B.Tech Minor Project Report

COT-415

on

AIR QUALITY PREDICTION

BY

N SRI HARSHA (11610186)
V SAI SHRAWAN (11610220)
P RAKESH KUMAR (11610222)

Under the Supervision of
Dr. S.K.JAIN, (Professor, NITK)

DEPARTMENT OF COMPUTER ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

KURUKSHETRA – 136119, HARYANA (INDIA)

November, 2019

CERTIFICATE

We, hereby certify that this work which is being presented in this B.Tech. Minor Project (COT-415) report entitled " **AIR QUALITY PREDICTION",** in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Engineering** is an authentic record work of my work carried out during the period from July, 2019 to November, 2019 under the supervision of Dr.S.K.Jain ,Professor, Computer Engineering Department.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

Signature of Candidate
N SRI HARSHA (11610186)
V SAI SHRAWAN (11610220)
P RAKESH KUMAR (11610222)

This is to certify that the above statements made by the candidates is correct to the best of my knowledge.

Date:

Signature of Supervisor
Dr. S.K. JAIN
(Professor, NITK)

TABLE OF CONTENTS

I. ABSTRACT

For the past few decades the contamination in air quality as well as pollution became a major threat. According to World Health Organization report, at least several deaths occur world wide usually because of air pollution. The prediction of quality of air helps people to for see the damage and it keeps the air pollution under control. In this regard , the machine Learning algorithms are proven to be immensely useful. The main focus of this project analyze the data recorded by India air quality. The index of air quality value is on the basis of concentration of some major air pollutants such as nitrogen dioxide, respirable particular matter, sulphur dioxide and suspended particular matter. Eventually, the classification of air quality ranges as good as satisfactory, moderately polluted, very poor and severe.

II. INTRODUCTION

Air pollutant :

An air pollutant that originates naturally or by artificial means have very harmful impact on humans. The ash formed from the eruption of volcano produces the primary pollutants. The Sulphur dioxide and Carbon dioxide gas is emitted from factories and motor vehicles respectively. The secondary pollutants are produced from the reaction or interaction of the primary pollutants the information about some of the air pollutants which are used in case of estimation of the index of air quality and there by analyze it qualitatively is as follows:

Sulphur Dioxide (SO_2):

Sulphur dioxide is one of the toxic gases which is formed as a by-product of extraction of copper and contamination of fossil fuels with components of SO_2. The processing of various materials that contains sulphur in the industry is the main source of SO_2. It is also implied by the fact that presence of 99% of sulphur dioxide in the air is due to various human sources. SO_2 has sharp and nasty smell. Moreover it is invisible. The reaction of SO_2 with some other substances forms various harmful components such as Sulfurous acid and sulphate particles. The harmful effect of SO_2 are evident swiftly through worse symptoms almost 15 minutes after its intake from respiration. It causes tight feeling around the chest because of shortness of breathe. It also causes coughing,wheezing and irritates the nose.

Nitrogen Dioxide(NO_2):

NO_x is the generic term which refers to Nitrogen oxides. NO2 is a hazardous gas which causes significant air pollution. It is composed of oxygen and nitrogen. It reacts with ozone and contribute to the particulate pollution in the out door air. It is actually one of the six massively spread air pollutants in accordance with the standards of national air quality. The presence of the No2 within the indoor air is mainly because of burning fossil fuels like wood or natural gas.NO2 has several adverse effects on human such as reduction in lung functioning, increased air ways inflammation and asthma effects. Recent research has linked in NO2 to the cardiovascular attacks and also sometimes results in the premature death of new born babies with lower weight.

Particulate Matter(PM):

PM is the summation of hazardous liquid and solid particles suspension in air . The complex mixture includes dust liquid droplets smoke etc.., i.e various inorganic and organic particles.The particles in air are either emitted directly due to burning of fuel and dust i.e carried by wind. It is emitted indirectly because of emission of various gases pollutants to air. The particulate matter is classified into two groups mainly on the basis of site:

1. The inhalable coarse fraction whose site ranges from 2.5 to 10 micrometer and consists of large particles (PM10 to PM2.5)
2. The fine fraction whose site is almost 2.5 micrometer and consists of smaller particles.

