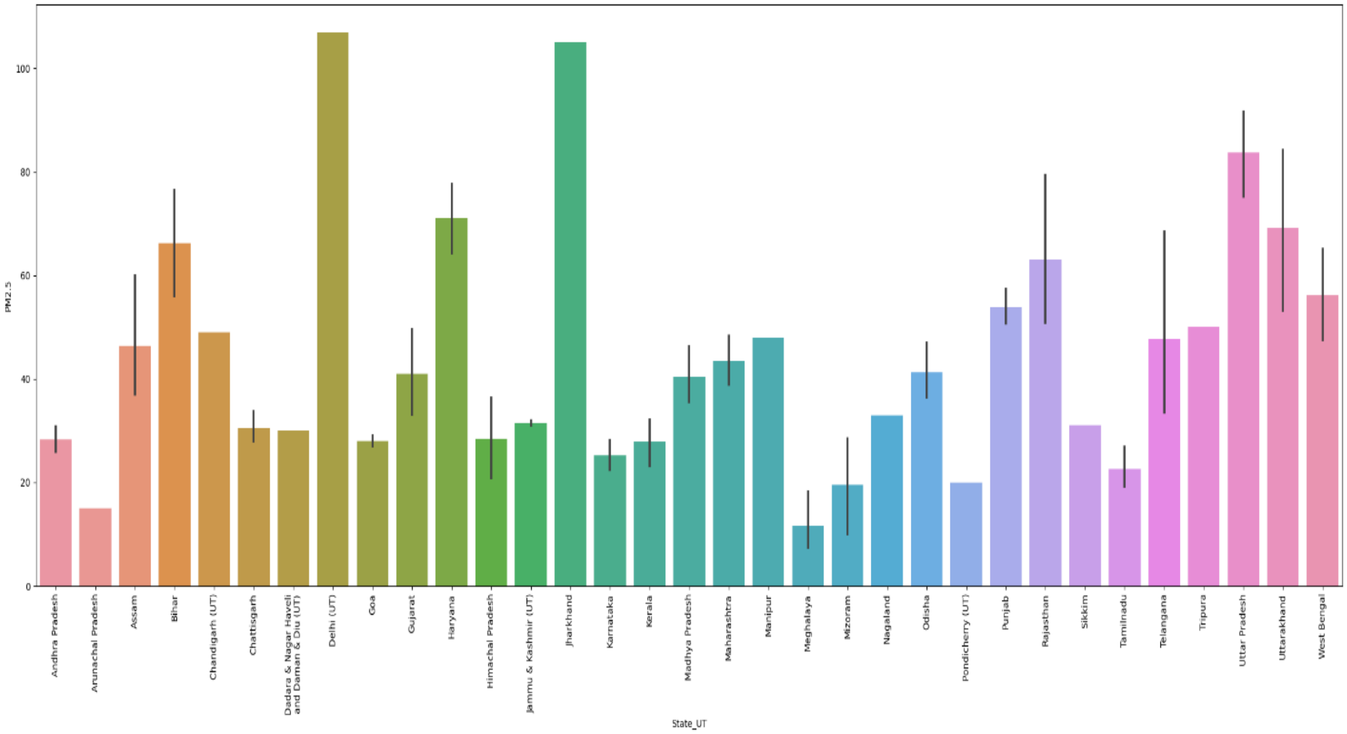


Fig 7.9 This visualization shows the name of the state having higher PM2.5 levels in the air which is Delhi followed by Jharkhand

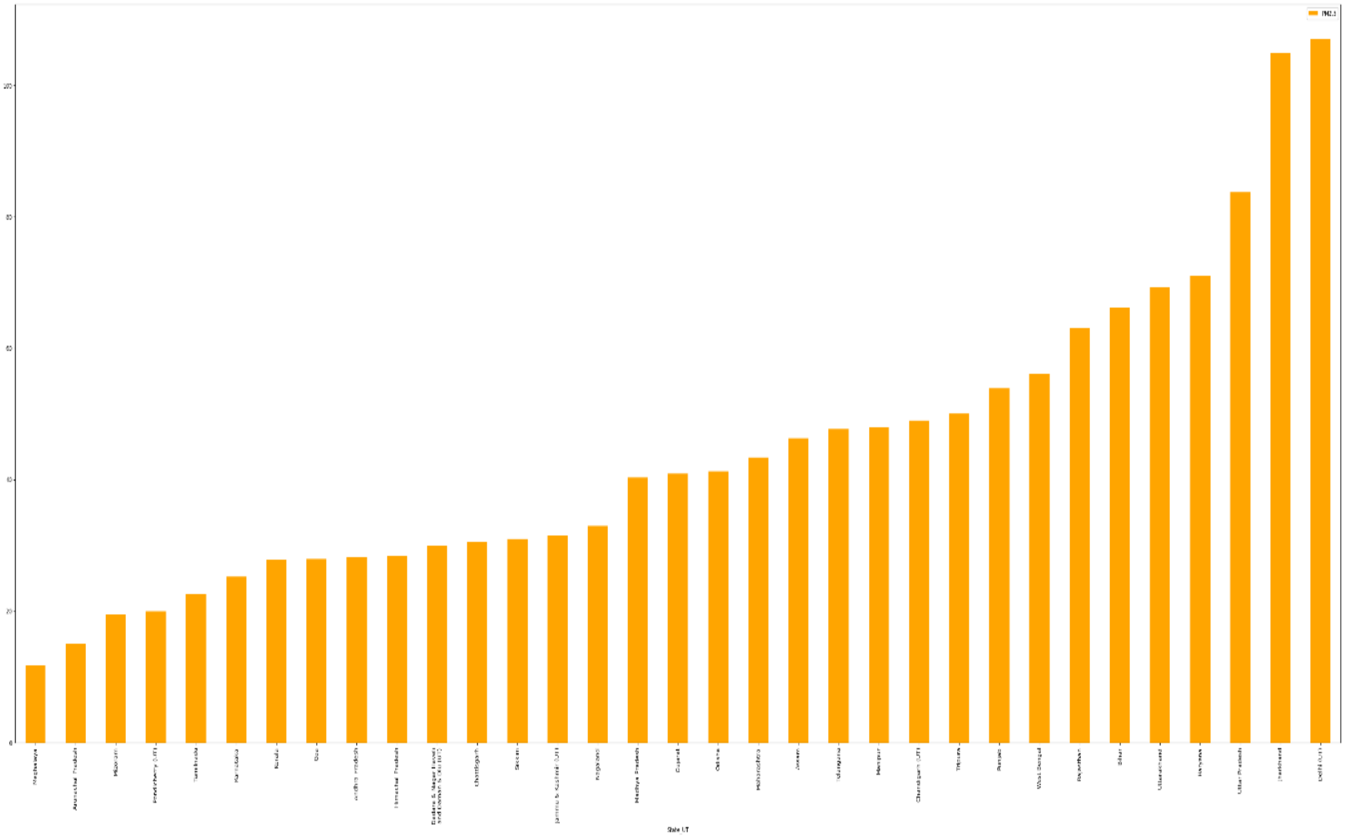


Fig7.10 States In An Increasing Order Based On Their PM2.5 Levels.

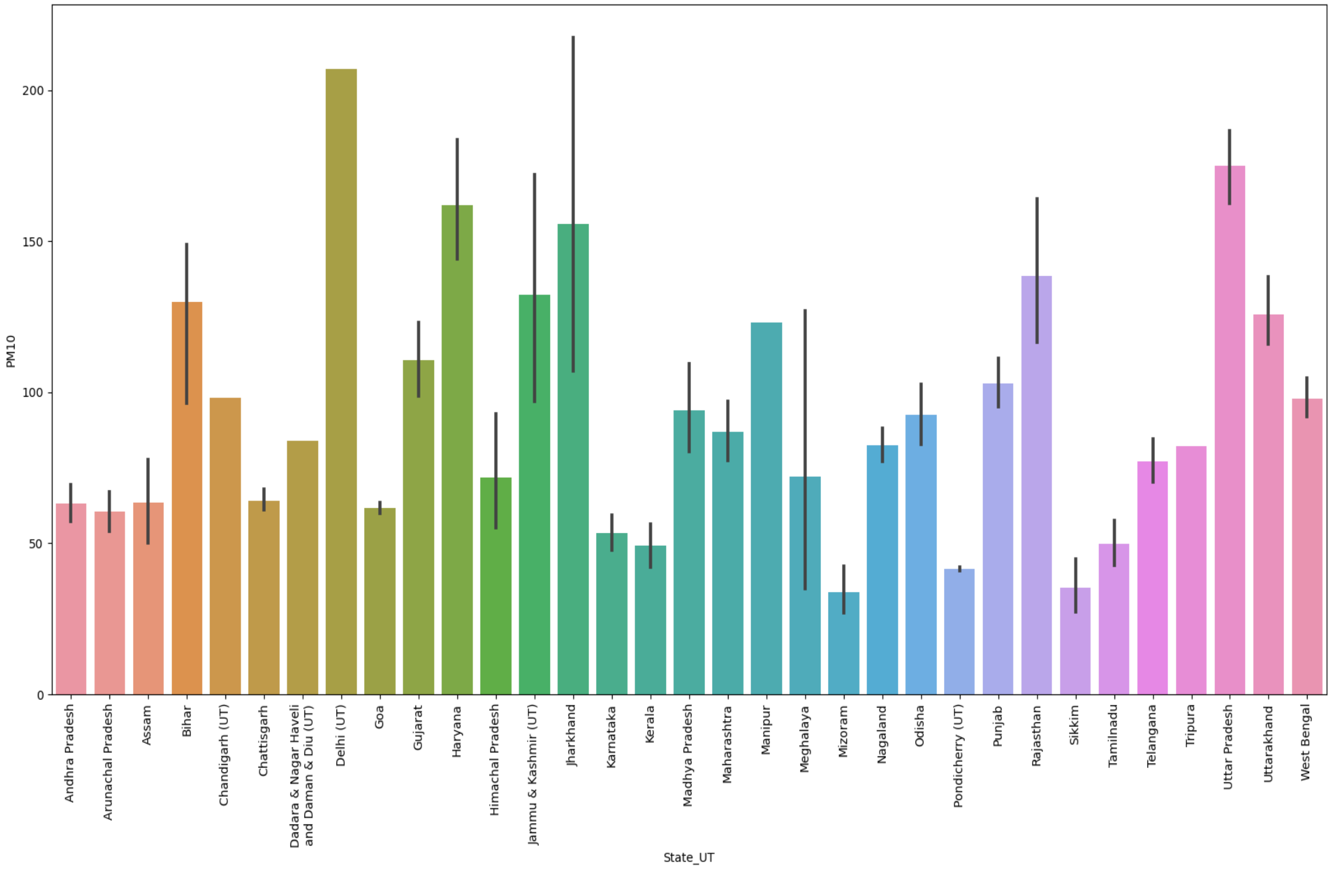


Fig7.11 This visualization shows the name of the state having higher PM10 levels in the air which is Delhi followed by Uttar Pradesh

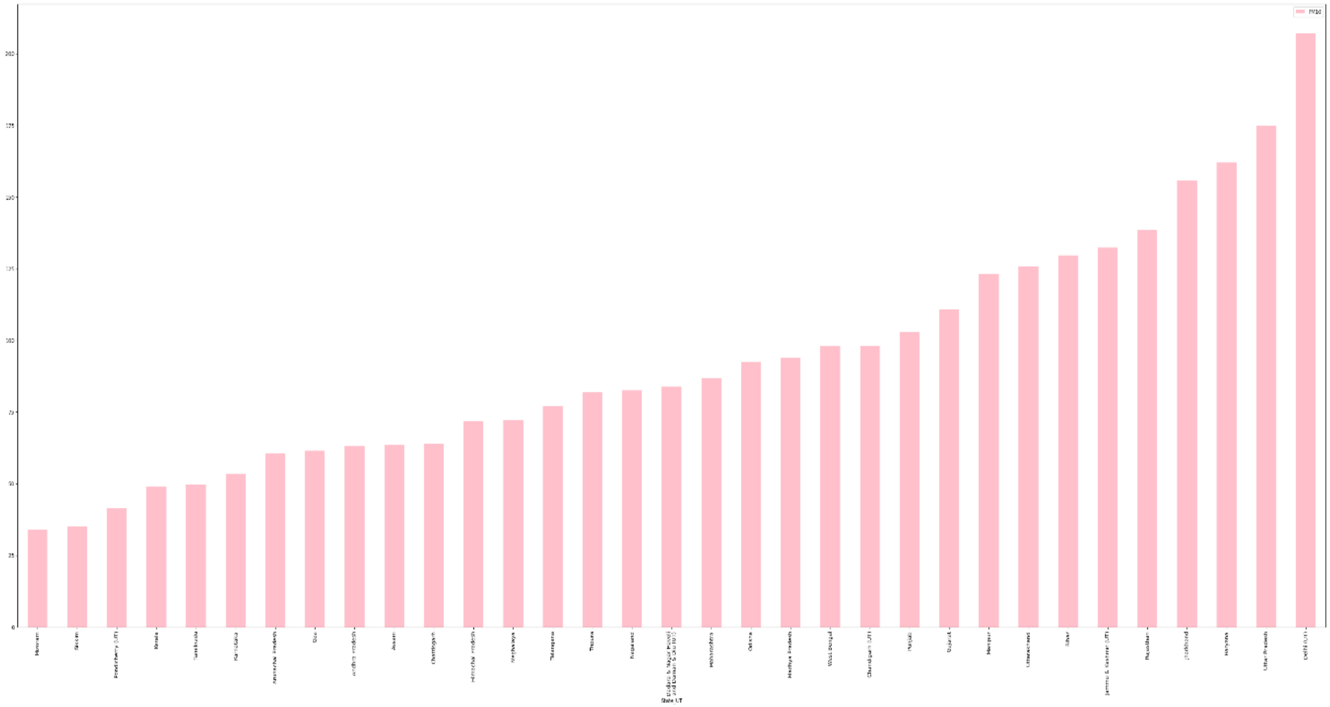


Fig7.12 States In An Increasing Order Based On Their PM10 Levels.

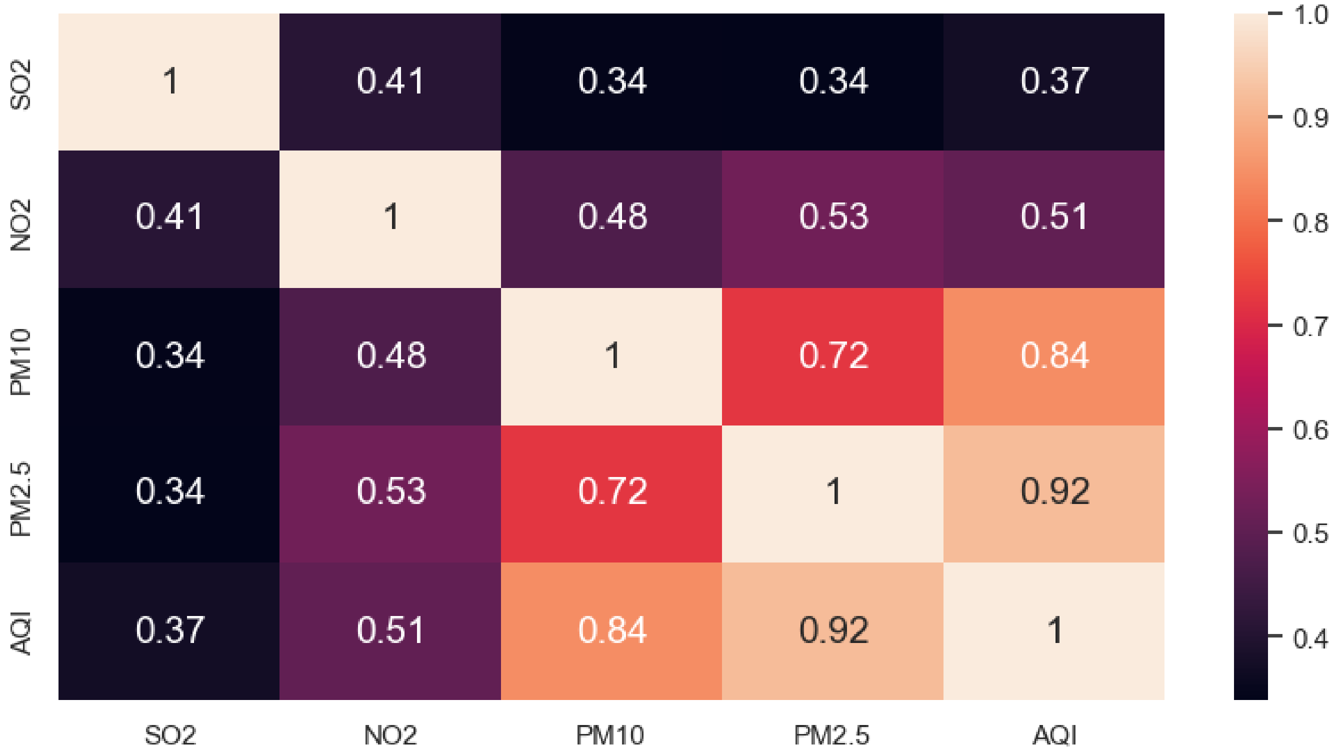
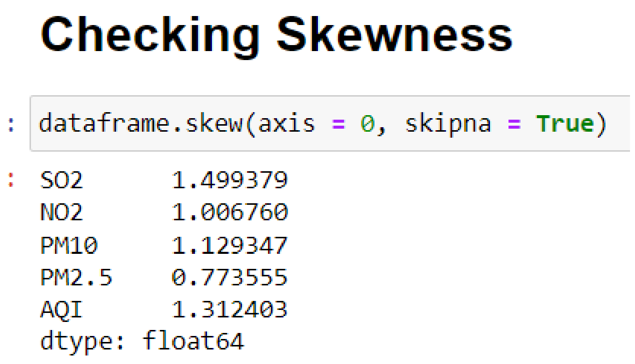


Fig 7.13 Correlation Heat Map between dependent variable i.e AQI and independent variable

This Shows that PM2.5 and PM10 has greater correlation and it indicates the presence of multicollinearity in our dataset as it will lead to dependency among explanatory variables

Table 7.3 Skewness



Checking Skewness

```
: dataframe.skew(axis = 0, skipna = True)
```

```
: SO2      1.499379
  NO2      1.006760
  PM10     1.129347
  PM2.5    0.773555
  AQI      1.312403
  dtype: float64
```