The particles whose size is lesser than 0.1 micrometer can be considered as ultra fine particles.

Most of the air born particulate matter quantity is because of the presence of fine particles. The fine particles are mainly formed from gaseous substances where as the coarse particles are formed from uncovered soil and mining operations, agricultural process and mechanical breakup.

The PM2.5 particles possess high risk to human health. The particles that are less than 10 micrometers in diameter can enter deep into lungs and even into blood circulation.

Data Set:

The data set released by the Central Pollution Control Board of India(CPCBI) and Ministry of Environment and forest under the National Data Sharing an Accessibility Policy(NDSAP).It is clear version data of historical daily ambient air quality

Data Description(Features) :

1. Stn_code(the code of corresponding station)

2. Sampling date(the date of sample collection)

3. States(The corresponding state in India)

4. Location (Location where sample collection is done)

5. Agencies

6. Type(Area type)

7. SO2 (The concentration of Sulphur dioxide in microgram per meter cube)

8. NO2 (The concentration of Nitrogen dioxide in microgram per meter cube)

9. RSPM (The concentration of respirable suspended particulate matter which is in the units of micrograms per meter cube).

10. SPM (The concentration of suspended particulate matter which is in the units of microgram per meter cube).

11. Location (Monitoring station)

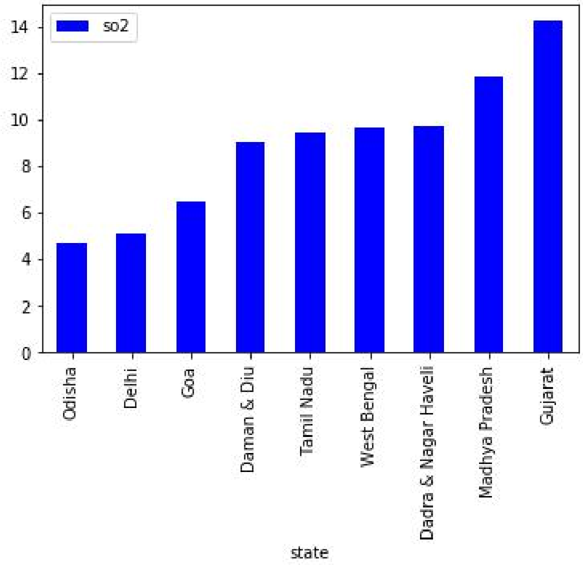
12. $PM_{2.5}$ (Particulate matter which is in the units of microgram per meter cube).

13. Date

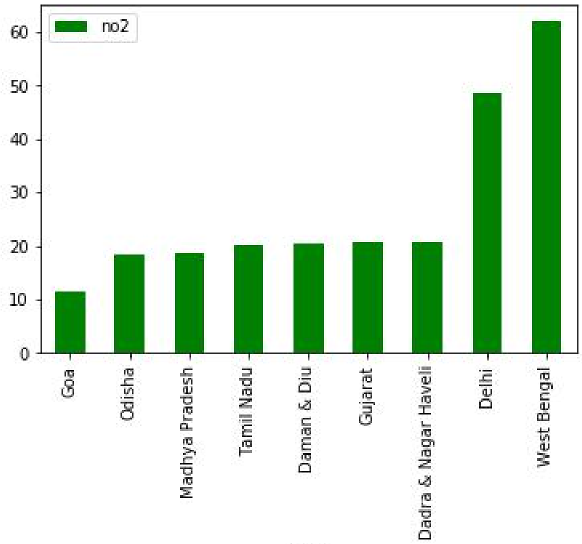
The Data set consists of 30,000 rows and columns.

Visualization of States and individual pollutants:

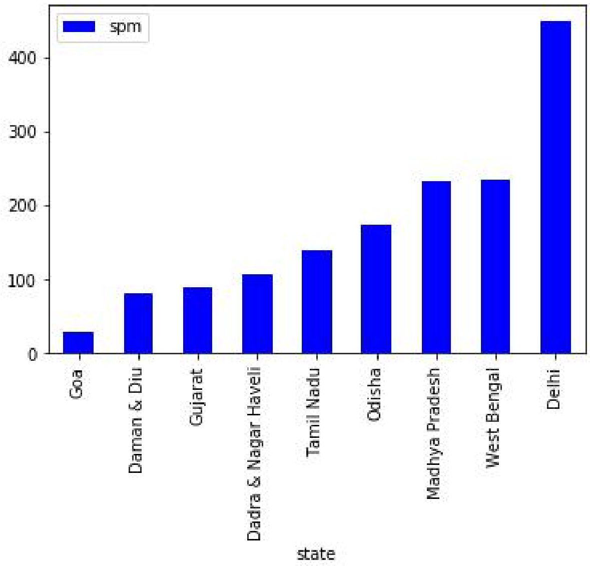
1. Concentration of Sulphur dioxide



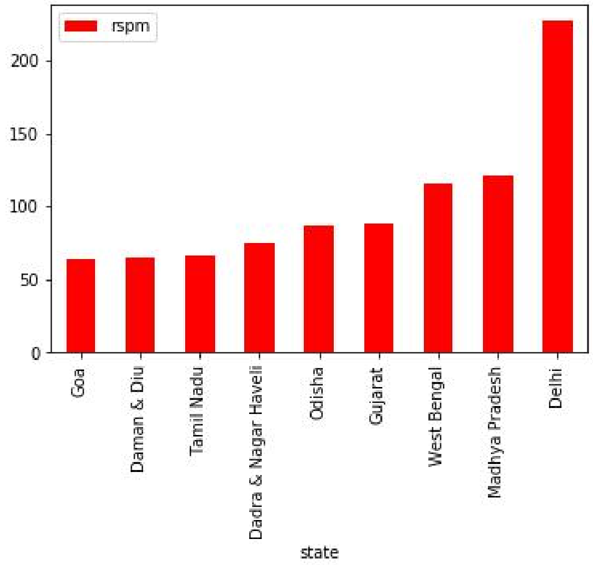
2. Concentration of Nitrogen dioxide



3. Concentration of suspended particulate matter



4. Concentration of Respirable Suspended particulate matter



Machine Learning algorithms used for prediction:

Here is the brief description about various machine learning prediction techniques used.

1. Multiple Linear Regression :

Multiple linear regression method, which is also called as the multiple regression. It can be realized as extension of the ordinary least squares(OLS) regression method which involves one or more explanatory variables. It is basically a statistical approach which uses various variables which are explanatory in order to predict the response variable's outcome.

The formula for the multiple regression is :

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

Here for n=i number of observations

Yi=the dependent variables

Xi =the explanatory variables

B0=y-interrupt (a term which is constant)

Bp = The coefficient of slope for each

T= The error term in the model (which are also called as residuals).

2. **Logistic Regression**

The logistic model which is also called logit model is used for probability modelling of a event or class.

Logistic regression is basically a statistical technique which is used for binary dependent variable modelling based on logistic function. Logit regression is also used for parameter estimation of a logistic model. In a logistic model,the logs-odds for the label one values is the linear combination of one or more than one independent variables where each variable can either be continuous or be binary.

3. Random Forest

The Random forest, which is also called as the Random decision forest corrects the behaviour of decision trees with respect to over fitting to the training set. It is basically an ensemble learning method in order to perform various tasks including classification and regression. A training time, it indulges in the construction of multitude of decision trees and produces a single class as output which is mean prediction of the individual trees for regression and mode of all the classes for the classification.

4. Decision Tree

A decision tree model which is also called as model of decision is a tool which uses an acyclic graph structure it also leads to possible consequences which include cost of the resources , event out comes based on decisions made. In a decision tree, each of the internal node notifies the test on some feature and each of the branch depicts the test out come and each of the leaf node represents the label of the class.

5. K- Nearest Neighbour

KNN is basically a supervised machine learning methodology, which is a non-parametic method used for regression as well as classification. In the K-nn regression, output produced is object attribute’s resultant value. In K-nn classification the output is the membership often object for a particular class based on the number of votes obtained by observing the classes for which k nearest neighbors belongs. The local approximation of function occurs and the computation is performed only after classification which implies K-nearest neighbour is actually a type of instant based learning, which can also be addressed as lazy learning.