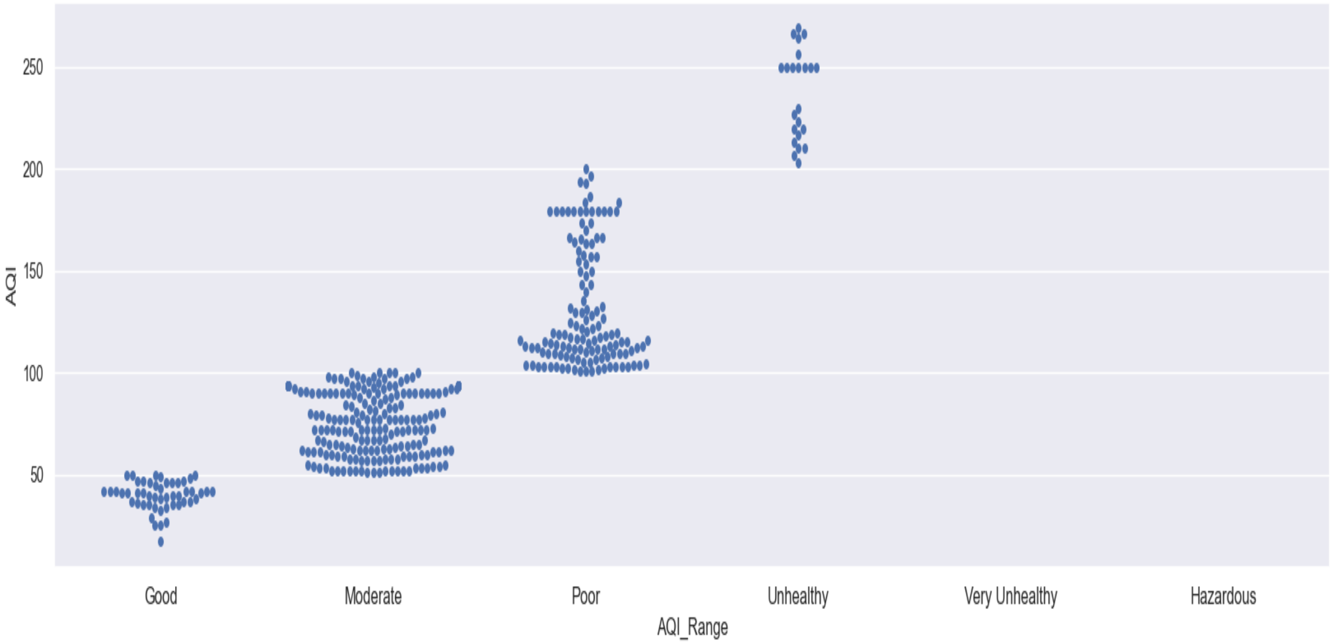


Fig 7.14 Swarm Plot of AQI Range

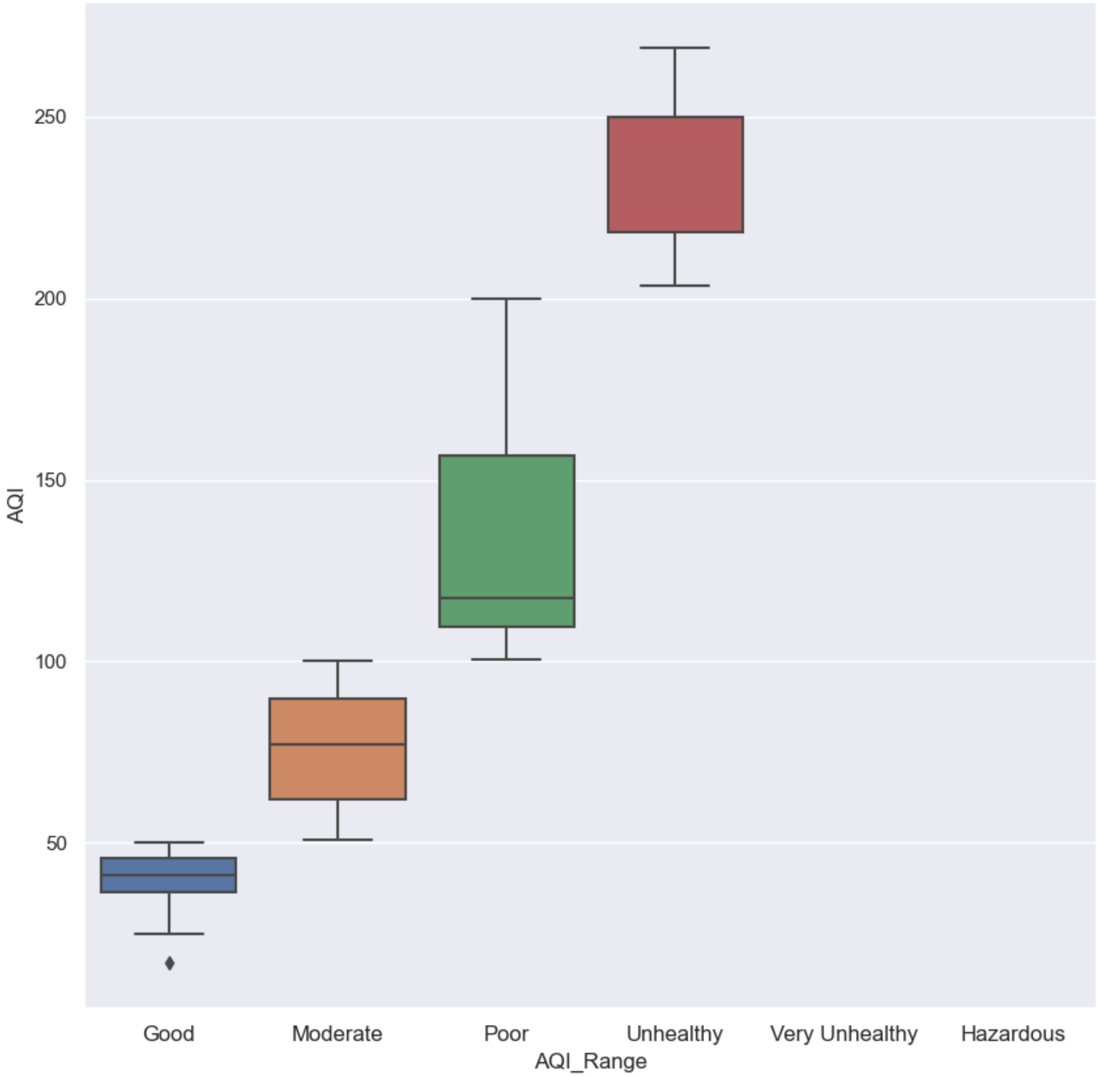


Fig7.15 Box Plot

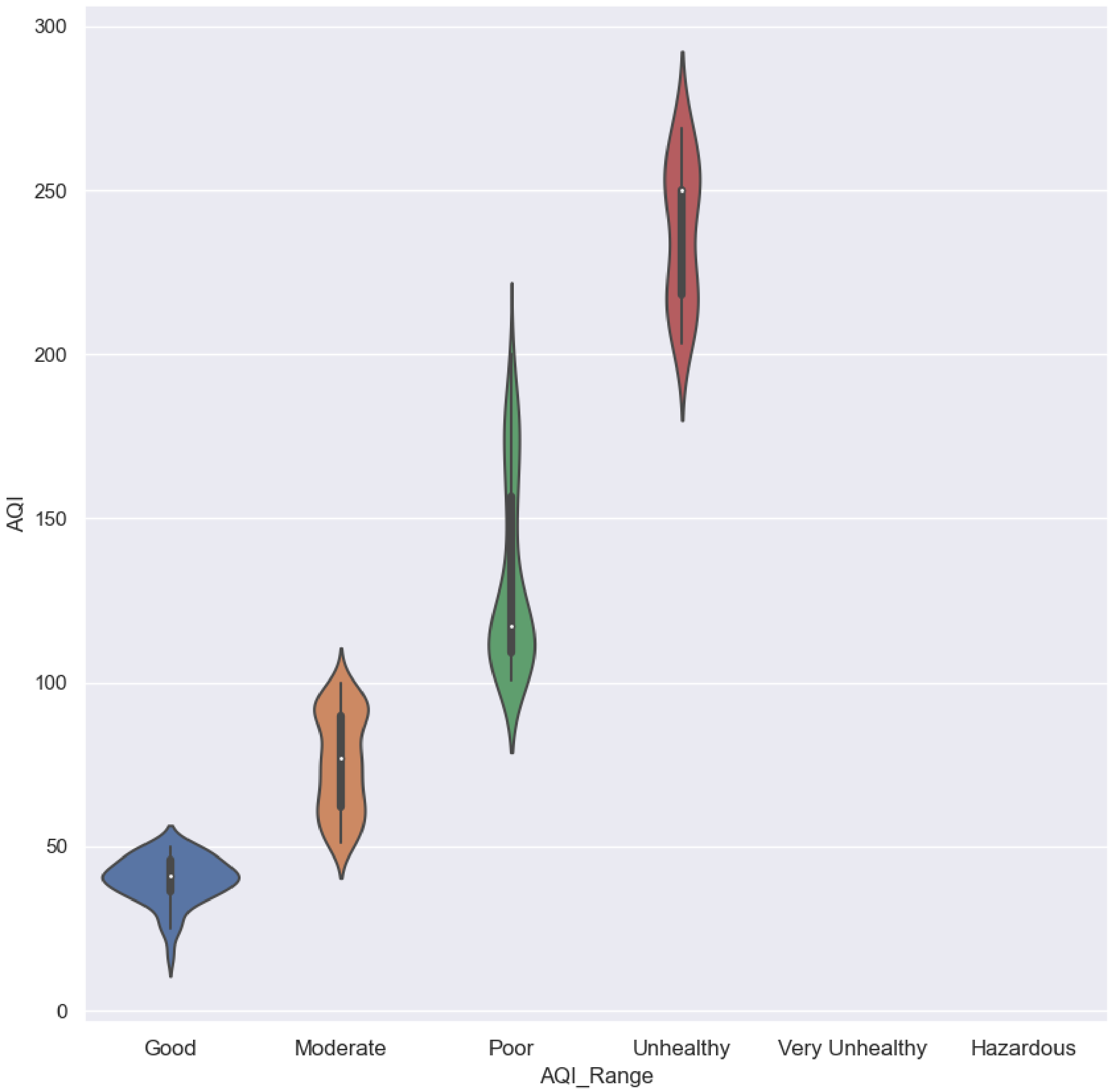


Fig7.16 Violin Plot

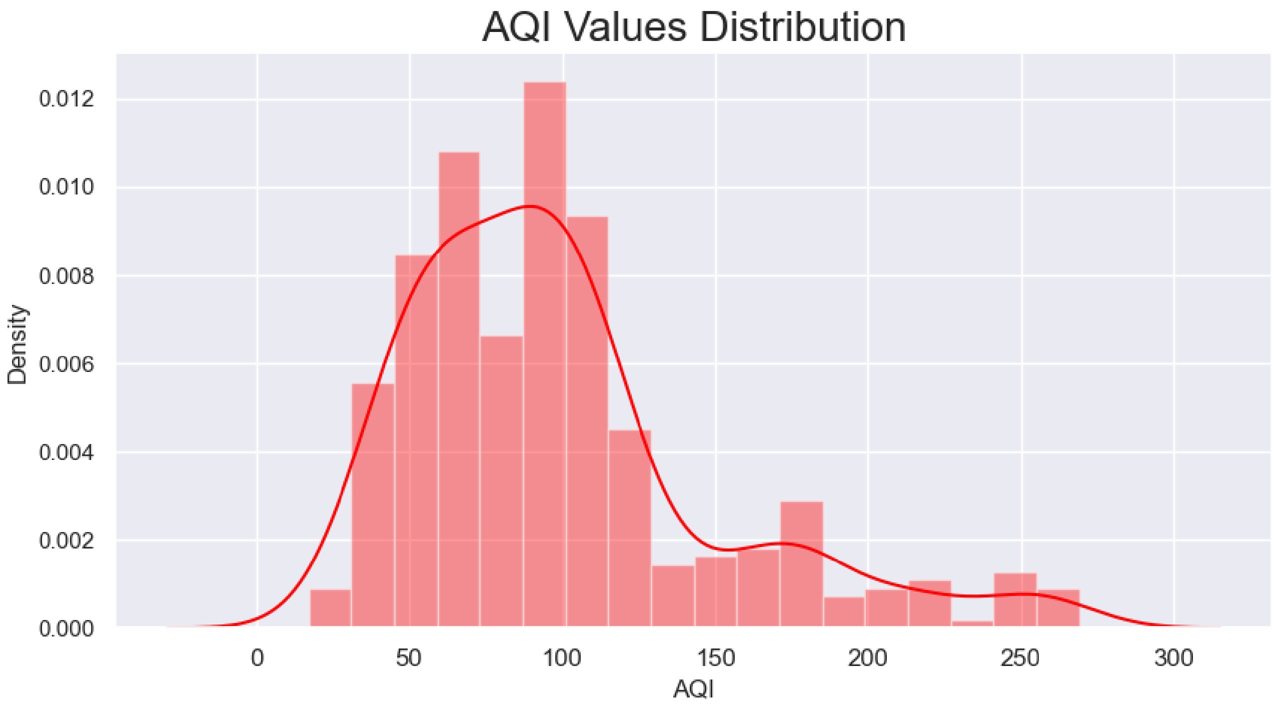


Fig 7.17 AQI Values Distribution

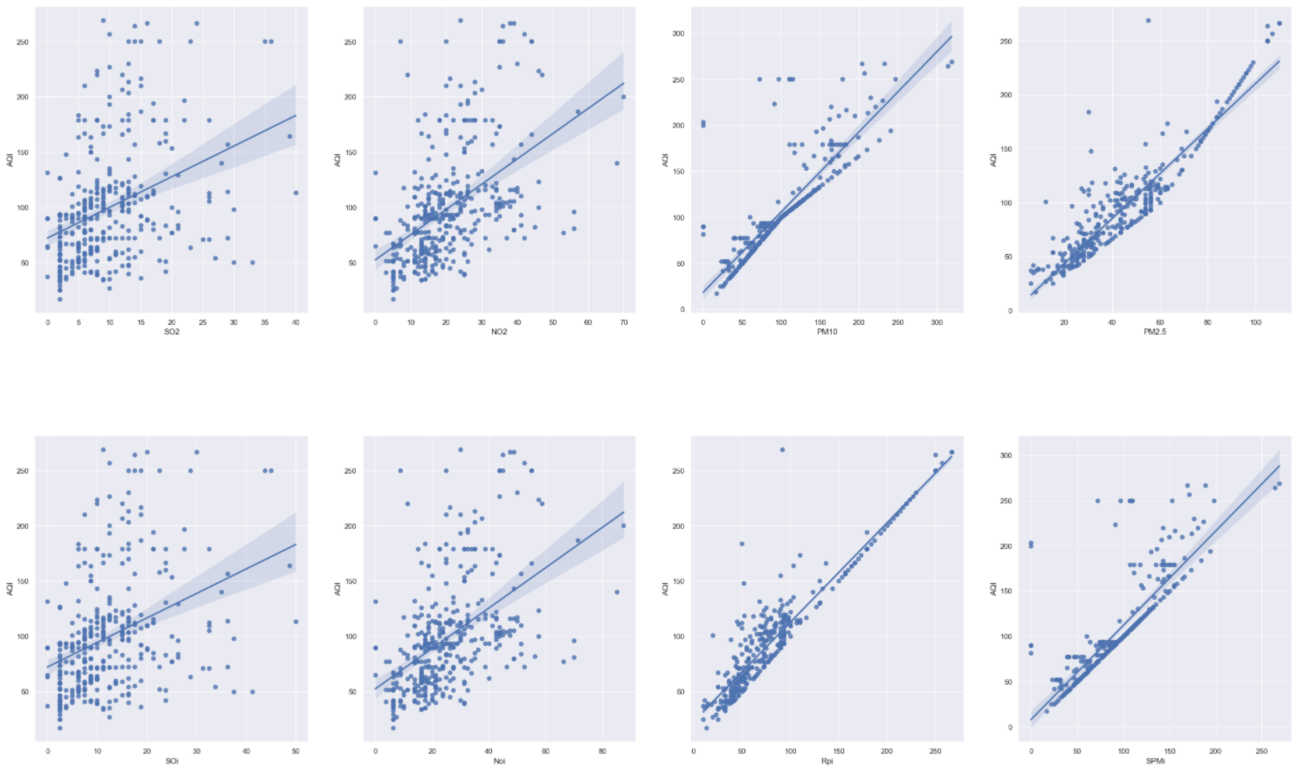
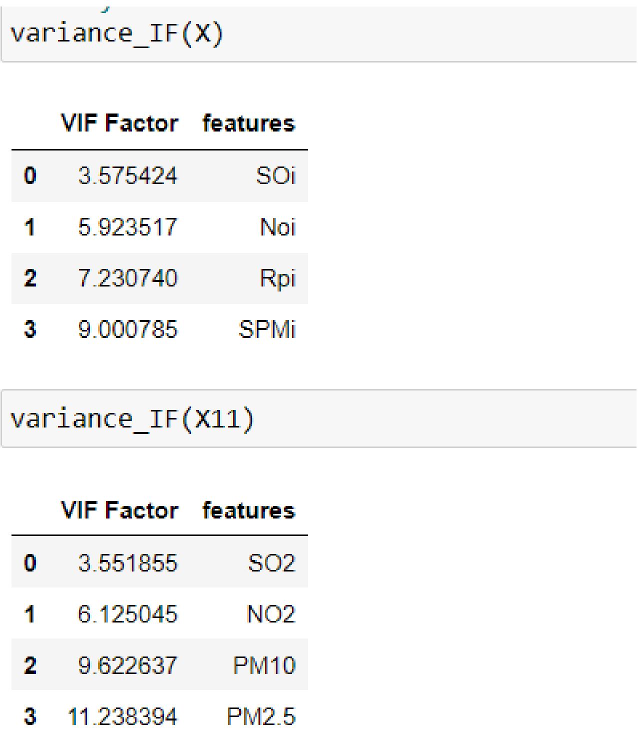


Fig 7.18 Distribution Of Correlated Factors

Distribution Of Correlated Factors- As AQI Is Dependent On The Subindex And In Correlation Analysis We Already Seen That AQI Is Highly Correlated With PM2.5 And PM10

Table 7.4 VIF for checking multicollinearity



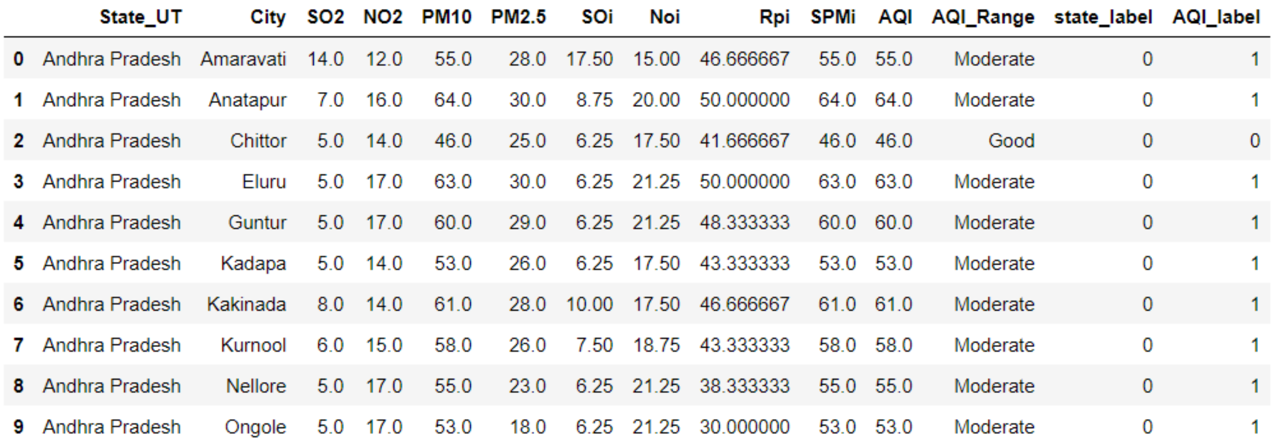
```
variance_IF(X)
```

	VIF Factor	features
0	3.575424	SOi
1	5.923517	Noi
2	7.230740	Rpi
3	9.000785	SPMi

```
variance_IF(X11)
```

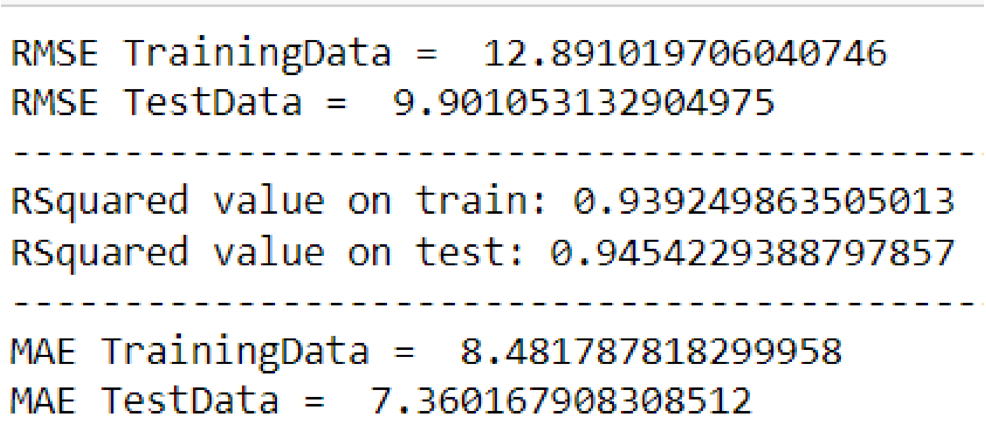
	VIF Factor	features
0	3.551855	SO2
1	6.125045	NO2
2	9.622637	PM10
3	11.238394	PM2.5

Table 7.5 DATASET after data preprocessing for model training



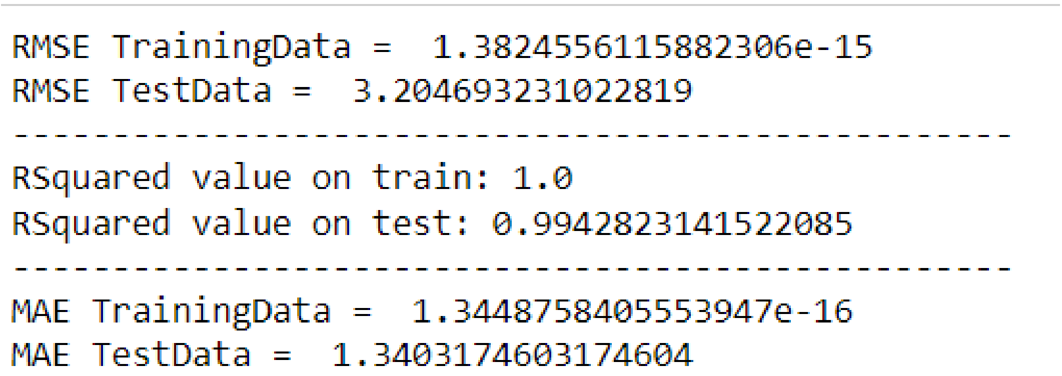
	State_UT	City	SO2	NO2	PM10	PM2.5	SOi	Noi	Rpi	SPMi	AQI	AQI_Range	state_label	AQI_label
0	Andhra Pradesh	Amaravati	14.0	12.0	55.0	28.0	17.50	15.00	46.666667	55.0	55.0	Moderate	0	1
1	Andhra Pradesh	Anatapur	7.0	16.0	64.0	30.0	8.75	20.00	50.000000	64.0	64.0	Moderate	0	1
2	Andhra Pradesh	Chittor	5.0	14.0	46.0	25.0	6.25	17.50	41.666667	46.0	46.0	Good	0	0
3	Andhra Pradesh	Eluru	5.0	17.0	63.0	30.0	6.25	21.25	50.000000	63.0	63.0	Moderate	0	1
4	Andhra Pradesh	Guntur	5.0	17.0	60.0	29.0	6.25	21.25	48.333333	60.0	60.0	Moderate	0	1
5	Andhra Pradesh	Kadapa	5.0	14.0	53.0	26.0	6.25	17.50	43.333333	53.0	53.0	Moderate	0	1
6	Andhra Pradesh	Kakinada	8.0	14.0	61.0	28.0	10.00	17.50	46.666667	61.0	61.0	Moderate	0	1
7	Andhra Pradesh	Kurnool	6.0	15.0	58.0	26.0	7.50	18.75	43.333333	58.0	58.0	Moderate	0	1
8	Andhra Pradesh	Nellore	5.0	17.0	55.0	23.0	6.25	21.25	38.333333	55.0	55.0	Moderate	0	1
9	Andhra Pradesh	Ongole	5.0	17.0	53.0	18.0	6.25	21.25	30.000000	53.0	53.0	Moderate	0	1

Table 7.6 Error metrics for Linear Regression



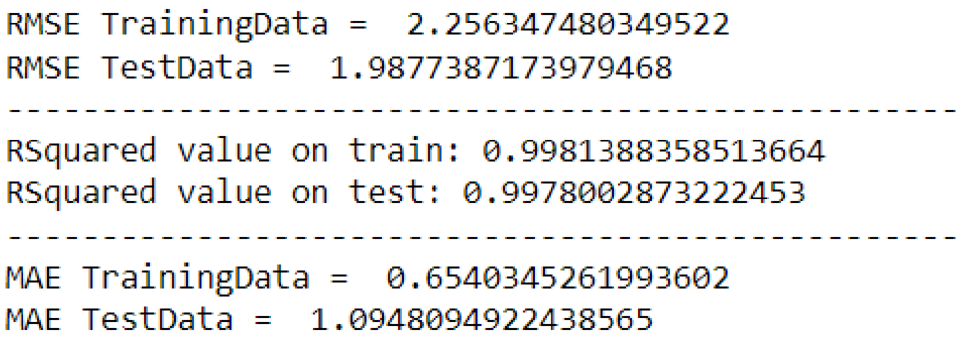
```
RMSE TrainingData =  12.891019706040746
RMSE TestData =  9.901053132904975
-----------------------------------------
RSquared value on train: 0.939249863505013
RSquared value on test: 0.9454229388797857
-----------------------------------------
MAE TrainingData =  8.481787818299958
MAE TestData =  7.360167908308512
```

Table 7.7 Error metrics for Decision Tree Regressor



```
RMSE TrainingData =  1.3824556115882306e-15
RMSE TestData =  3.204693231022819
-----------------------------------------------
RSquared value on train: 1.0
RSquared value on test: 0.9942823141522085
-----------------------------------------------
MAE TrainingData =  1.3448758405553947e-16
MAE TestData =  1.3403174603174604
```

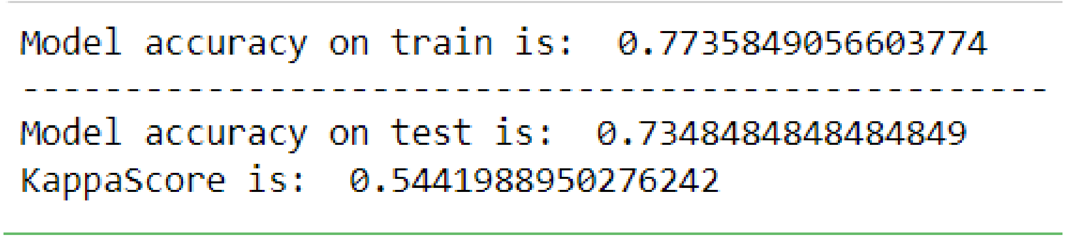
Table 7.8 Error metrics for Random Forest Regressor



```
RMSE TrainingData =  2.256347480349522
RMSE TestData =  1.9877387173979468
-------------------------------------------------
RSquared value on train: 0.9981388358513664
RSquared value on test: 0.9978002873222453
-------------------------------------------------
MAE TrainingData =  0.6540345261993602
MAE TestData =  1.0948094922438565
```

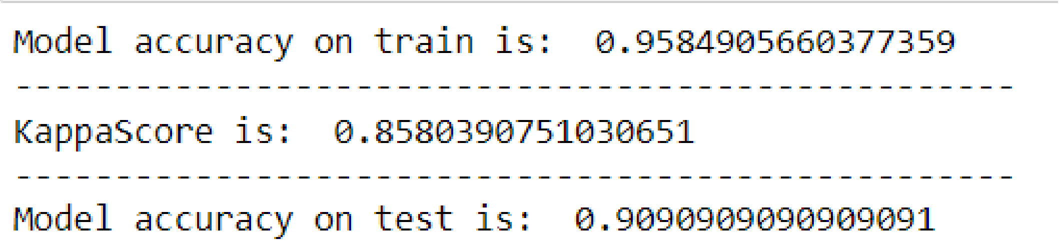
Table 7.9 Performance metrics for classification algoritthms

Logistic regression



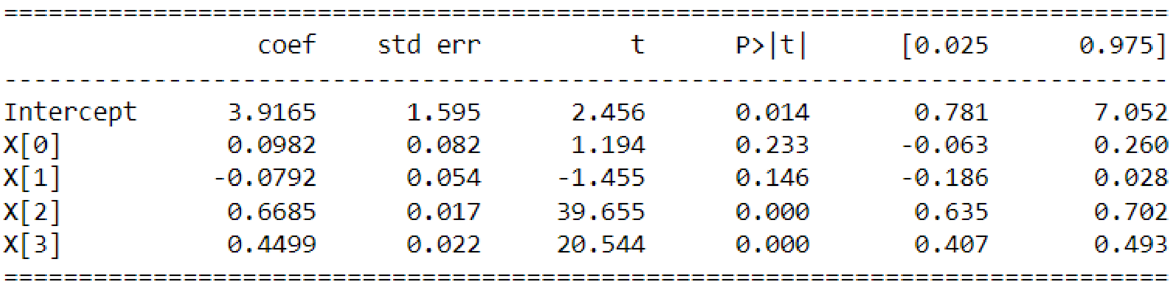
```
Model accuracy on train is:  0.7735849056603774
-----------------------------------------------
Model accuracy on test is:  0.7348484848484849
KappaScore is:  0.5441988950276242
```

K-Nearest Neighbours

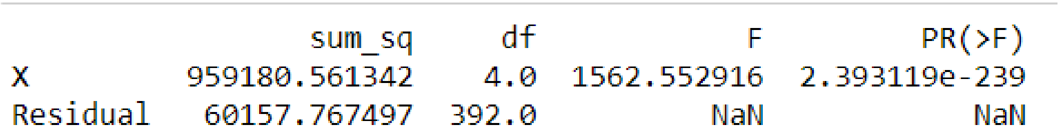


```
Model accuracy on train is:  0.9584905660377359
-----------------------------------------------
KappaScore is:  0.8580390751030651
-----------------------------------------------
Model accuracy on test is:  0.9090909090909091
```

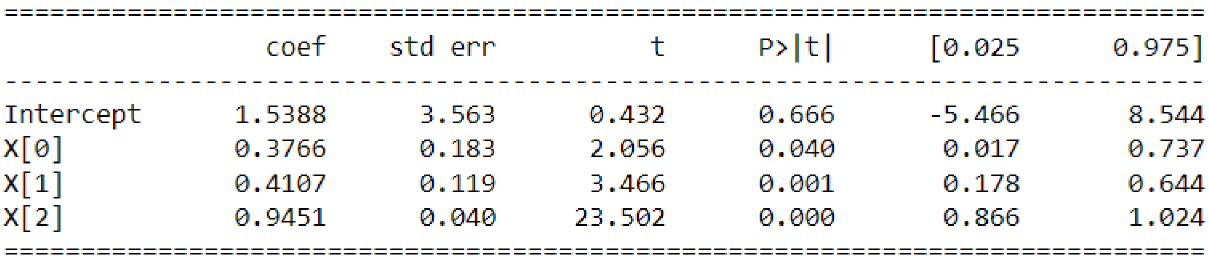
Table 7.10 ANOVA ANALYSIS



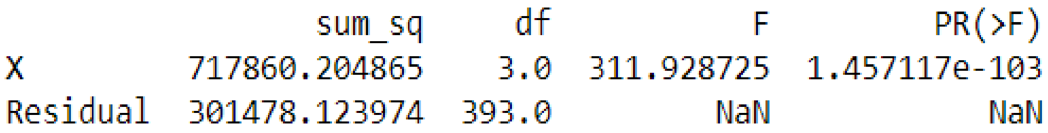
```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      3.9165      1.595      2.456      0.014       0.781       7.052
X[0]           0.0982      0.082      1.194      0.233      -0.063       0.260
X[1]          -0.0792      0.054     -1.455      0.146      -0.186       0.028
X[2]           0.6685      0.017     39.655      0.000       0.635       0.702
X[3]           0.4499      0.022     20.544      0.000       0.407       0.493
==============================================================================
```



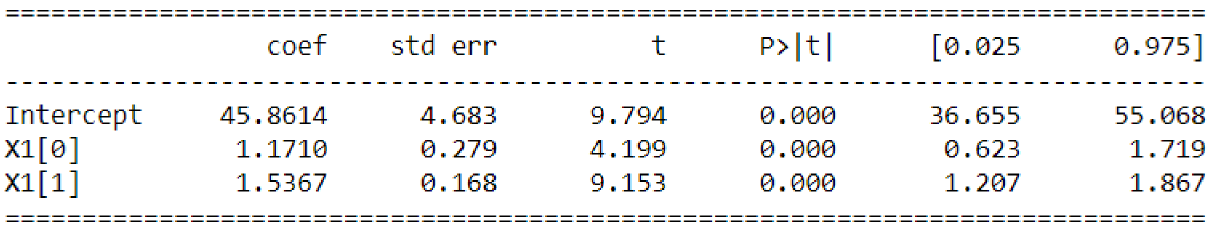
```
                 sum_sq     df            F         PR(>F)
X         959180.561342    4.0  1562.552916  2.393119e-239
Residual   60157.767497  392.0          NaN            NaN
```



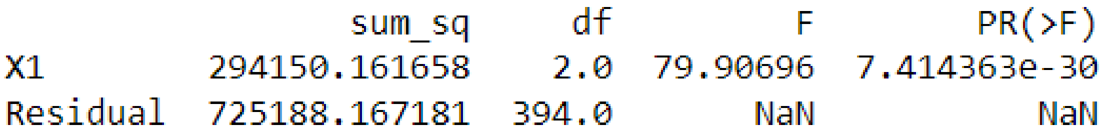
```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.5388      3.563      0.432      0.666      -5.466       8.544
X[0]           0.3766      0.183      2.056      0.040       0.017       0.737
X[1]           0.4107      0.119      3.466      0.001       0.178       0.644
X[2]           0.9451      0.040     23.502      0.000       0.866       1.024
==============================================================================
```



```
                 sum_sq     df           F         PR(>F)
X         717860.204865    3.0  311.928725  1.457117e-103
Residual  301478.123974  393.0         NaN            NaN
```



```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     45.8614      4.683      9.794      0.000      36.655      55.068
X1[0]          1.1710      0.279      4.199      0.000       0.623       1.719
X1[1]          1.5367      0.168      9.153      0.000       1.207       1.867
==============================================================================
```



```
                 sum_sq     df         F        PR(>F)
X1        294150.161658    2.0  79.90696  7.414363e-30
Residual  725188.167181  394.0       NaN           NaN
```