III. MOTIVATION

A major study carried out in the 650 cities across the globe regarding the urban pollution has confirmed that air pollution has hazardous impact on the health of the human beings in short term.It is linked to the death of the people who frequently suffer from illness and elderly people in a discrete manner.It has been scientifically proven that quality of air massively effects health of the people and pollution free environment greatly signifies paves the pay to lead a better quality of life.Because of air pollution, Many bodily functions are tampered such as fatigue,cardiovascular,nervous system and reproductive system damage and so on.

In order to overcome this scenario,various models of machine learning are used such as the linear model, logistic model, model containing decision tree, model containing random forest, knn model and support vector machine model in order to predict index value of air quality.The Indian air quality index uses bands. It is on the basis of the concentration of major air pollutants. The key benefit of using an AQI is that it has the ability to communicate data with public,

conveniently and easily . AQI values are color coded and easily scaled, which exempts one to understand concentrations and units. It is rarely valuable to transform the data into a from that the public can understand. The AQI value can be presented in an understandable and recognizable form for everyone. More over it requires minimum effort and minimum calculation inorder to compute the index value of air quality.

IV. LITERATURE SURVEY

The prominent work has been done in the air quality prediction using machine learning as well as approaches by various researches and scholars. Richard O sinnott and ziyue Gyaan collected air pollution data , especially PM2.5 value (Particulate matter less than 2.5 micrometers) from various web based resources and predicted air pollution using various models such as Artificial neural networks(ANN), the Long short term memory(LTSM) recurrent neural networks. Ying Zhang , Yanhao Wang , Minghe Gad proposed an improved version of air quality prediction method based in light GBM mode for the prediction of the concentration of fine particulate matter, PM2.5 in Beijing at 35 stations which monitors the quality of air. Mahanijah Md kamal , Rozita Jailini worked towards the investigation of effectiveness of the Neural network that propagate backwards (BPNN) and the Artificial neural networks(ANN) for the prediction of the ambient air quality for air quality monitoring in Malaysian states. Mohamad Reza Delavar, Amin Gholami used various prediction models to determine air pollution in Tehran city based on concentrations of $PM_{2.5}$ and $PM_{10.}$ The machine learning methods used to include geographically weighted regression, artificial neural network, support vector machine and auto-regressive non linear neural network with latter being the most reliable model. Costandna Veljanovska , Angel Dimoski compared four machine learning algorithms neural network, decision tree model, k-nearest neighbors technique and support vector machines in terms of prediction of air quality index. The database used contain meteorological information of each day of 2017 in the republic of Macedonia capital city. It was proven that these algorithms when implemented predicts AQI value in an efficient manner.

V. PROPOSED APPROACH

The given data set should be normalized in accordance with the prediction requirements before getting trained with various machine learning models.

Data cleaning:

Drop the unnecessary columns:

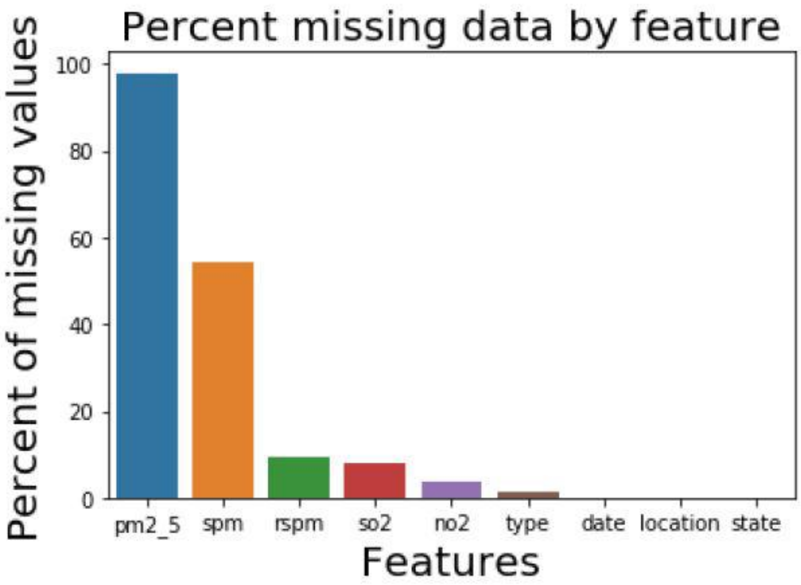
Stn_code

Agency

Sampling_date

Location_monitoring_station

Percent of missing values (Bar plot):



In order to perform the further analysis, missing values in the data set are filled by the mean of each columns grouped by states. This is done through the method of Imputation.