Hotspot analysis using SatSCan

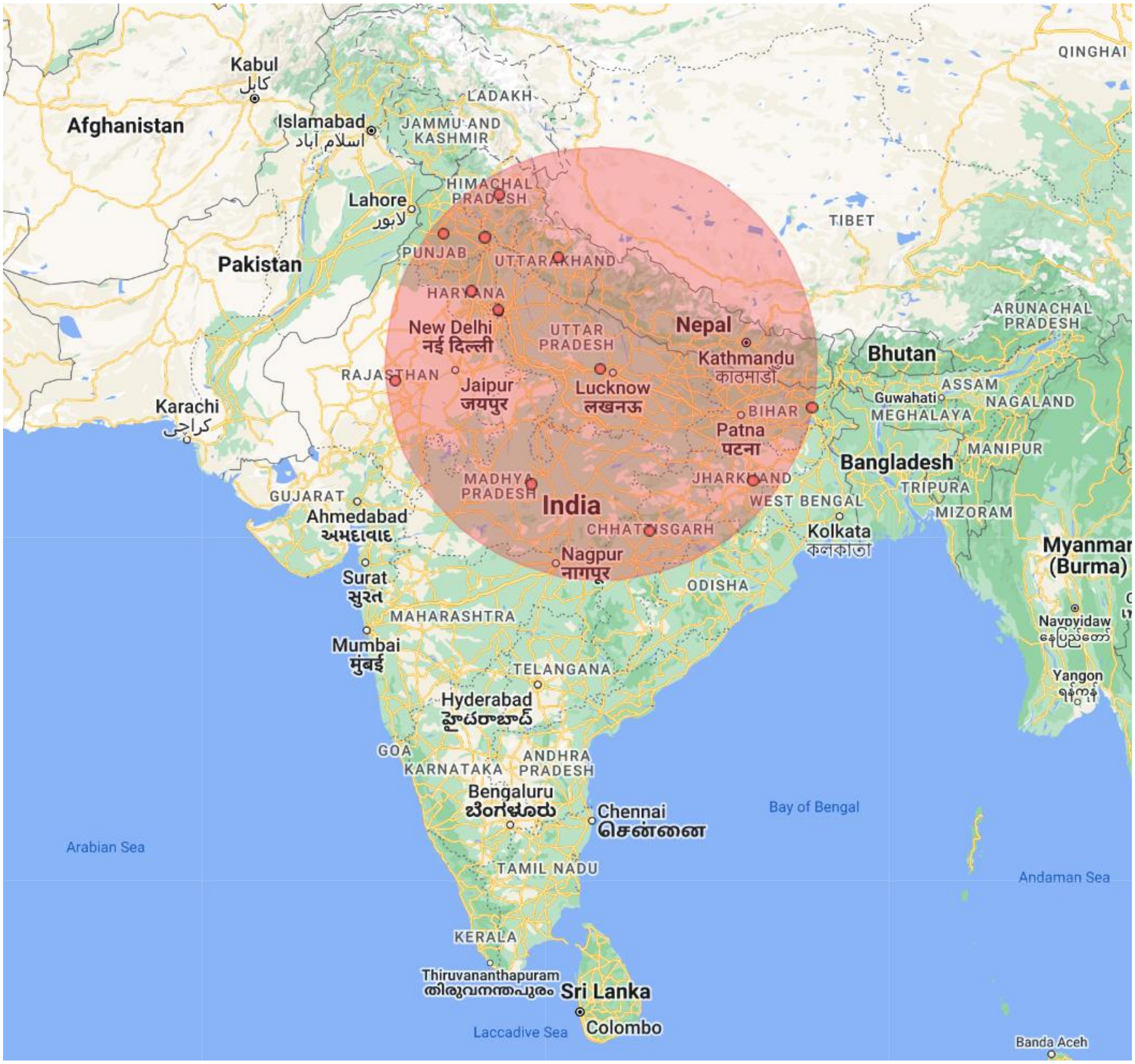


Fig 7.18 Normal Model for PM2.5 Concentration
For high values

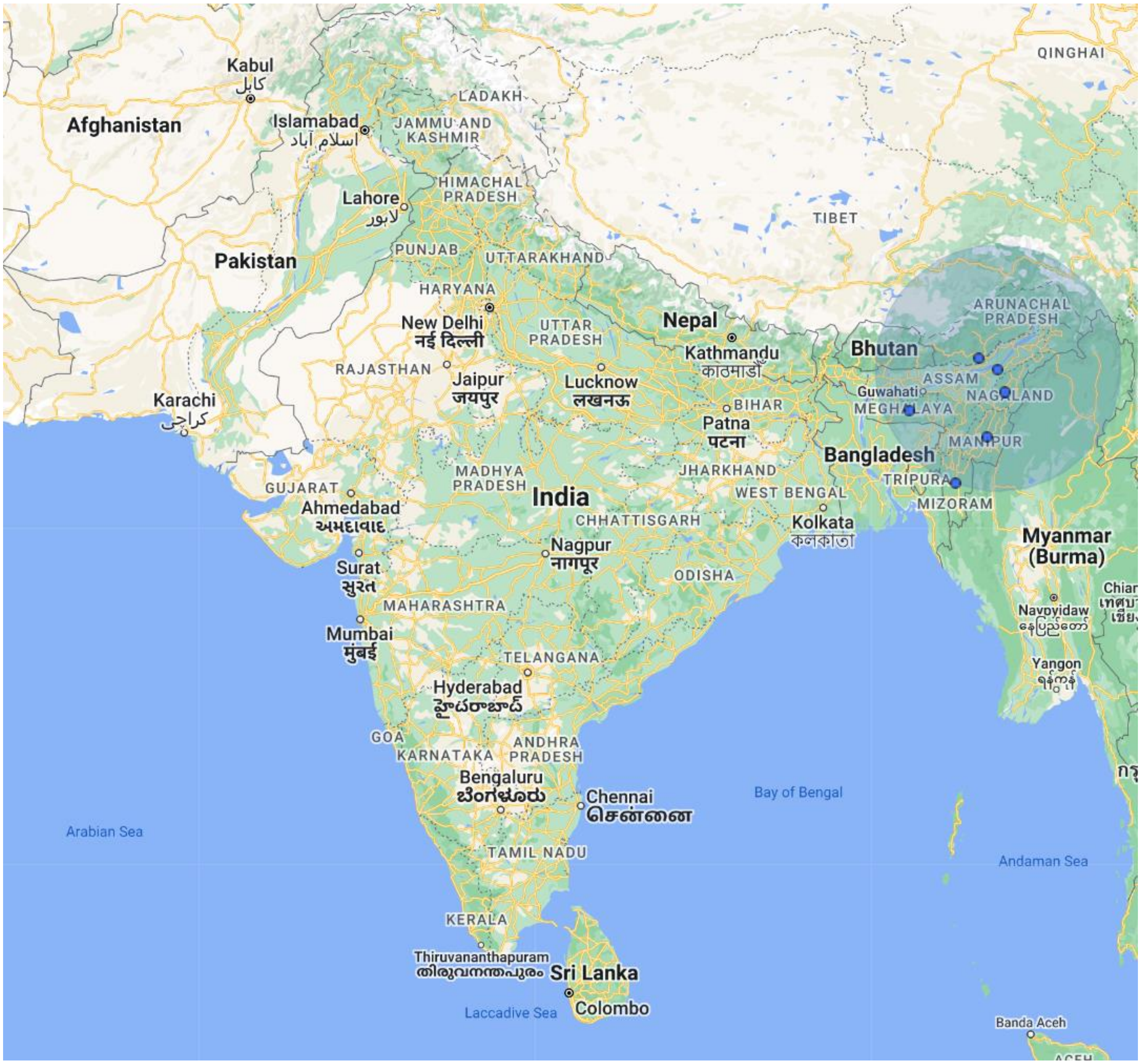


Fig 7.19 Normal Model for PM2.5 Concentration
For low values

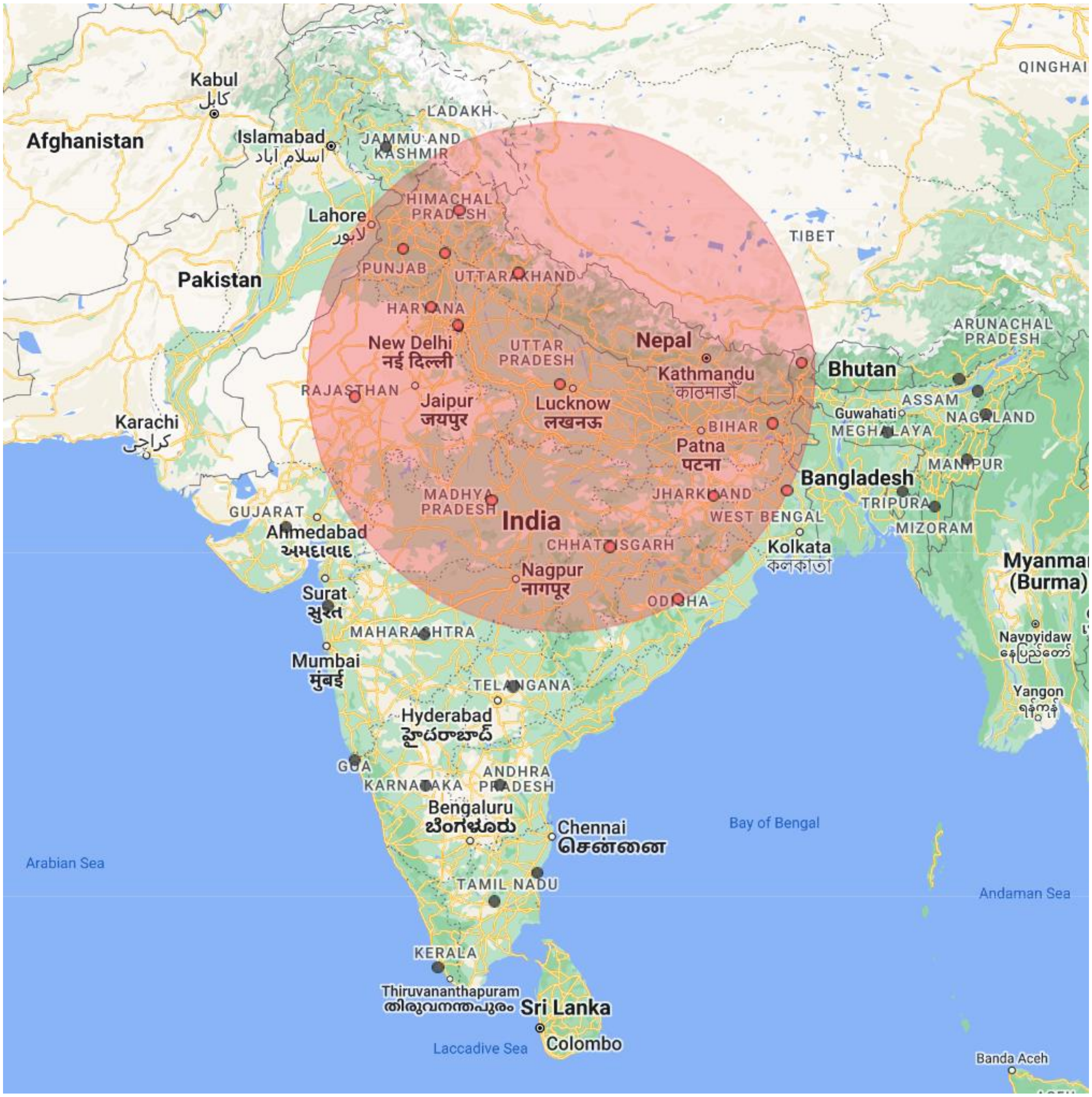


Fig 7.20 Ordinal model for PM2.5

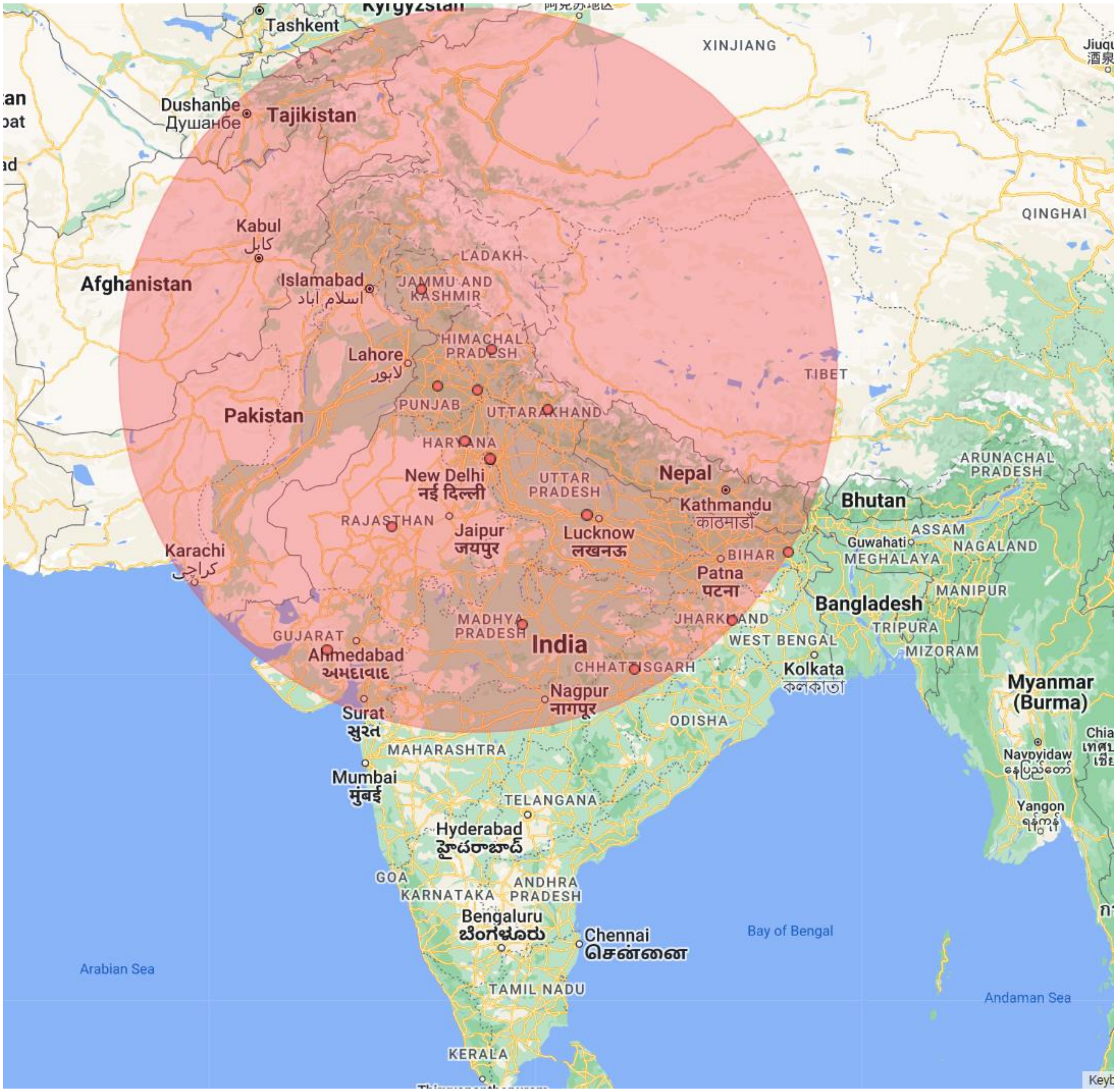


Fig7.21 Ordinal Model for PM10

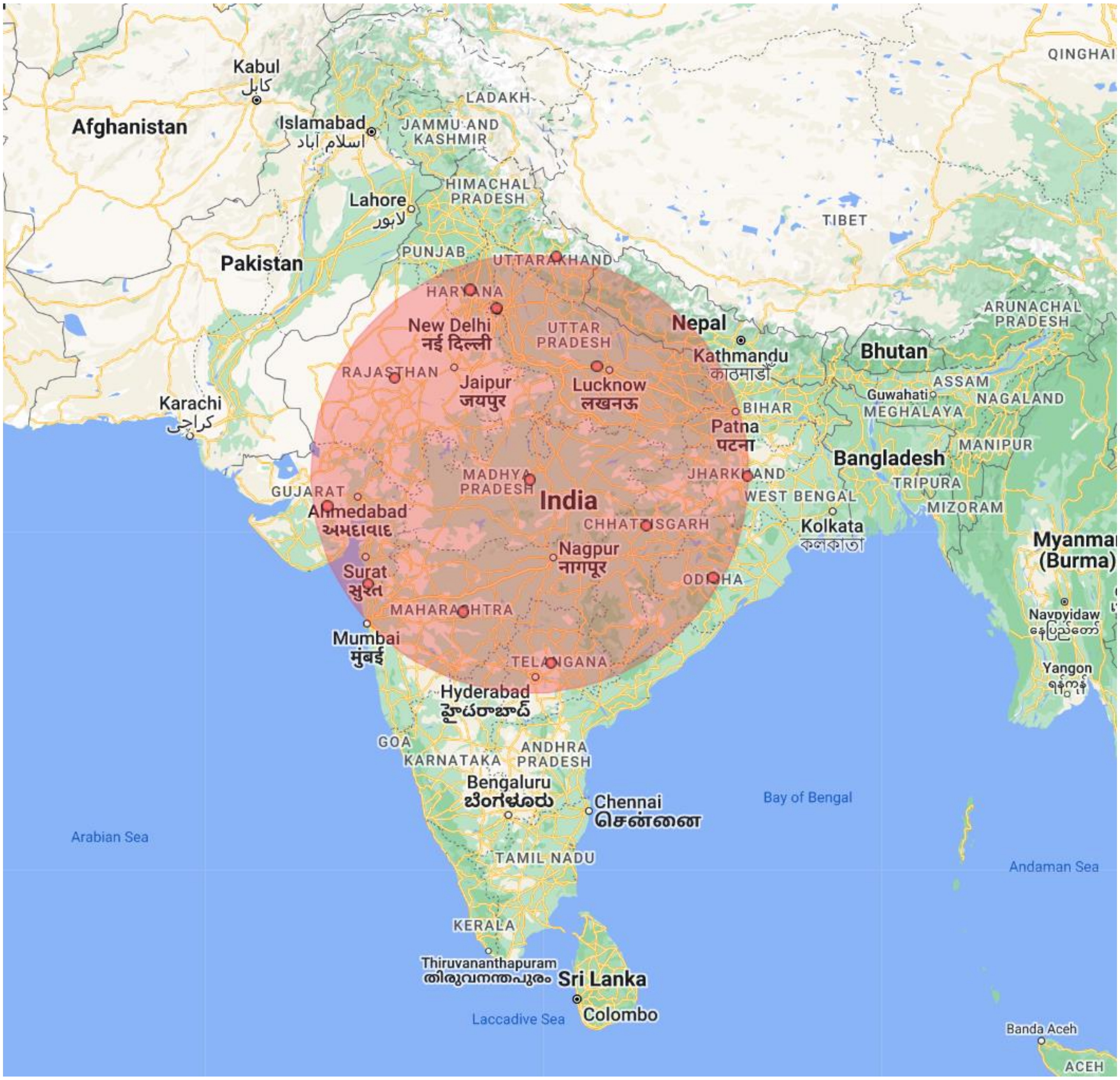


Fig 7.22 Ordinal Model For SO_2

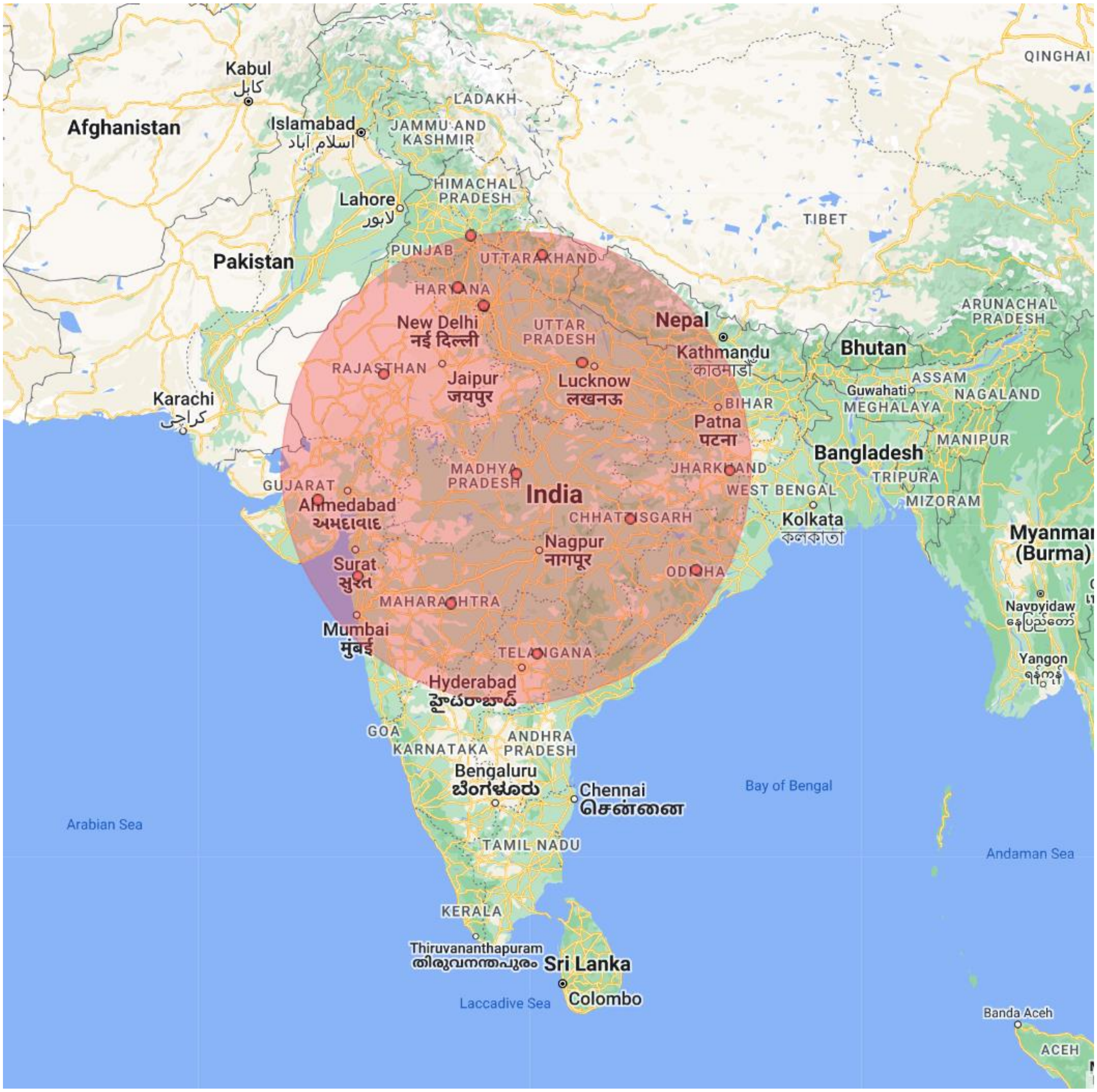


Fig 7.23 Ordinal Model For NO_2

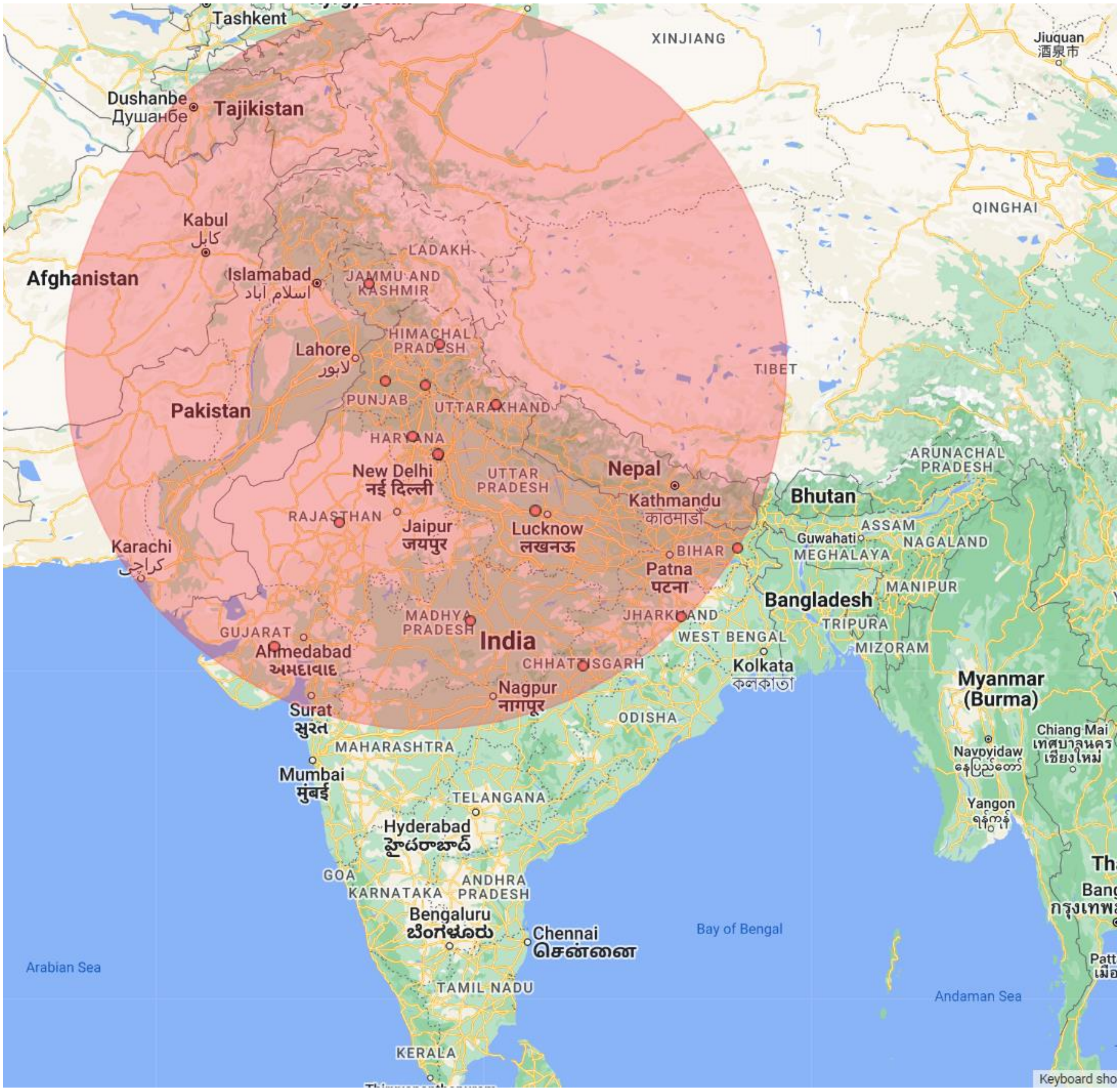
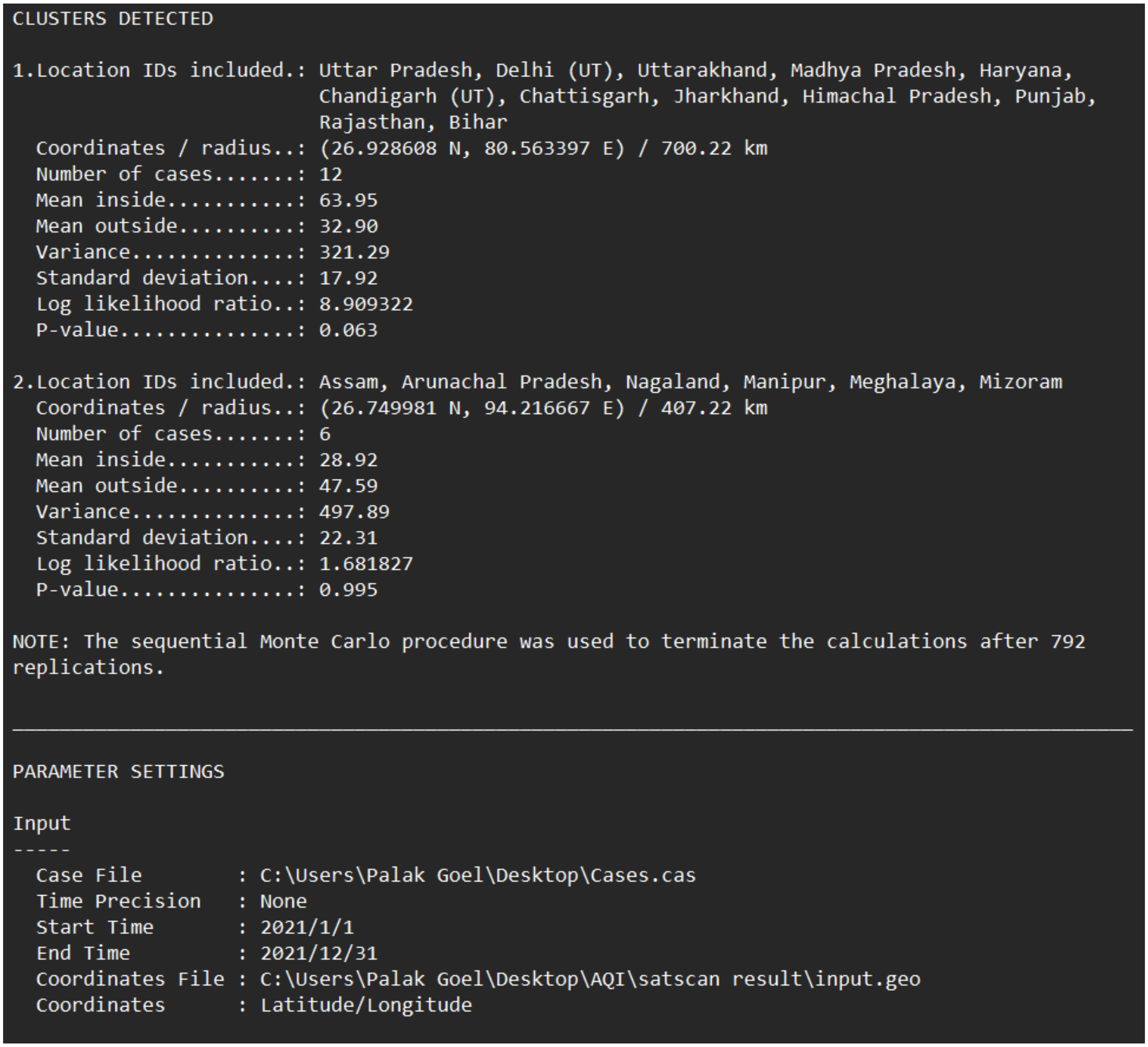


Fig 7.24 Ordinal Model For AQI

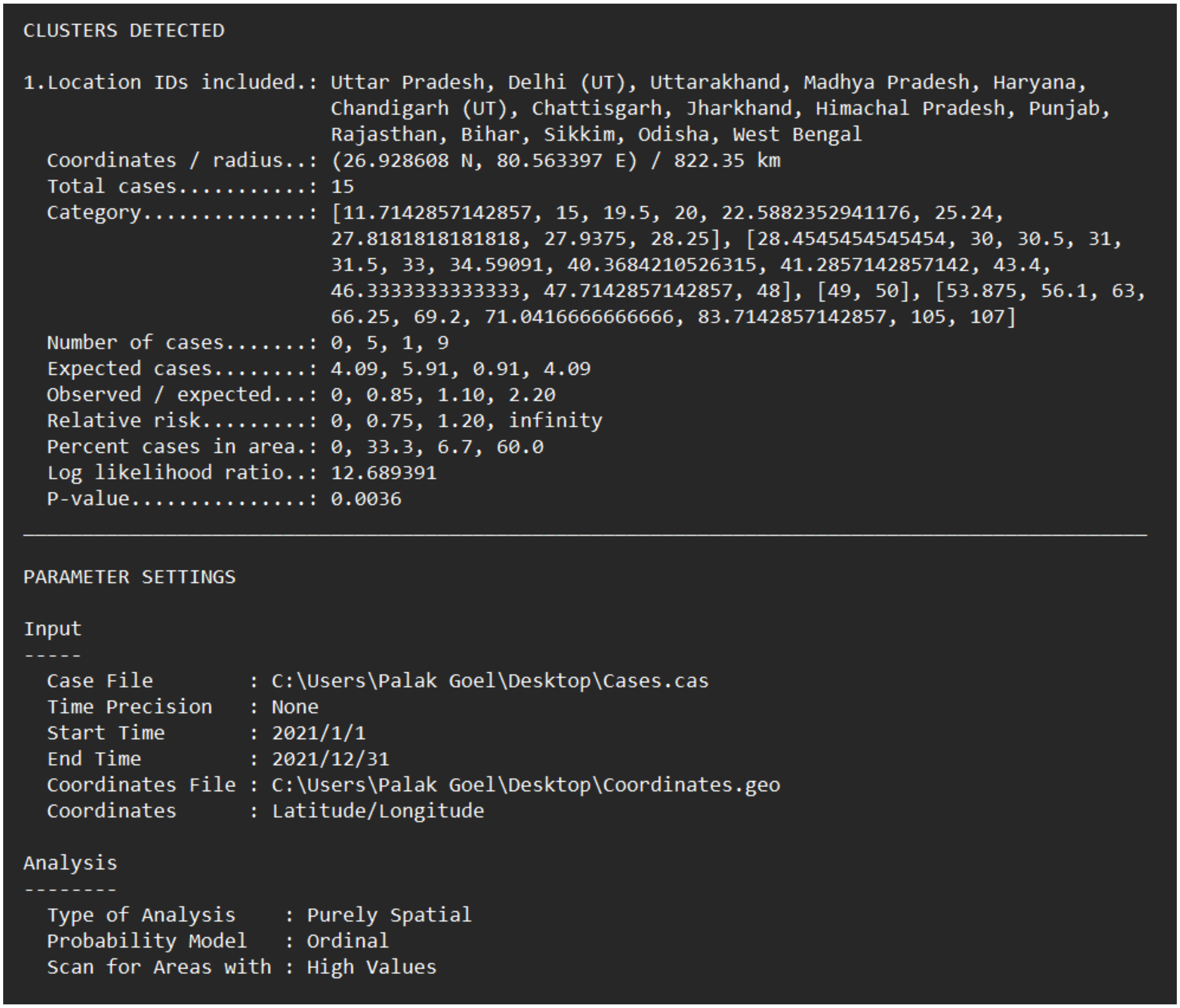
Screenshots of Results for hotspot analysis

1. **For PM2.5 (Normal Model)**



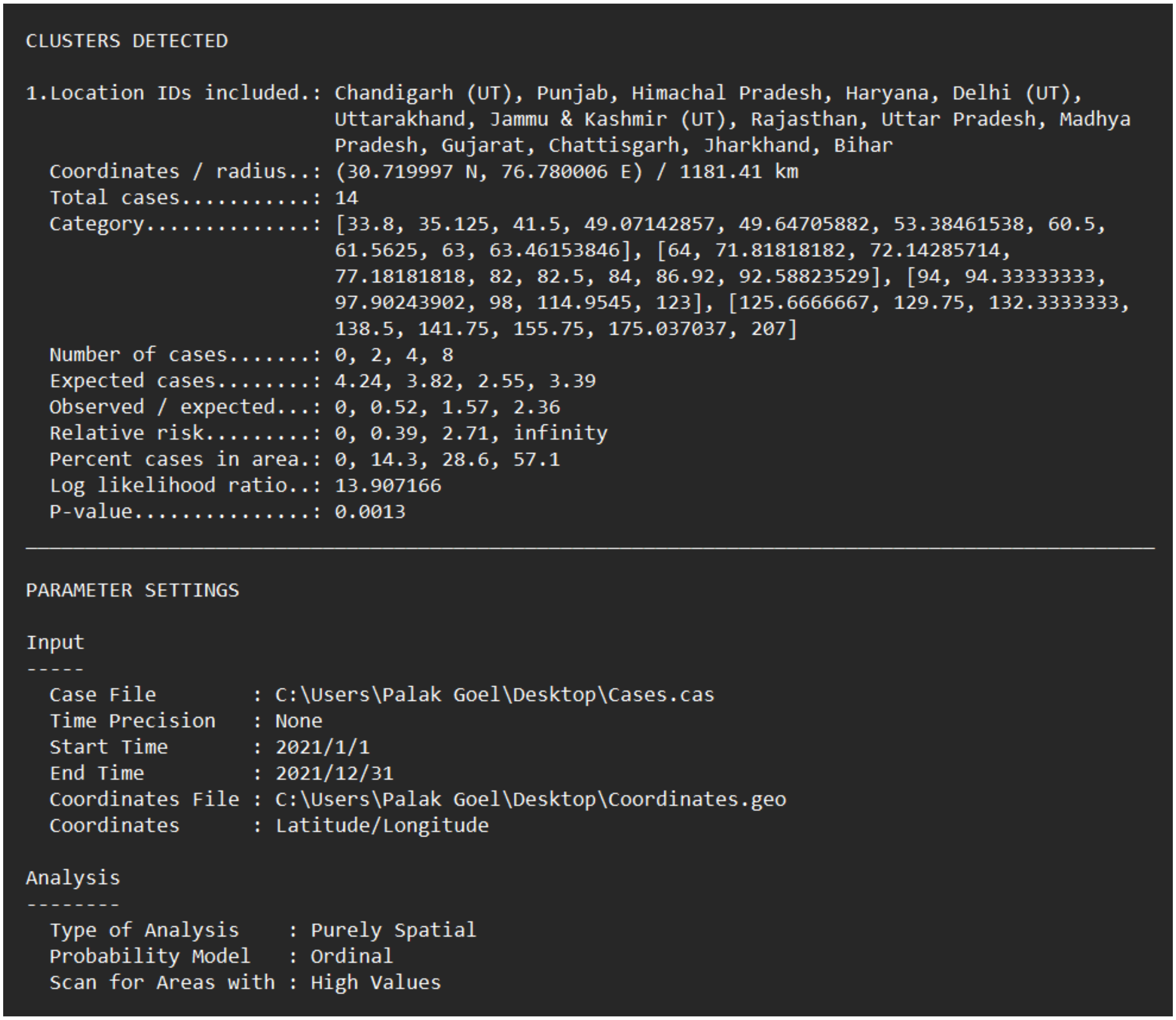
```
CLUSTERS DETECTED

1.Location IDs included.: Uttar Pradesh, Delhi (UT), Uttarakhand, Madhya Pradesh, Haryana,
                          Chandigarh (UT), Chattisgarh, Jharkhand, Himachal Pradesh, Punjab,
                          Rajasthan, Bihar
  Coordinates / radius..: (26.928608 N, 80.563397 E) / 700.22 km
  Number of cases.......: 12
  Mean inside...........: 63.95
  Mean outside..........: 32.90
  Variance..............: 321.29
  Standard deviation....: 17.92
  Log likelihood ratio..: 8.909322
  P-value...............: 0.063

2.Location IDs included.: Assam, Arunachal Pradesh, Nagaland, Manipur, Meghalaya, Mizoram
  Coordinates / radius..: (26.749981 N, 94.216667 E) / 407.22 km
  Number of cases.......: 6
  Mean inside...........: 28.92
  Mean outside..........: 47.59
  Variance..............: 497.89
  Standard deviation....: 22.31
  Log likelihood ratio..: 1.681827
  P-value...............: 0.995

NOTE: The sequential Monte Carlo procedure was used to terminate the calculations after 792
replications.

______________________________________________________________________________________________

PARAMETER SETTINGS

Input
-----
  Case File        : C:\Users\Palak Goel\Desktop\Cases.cas
  Time Precision   : None
  Start Time       : 2021/1/1
  End Time         : 2021/12/31
  Coordinates File : C:\Users\Palak Goel\Desktop\AQI\satscan result\input.geo
  Coordinates      : Latitude/Longitude
```

2. For PM2.5 (Ordinal model)



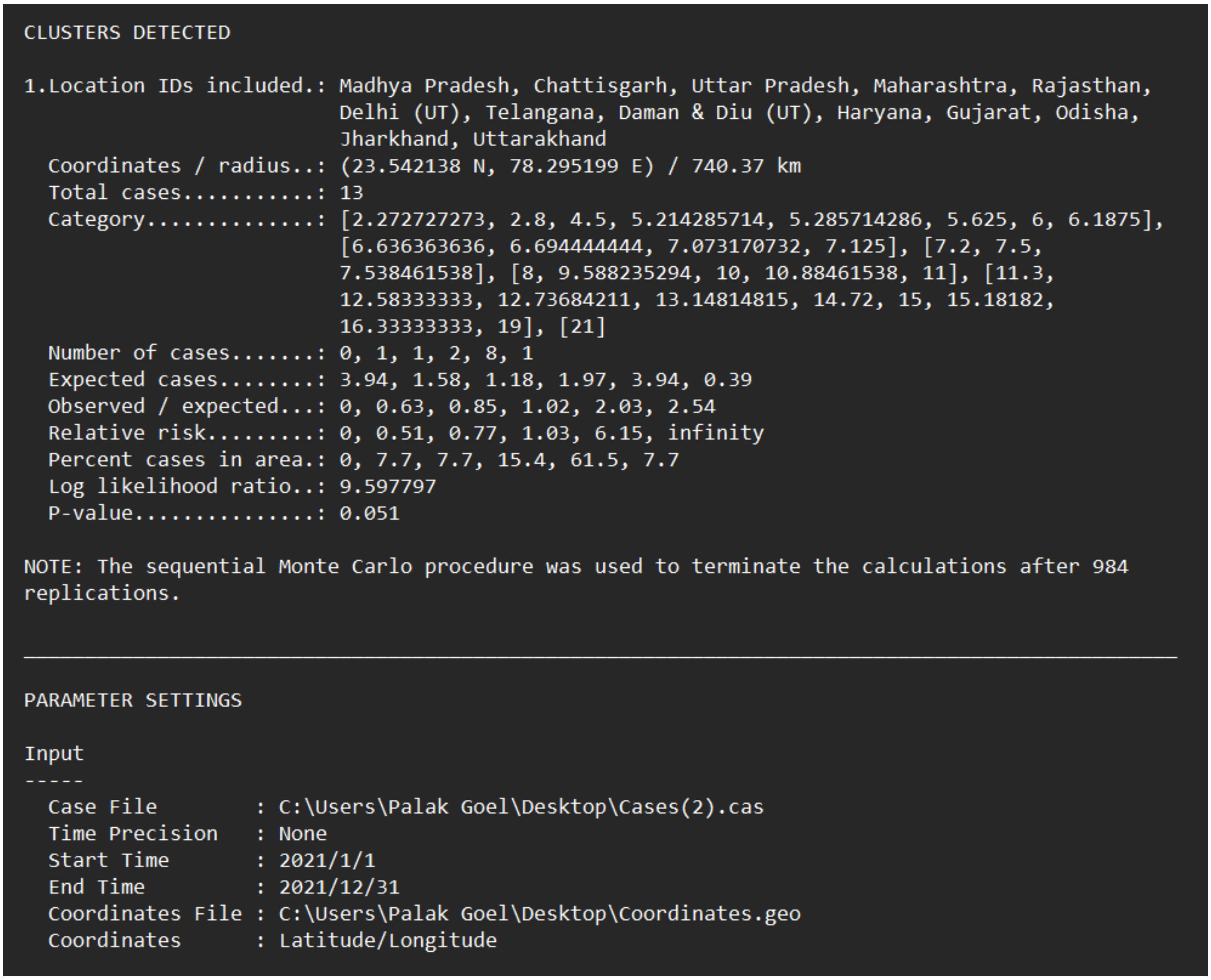
```
CLUSTERS DETECTED

1.Location IDs included.: Uttar Pradesh, Delhi (UT), Uttarakhand, Madhya Pradesh, Haryana,
                          Chandigarh (UT), Chattisgarh, Jharkhand, Himachal Pradesh, Punjab,
                          Rajasthan, Bihar, Sikkim, Odisha, West Bengal
  Coordinates / radius..: (26.928608 N, 80.563397 E) / 822.35 km
  Total cases...........: 15
  Category..............: [11.7142857142857, 15, 19.5, 20, 22.5882352941176, 25.24,
                          27.8181818181818, 27.9375, 28.25], [28.4545454545454, 30, 30.5, 31,
                          31.5, 33, 34.59091, 40.3684210526315, 41.2857142857142, 43.4,
                          46.3333333333333, 47.7142857142857, 48], [49, 50], [53.875, 56.1, 63,
                          66.25, 69.2, 71.0416666666666, 83.7142857142857, 105, 107]
  Number of cases.......: 0, 5, 1, 9
  Expected cases........: 4.09, 5.91, 0.91, 4.09
  Observed / expected...: 0, 0.85, 1.10, 2.20
  Relative risk.........: 0, 0.75, 1.20, infinity
  Percent cases in area.: 0, 33.3, 6.7, 60.0
  Log likelihood ratio..: 12.689391
  P-value...............: 0.0036
_______________________________________________________________________________________________

PARAMETER SETTINGS

Input
-----
  Case File        : C:\Users\Palak Goel\Desktop\Cases.cas
  Time Precision   : None
  Start Time       : 2021/1/1
  End Time         : 2021/12/31
  Coordinates File : C:\Users\Palak Goel\Desktop\Coordinates.geo
  Coordinates      : Latitude/Longitude

Analysis
--------
  Type of Analysis    : Purely Spatial
  Probability Model   : Ordinal
  Scan for Areas with : High Values
```