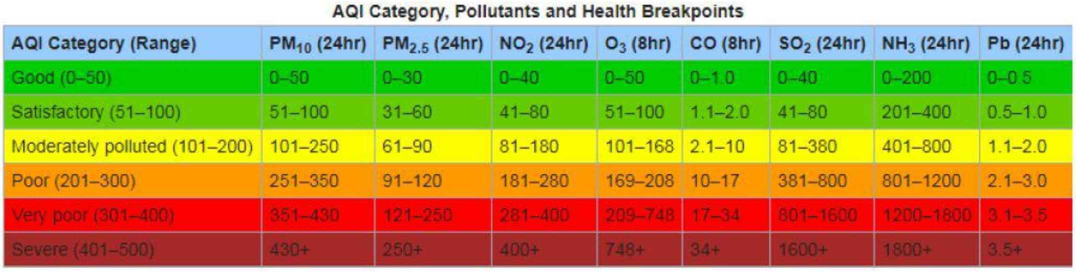
Missing data can create problems for data analysis. Instead of list wise deletion of missing values cases imputation can be realized as a way to avoid this pitfall. This method basically deals with the replacement of missing data with substituted values.

Derivation for individual pollutant index and AQI:

The air quality index transforms complex air quality data that consists of various air pollutants into a single value which represents air index. Color and nomenclature also act as a tool for air quality status communication to the people in easily understandable and highly effective form.

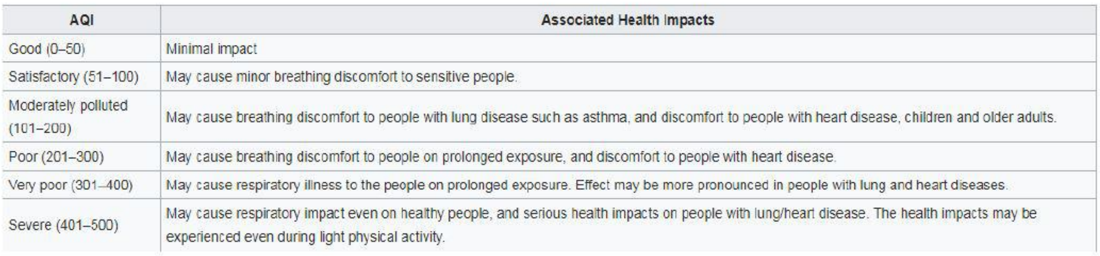
On the basis of the ambient concentration levels of the major pollutants of air and their affect on human health which is also called as health breakpoints. The AQI categories are named as severe,very poor,moderately polluted,satisfactory and good. As the AQI arises, the risk of human health intensifies.

The national air monitoring program(NAMP) has been conducted on more than 342 monitoring stations over 240 cities. This programme was operated by both state and central control boards. According to the prescription of national ambient air quality standards,the propose AQI consider eight air pollutants.They are SO_2, NO_2, NH_3, PB, PM2.5, PM10, CO, O_3 whose values of AQI along with concentrations and the impacts on heath associated to it are represented as follows:



AQI Category, Pollutants and Health Breakpoints

AQI Category (Range)	PM_{10} (24hr)	$PM_{2.5}$ (24hr)	NO_2 (24hr)	O_3 (8hr)	CO (8hr)	SO_2 (24hr)	NH_3 (24hr)	Pb (24hr)
Good (0–50)	0–50	0–30	0–40	0–50	0–1.0	0–40	0–200	0–0.5
Satisfactory (51–100)	51–100	31–60	41–80	51–100	1.1–2.0	41–80	201–400	0.5–1.0
Moderately polluted (101–200)	101–250	61–90	81–180	101–168	2.1–10	81–380	401–800	1.1–2.0
Poor (201–300)	251–350	91–120	181–280	169–208	10–17	381–800	801–1200	2.1–3.0
Very poor (301–400)	351–430	121–250	281–400	209–748	17–34	801–1600	1200–1800	3.1–3.5
Severe (401–500)	430+	250+	400+	748+	34+	1600+	1800+	3.5+



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

Computing the AQI:

The AQI i.e index of air quality can be realized as piece vise function of concentration of pollutant. A jump of discontinuity of an unit of AQI occurs at the boundaries between categories of AQI. The conversion of concentration to AQI value can be done through the following equation.

$$I = \frac{I_{high} - I_{low}}{C_{high} - C_{low}}(C - C_{low}) + I_{low}$$

Where

C= Pollutant Concentration

I= The index of quality of air

C_{low} = The break point of concentration which is less than C.