3. For PM10 (Ordinal Model)



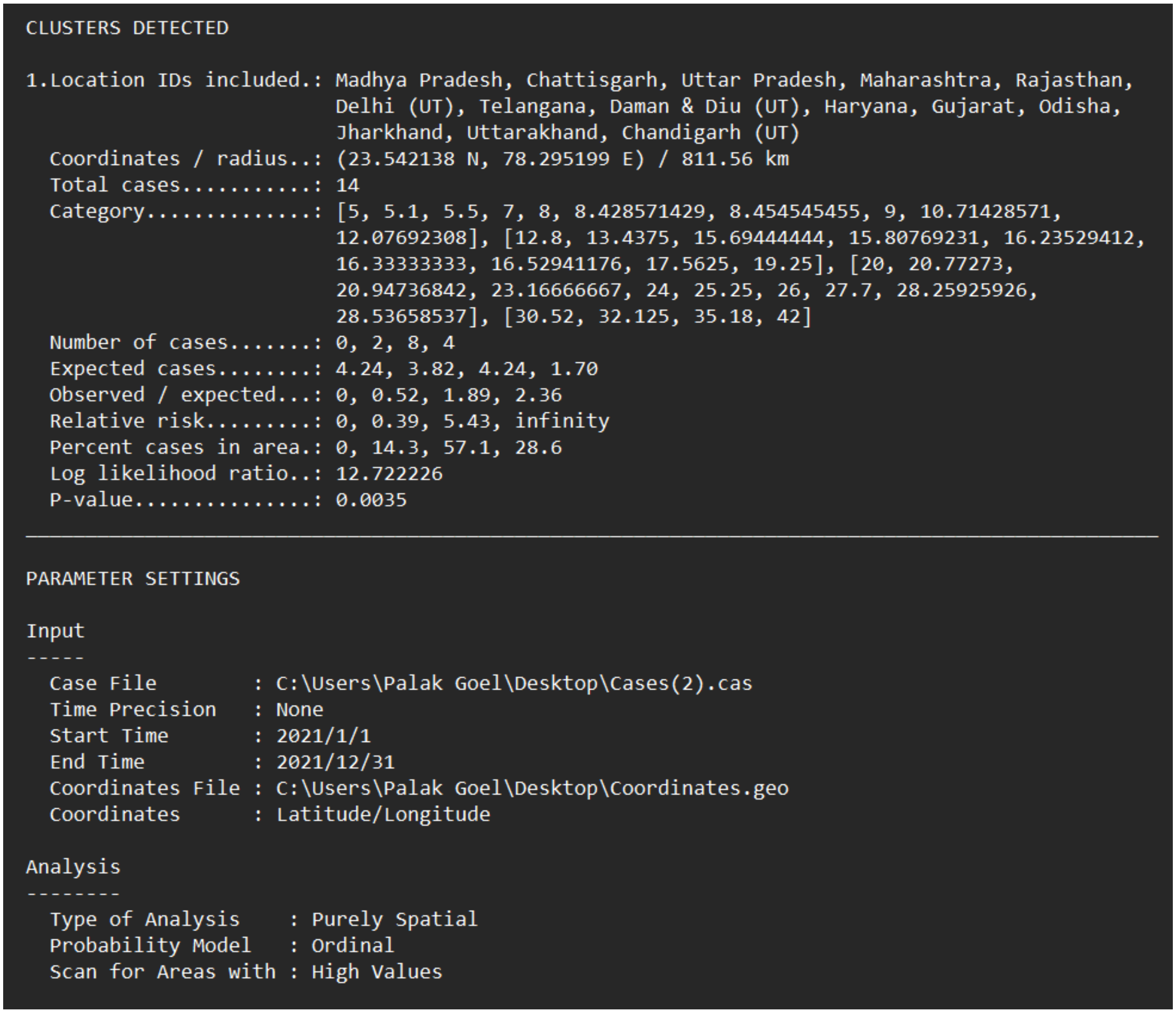
```
CLUSTERS DETECTED

1.Location IDs included.: Chandigarh (UT), Punjab, Himachal Pradesh, Haryana, Delhi (UT),
                          Uttarakhand, Jammu & Kashmir (UT), Rajasthan, Uttar Pradesh, Madhya
                          Pradesh, Gujarat, Chattisgarh, Jharkhand, Bihar
  Coordinates / radius..: (30.719997 N, 76.780006 E) / 1181.41 km
  Total cases...........: 14
  Category..............: [33.8, 35.125, 41.5, 49.07142857, 49.64705882, 53.38461538, 60.5,
                          61.5625, 63, 63.46153846], [64, 71.81818182, 72.14285714,
                          77.18181818, 82, 82.5, 84, 86.92, 92.58823529], [94, 94.33333333,
                          97.90243902, 98, 114.9545, 123], [125.6666667, 129.75, 132.3333333,
                          138.5, 141.75, 155.75, 175.037037, 207]
  Number of cases.......: 0, 2, 4, 8
  Expected cases........: 4.24, 3.82, 2.55, 3.39
  Observed / expected...: 0, 0.52, 1.57, 2.36
  Relative risk.........: 0, 0.39, 2.71, infinity
  Percent cases in area.: 0, 14.3, 28.6, 57.1
  Log likelihood ratio..: 13.907166
  P-value...............: 0.0013
_______________________________________________________________________________________________

PARAMETER SETTINGS

Input
-----
  Case File        : C:\Users\Palak Goel\Desktop\Cases.cas
  Time Precision   : None
  Start Time       : 2021/1/1
  End Time         : 2021/12/31
  Coordinates File : C:\Users\Palak Goel\Desktop\Coordinates.geo
  Coordinates      : Latitude/Longitude

Analysis
--------
  Type of Analysis    : Purely Spatial
  Probability Model   : Ordinal
  Scan for Areas with : High Values
```

4. For SO2 (Ordinal Model)



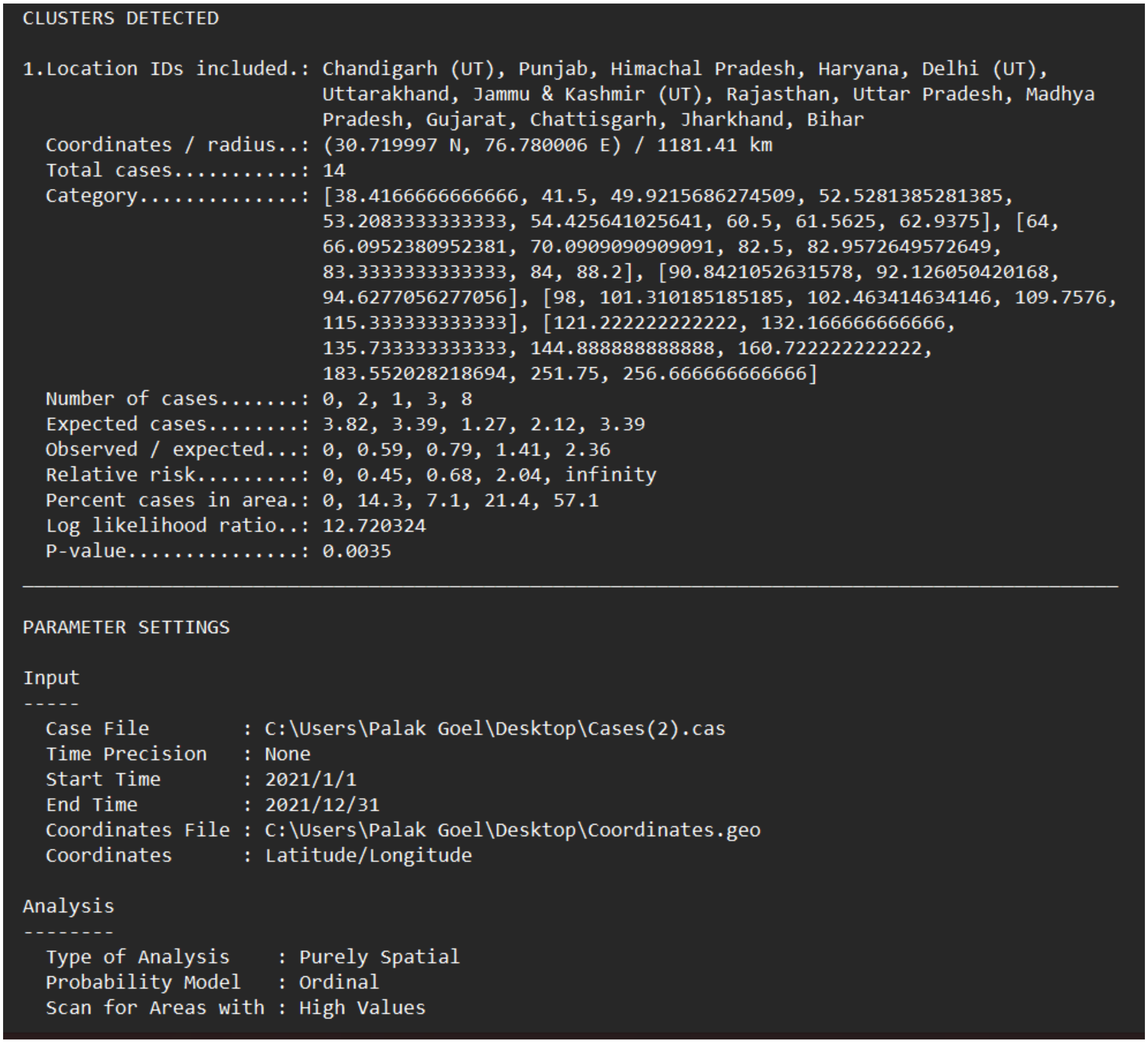
```
CLUSTERS DETECTED

1.Location IDs included.: Madhya Pradesh, Chattisgarh, Uttar Pradesh, Maharashtra, Rajasthan,
                          Delhi (UT), Telangana, Daman & Diu (UT), Haryana, Gujarat, Odisha,
                          Jharkhand, Uttarakhand
  Coordinates / radius..: (23.542138 N, 78.295199 E) / 740.37 km
  Total cases...........: 13
  Category..............: [2.272727273, 2.8, 4.5, 5.214285714, 5.285714286, 5.625, 6, 6.1875],
                          [6.636363636, 6.694444444, 7.073170732, 7.125], [7.2, 7.5,
                          7.538461538], [8, 9.588235294, 10, 10.88461538, 11], [11.3,
                          12.58333333, 12.73684211, 13.14814815, 14.72, 15, 15.18182,
                          16.33333333, 19], [21]
  Number of cases.......: 0, 1, 1, 2, 8, 1
  Expected cases........: 3.94, 1.58, 1.18, 1.97, 3.94, 0.39
  Observed / expected...: 0, 0.63, 0.85, 1.02, 2.03, 2.54
  Relative risk.........: 0, 0.51, 0.77, 1.03, 6.15, infinity
  Percent cases in area.: 0, 7.7, 7.7, 15.4, 61.5, 7.7
  Log likelihood ratio..: 9.597797
  P-value...............: 0.051

NOTE: The sequential Monte Carlo procedure was used to terminate the calculations after 984
replications.

_______________________________________________________________________________________________

PARAMETER SETTINGS

Input
-----
  Case File        : C:\Users\Palak Goel\Desktop\Cases(2).cas
  Time Precision   : None
  Start Time       : 2021/1/1
  End Time         : 2021/12/31
  Coordinates File : C:\Users\Palak Goel\Desktop\Coordinates.geo
  Coordinates      : Latitude/Longitude
```

5. For NO2 (Ordinal Model)



```
CLUSTERS DETECTED

1.Location IDs included.: Madhya Pradesh, Chattisgarh, Uttar Pradesh, Maharashtra, Rajasthan,
                          Delhi (UT), Telangana, Daman & Diu (UT), Haryana, Gujarat, Odisha,
                          Jharkhand, Uttarakhand, Chandigarh (UT)
  Coordinates / radius..: (23.542138 N, 78.295199 E) / 811.56 km
  Total cases...........: 14
  Category..............: [5, 5.1, 5.5, 7, 8, 8.428571429, 8.454545455, 9, 10.71428571,
                          12.07692308], [12.8, 13.4375, 15.69444444, 15.80769231, 16.23529412,
                          16.33333333, 16.52941176, 17.5625, 19.25], [20, 20.77273,
                          20.94736842, 23.16666667, 24, 25.25, 26, 27.7, 28.25925926,
                          28.53658537], [30.52, 32.125, 35.18, 42]
  Number of cases.......: 0, 2, 8, 4
  Expected cases........: 4.24, 3.82, 4.24, 1.70
  Observed / expected...: 0, 0.52, 1.89, 2.36
  Relative risk.........: 0, 0.39, 5.43, infinity
  Percent cases in area.: 0, 14.3, 57.1, 28.6
  Log likelihood ratio..: 12.722226
  P-value...............: 0.0035
______________________________________________________________________________________________

PARAMETER SETTINGS

Input
-----
  Case File        : C:\Users\Palak Goel\Desktop\Cases(2).cas
  Time Precision   : None
  Start Time       : 2021/1/1
  End Time         : 2021/12/31
  Coordinates File : C:\Users\Palak Goel\Desktop\Coordinates.geo
  Coordinates      : Latitude/Longitude

Analysis
--------
  Type of Analysis    : Purely Spatial
  Probability Model   : Ordinal
  Scan for Areas with : High Values
```

6. For AQI (Ordinal Model)



```
CLUSTERS DETECTED

1.Location IDs included.: Chandigarh (UT), Punjab, Himachal Pradesh, Haryana, Delhi (UT),
                          Uttarakhand, Jammu & Kashmir (UT), Rajasthan, Uttar Pradesh, Madhya
                          Pradesh, Gujarat, Chattisgarh, Jharkhand, Bihar
  Coordinates / radius..: (30.719997 N, 76.780006 E) / 1181.41 km
  Total cases...........: 14
  Category..............: [38.4166666666666, 41.5, 49.9215686274509, 52.5281385281385,
                          53.2083333333333, 54.425641025641, 60.5, 61.5625, 62.9375], [64,
                          66.0952380952381, 70.0909090909091, 82.5, 82.9572649572649,
                          83.3333333333333, 84, 88.2], [90.8421052631578, 92.126050420168,
                          94.6277056277056], [98, 101.310185185185, 102.463414634146, 109.7576,
                          115.333333333333], [121.222222222222, 132.166666666666,
                          135.733333333333, 144.888888888888, 160.722222222222,
                          183.552028218694, 251.75, 256.666666666666]
  Number of cases.......: 0, 2, 1, 3, 8
  Expected cases........: 3.82, 3.39, 1.27, 2.12, 3.39
  Observed / expected...: 0, 0.59, 0.79, 1.41, 2.36
  Relative risk.........: 0, 0.45, 0.68, 2.04, infinity
  Percent cases in area.: 0, 14.3, 7.1, 21.4, 57.1
  Log likelihood ratio..: 12.720324
  P-value...............: 0.0035
_______________________________________________________________________________________________

PARAMETER SETTINGS

Input
-----
  Case File        : C:\Users\Palak Goel\Desktop\Cases(2).cas
  Time Precision   : None
  Start Time       : 2021/1/1
  End Time         : 2021/12/31
  Coordinates File : C:\Users\Palak Goel\Desktop\Coordinates.geo
  Coordinates      : Latitude/Longitude

Analysis
--------
  Type of Analysis    : Purely Spatial
  Probability Model   : Ordinal
  Scan for Areas with : High Values
```

8. LIMITATIONS

Air quality prediction models have a number of limitations, including:

- Limited data: Air quality prediction models require a data in huge amount to make accurate predictions. However, in many parts of the world, there is limited data available on air quality, which can make it difficult to develop accurate models.
- Complex factors: Air quality is affected by a wide range of complex factors, including weather patterns, topography, and human activity. Modeling all of these factors accurately can be challenging.
- Uncertainty: There is always some degree of uncertainty associated with air quality prediction models. This can be due to limitations in the data, or the fact that air quality is influenced by many variables that are difficult to predict.
- Scale: Air quality models are typically developed at a regional or city-wide scale, which can limit their usefulness for predicting air quality at a more localized level.
- Model assumptions: Air quality prediction models are based on certain assumptions about the behavior of pollutants and their interactions with the environment. These assumptions may not always hold true, leading to inaccuracies in the model predictions.
- Lack of real-time data: Air quality models rely on data that is often several hours or even days old. This means that they may not accurately reflect the current air quality conditions in a given area.
- Difficulty in capturing spatial variability: Air quality prediction models may not be able to accurately capture the spatial variability of air quality within a city or region, leading to inaccuracies in predictions for specific locations.

9. RESULTS AND DISCUSSION

As our prediction model contains many limitations so it is a very challenging task to predict the AQI values having so much of variability. I also employ simulation techniques for model validation, which demonstrates that the analytical approach outperforms the traditional approach in terms of real-world values. Imputation is used to remove the Nan Values as the first job in this research project. Later check the skewness and conduct correlation analyses between AQI and predictor factors as well as within predictors. I was able to choose features for our model with the aid of this correlation study.EDA was also used to uncover various hidden details and patterns in our dataset, including the fact that Jharkhand and Tirupura has greater SO2 concentrations than Delhi and Telangana has higher NO2 levels.

Delhi and Jharkhand have greater PM2.5 concentrations.

Higher concentrations of PM10 are found in Delhi and Uttar Pradesh.

This demonstrates a stronger correlation between PM2.5 and PM10, pointing to multicollinearity in our dataset, which will cause dependency among explanatory variables

Plotted the AQI range using the box plot, swarm plot, and violin plot that show the automatic AQI values.

Moderate	196
Poor	128
Good	50
Unhealthy	23

After splitting the dataset into training and testing we applied machine learning algorithms on them then Found that K nearest neighbour is doing better in prediction as after addressing the issue of over fitting in Decision tree and random forest algorithm by removing the most correlated variable Pm2.5 and also the variance inflation factor of that variable was greater than 10 so we removed them but after removing that variable in model training the error metrics and performance metrics values steeped down which shows that the model will lead to less precision in predicting. In that case also K nearest neighbour works absolutely well and support vector machine algorithms has the lowest precision among all the algorithms so it under perform for AQI prediction

Results of Anova Analysis

The larger the *F* value, the more likely it is that the variation associated with the AQI is real and not due to chance.

The p value for the F statistic is shown in the Pr(>F) column. This demonstrates how likely it is that the F value would have happened if the null hypothesis that there is no difference between the group means were true.Because the *p* value of the independent variables, X consist of particulate matter , is statistically significant ($p < 0.05$), it is likely that particulate matter type does have a significant effect on AQI.After removing the independent variable I observed that R-Square value for the model decreased which indicates that prediction for that model will be less precised so we are taking SO2, NO2, PM2.5, PM10 as our independent variables.

Results of Machine learning algorithms:

REGRESSION ALGORITHM	**R2 Score(for PM10,PM2.5, SO2, NO2) For train data**	**R2 Score(PM10, SO2, NO2) For train data**
Linear Regression	0.94	0.68
Support Vector Machine	0.68	0.51
Decision Tree	1	0.99
Random Forest	0.99	0.96

REGRESSION ALGORITHM	**R2 Score(for PM10,PM2.5, SO2, NO2)for test data**	**R2 Score(PM10, SO2, NO2) For test data**
Linear Regression	0.95	0.81
Support Vector Machine	0.77	0.74
Decision Tree	0.99	0.58
Random Forest	0.99	0.77

Classification Algorithm	**Accuracy Score(for PM10,PM2.5, SO2, NO2) FOR train data**	**Accuracy Score (for PM10, SO2, NO2) for train data**
Logistic Reression	0.77	0.81
K Nearest Neighbour	0.95	0.89

Classification Algorithm	**Accuracy Score(for PM10,PM2.5, SO2, NO2)for test data**	**Accuracy Score (for PM10, SO2, NO2) for test data**
Logistic Reression	0.73	0.77
K Nearest Neighbour	0.91	0.90

ANOVA	For SO2, NO2, PM2.5, PM10	For SO2, NO2, PM10	For SO2, NO2
R-Square Value	0.94	0.70	0.29

In model validation simulation techniques, the analytical method outperforms the conventional method in terms of real-world values. Continuing the process by executing it repeatedly will enhance the value of root mean square and produce a nearly zero value for root mean square. It is not possible to manually perform iterations in Excel, so the result of one iteration is shown here. The majority of the work can be performed using @irisk software.

Satscan is utilised to ascertain the statistical significance of clusters reported in space or time. Consequently, my model identifies the concentrations for each type of particulate matter and AQI value.

Comparing the result of the normal model and the ordinal model for PM2.5 in India reveals that the ordinal model provides a more significant result, as well as a log-likelihood value that defines less overlapping in clusters and their significance. Compare numerous circles and choose the one with the highest likelihood value. This region is said to be the most probable cluster. This indicates that such a cluster is caused by human intervention and is unlikely to occur by accident. Under the null hypothesis that there are no clusters, the Monte Carlo sampling method must produce random copies of the data set. A likelihood ratio test must be conducted to compare the most likely clusters in actual and arbitrarily generated data sets. A cluster is statistically significant when its log likelihood ratio exceeds the significance level's critical value i.e alpha.

10. REFERENCES

[1] Sachin-Bhoite , Air_Quality_Prediction_using_Machine_Learning_Algorithms , Assistant Professor Computer Science MIT-WPU, Pune, India https://www.researchgate.net/profile/Sachin-Bhoite/publication/335911816_Air_Quality_Prediction_using_Machine_Learning_Algorithms/links/5d836662299bf1996f77746f/Air-Quality-Prediction-using-Machine-Learning-Algorithms.pdf

[2] K. Kumar, B. P. Pande , Islamic Azad University [2022], Air pollution prediction with machine learning: a case study of Indian cities, https://link.springer.com/article/10.1007/s13762-022-04241-5

[3] Jash J Patel1 , Prof. Neha R Patel2 , Prof. Hetal Gaudani3, [2022] Estimating of Particulate Matter using Machine Learning Approaches, , BVM Engineering College, Vallabh Vidyanagar, Gujarat, India , https://ijcrt.org/papers/IJCRT2204234.pdf

[4] Mr K. S. Raghu Kumar1 , Hemanth S2 , Swetha V3 , Sunil Naik V. S4, Prophecy of Air Quality using KNN-LSTM, Volume 2, Issue 9, June 2022, Rao Bahadur Y Mahabaleswarappa Engineering College Bellary, Karnataka, India , https://ijarsct.co.in/Paper5364.pdf

[5] Mrs. A. Gnana Soundari MTech, (PhD) Mrs. J. Gnana Jeslin M.E, (PhD) Akshaya A.C, Assistant Professor Jeppiaar Engineering College, INDIAN AIR QUALITY PREDICTION AND ANALYSIS USING MACHINE LEARNING Volume 14, Number 11, 2019 , https://www.ripublication.com/ijaerspl2019/ijaerv14n11spl_34.pdf

[6] Mrs. A. Gnana Soundari Mtech, (Phd), Mrs. J. Gnana Jeslin M.E, (Phd), Akshaya A.C. “Indian Air Quality Prediction And Analysis Using Machine Learning”. International Journal of Applied Engineering Research ISSN 0973-4

[7] Dragomir, Elia Georgian. "Air quality index prediction using K-nearest neighbor technique no. 1 (2010): 103-108.

[8] Gaganjot Kaur Kang, Jerry ZeyuGao, Sen Chiao, Shengqiang Lu, and Gang Xie(2018)." Air Quality Prediction: Big Data and Machine Learning Approaches". International Journal of Environmental Science and Development, Vol. 9, No. 1, January 2018

[9] Ruchi Raturi, Dr. J.R. Prasad. "Recognition Of Future Air Quality Index Using Artificial Neural Network". International Research Journal of Engineering and Technology (IRJET) .e-ISSN: 2395-0056 p-IS

[10] Aditya C R, Chandana R Deshmukh, Nayana D K, Praveen Gandhi Vidyavastu (2018)." Detection and Prediction of Air Pollution using Machine Learning Models". International Journal of Engineering Trends and Technology (IJETT) – volume 59 Issue 4 – May 2018

[11]Supriya Raheja , MohammadS. Obaidat, Balqies Sadoun , Sahil Malik , Anuj Rani , Manoj Kumar , Thompson Stephan(2021) Modeling and simulation of urban air quality with a 2-phase assessment technique, Volume 109, May 2021, 102281

[12] Alade IO, Rahman MAA, Saleh TA (2019a) Predicting the specifc heat capacity of alumina/ethylene glycol nanofuids using support vector regression model optimized with Bayesian algorithm. Sol Energy 183:74–82. https://doi.org/10.1016/j.solener.2019.02.060

[13]Gopalakrishnan V (2021) Hyperlocal air quality prediction using machine learning. Towards data science. https://towardsdatascie nce.com/hyperlocal-air-quality-prediction-using-machine-learn ing-ed3a661b9a71

[14] Challa Venkara Srinivas et al ," Data Assimilation and performance of Wrf for Air Quality Modeling in Mississippi Gulf Coastal Region "

[15] Gurjar BR (2021) Air pollution in india: major issues and challenges. energy future 9(2):12–27. https://www.magzter.com/stories/Education/Energy-Future/AIR-POLLUTION-IN-INDIA-MAJORISSUES-AND-CHALLENGES

[16] Sharma M E A McBean and U.Ghosh, “Prediction of atmospheric sulphate deposition at sensitive receptors in northern India”, Atmospheric Environment 29.16(1995): 2157- 2162.

[17] IHME (2019) State of global air 2019 report. http://www.healthdata. org/news-release/state-global-air-2019-report

[18] Hutchison Keith D., Solar Smith and Shazia J. Faruqui, “Correlating MODIS aerosol optical thickness data with ground-based PM2.5 observations across Texas for use in a real time air quality prediction system, “ Atmospheric Environment 39.37(2005) :7190 – 7203

[19] Wang Z et al , “ A nested air quality prediction modelling system for urban and regional scales : Application for high high-ozone episode in Taiwan “ Water, Air and Soil Pollution130.1-4(2001):391-396

[20]Madhuri VM, Samyama GGH, Kamalapurkar S (2020) Air pollution prediction using machine learning supervised learning approach. Int J Sci Technol Res 9(4):118–123

[21]Nallakaruppan, M. K., and U. Senthil Kumaran. "Quick fix for obstacles emerging in management recruitment measure using IOTbased candidate selection." Service Oriented Computing and Applications 12.3-4 (2018): 275- 284.

[22] Nallakaruppan, M. K., and Harun SurejIlango(2017). "Location Aware Climate Sensing and Real Time Data Analysis." Computing and Communication Technologies (WCCCT), 2017 World Congress on. IEEE, 2017.

[23] Liang Y, Maimury Y, Chen AH, Josue RCJ (2020) Machine learningbased prediction of air quality. Appl Sci 10(9151):1–17. https://doi.org/10.3390/app10249151

[24]Gokhale sharad and Namita Raokhande, “Performance evaluation of air quality models for predicting PM10 and PM2.5 concentrations at urban traffic intersection during winter period”, Science of the total environment 394.1(2008): 9- 24.

[25] Singh Kunwar P., Shikha Gupta and Premanjali Rai, " Identifying pollution sources and prediction urban air quality using ensemble learning methods", Atmospheric environment80 (2013): 426-43

[26] Madan T, Sagar S, Virmani D (2020) Air quality prediction using machine learning algorithms– a review. In: 2nd international conference on advances in computing, communication control and networking (ICACCCN) https://doi.org/10.1109/ICACCCN51052.2020.9362912

[27] Mahalingam U, Elangovan K, Dobhal H, Valliappa C, Shrestha S, Kedam G (2019) A machine learning model for air quality prediction for smart cities. In: 2019 international conference on wireless communications signal processing and networking (WiSPNET). IEEE 452–457. https://doi.org/10.1109/WiSPNET45539.2019. 9032734

[28] Monisri PR, Vikas RK, Rohit NK, Varma MC, Chaithanya BN (2020) Prediction and analysis of air quality using machine learning. Int J Adv Sci Technol 29(5):6934–6943

[29] Nahar K, Ottom MA, Alshibli F, Shquier MA (2020) Air quality index using machine learning—a jordan case study. COMPUSOFT, Int J Adv Comput Technol 9(9):3831–3840

[30] Patil RM, Dinde HT, Powar SK (2020) A literature review on prediction of air quality index and forecasting ambient air pollutants using machine learning algorithms 5(8):1148–1152

[31] Rogers CD (2019) Pollution's impact on historical monuments pollution's impact on historical monuments. SCIENCING. https://sciencing.com/about-6372037-pollution-s-impact-historical-monum ents.html

[32] Rybarczyk Y, Zalakeviciute R (2017) Regression models to predict air pollution from afordable data collections. In: H. Farhadi (Ed.), Machine learning advanced techniques and emerging applications pp 15–48. IntechOpen. https://doi.org/10.5772/intechopen.71848

[33] Rybarczyk Y, Zalakeviciute R (2021) Assessing the COVID-19 impact on air quality: a machine learning approach. Geophys Res Lett. https://doi.org/10.1029/2020GL091202

[34]Spatio-Temporal Aspect of Suicide and Suicidal Ideation: An Application of SaTScan to Detect Hotspots in Four Major Cities of Tamil Nadu By Anjali, B. Rushi Kumar*, and Jitendra Kumar
Department of Mathematics, School of Advanced Sciences, VIT, Vellore-632014, TN
https://www.researchgate.net/publication/358804579_SpatioTemporal_Aspect_of_Suicide_and_Suicidal_Ideation_An_Application_of_SaTScan_to_Detect_Hotspots_in_Four_Major_Cities_of_Tamil_Nadu