C_{high} =The break point of concentration which is greater than C.

I_{low} = The breakpoint of index that corresponds to C_{low}.

I_{high} = The breakpoint of index that corresponds to C_{high}.

The stated formula is used in the calculation of pollution index for SO_2, NO_2, PM2.5, RSPM, SPM.

The calculation of a sub index is calculated for each of the mentioned air pollutants based on the likely health impacts and the ambient concentrations. The overall AQI is reflected by the value of worst sub index. If multiple air pollutants exhibit it at the monitoring location site, the largest AQI value of the pollutant is considered for the corresponding location.

After this step data analysis and visualization is done and as per the required criteria. The data has also been standard dated accordingly. The various machine learning models that used on this modified data to predict the air quality index(AQI) are presented as follows:

Linear regression model 1:

Training Features : SOi, NOi, RSPMi, SPMi (these are the pollution indices calculated)

Target Feature: AQI (the calculated index of air quality)

The split of data set occurred such that 80% is for training and 20% is for testing purposes.

Linear regression model2 :

Training Features : SO2, NO2, RSPM, SPM (the concentration of given pollutants).

Target Feature: AQI (the calculated index of air quality)

The split of data set occurred such that 80% is for training and 20% is for testing purposes.

Classification : For the AQI values a new column that contains corresponding AQI range values was added using a function for the multiclass classification.

Logistic Regression model 1:

Training Features : SOi, NOi, RSPMi, SPMi (these air pollution indices)

Target Feature: AQI_range(the calculated range of air quality index)

The split of data set occurred such that 80% is for training and 20% is for testing purposes.

Logistic regression model2 :

Training Features : SO2, NO2, RSPM, SPM (the concentration of given pollutants).

Target Feature: AQI_range (the calculated range of air quality index)

The split of data set occurred such that 80% is for training and 20% is for testing purposes.

Decision Tree Classifier model :

Training Features : SO2, NO2, RSPM, SPM (the concentration of given pollutants).

Target Feature: AQI_range (the calculated range of air quality index)

The split of data set occurred such that 80% is for training and 20% is for testing purposes.

Random Forest Classifier model :

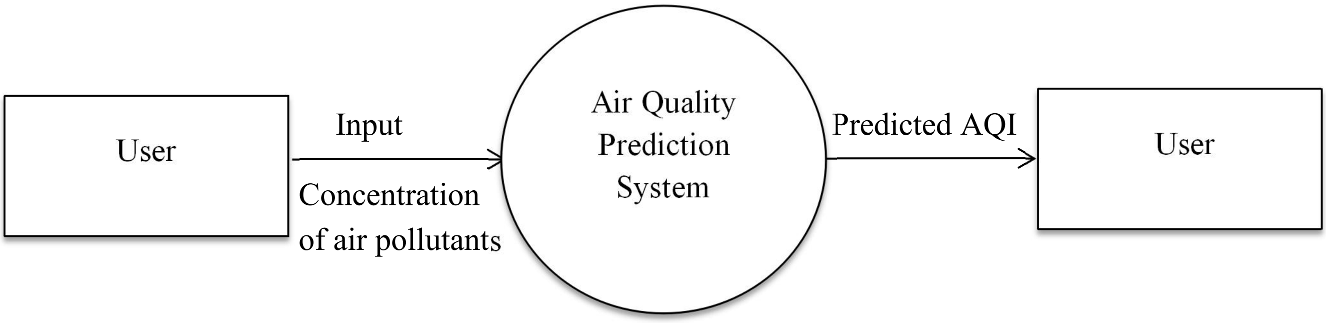
Training Features : SO2, NO2, RSPM, SPM (the concentration of given pollutants).

Target Feature: AQI_range (the calculated range of air quality index)

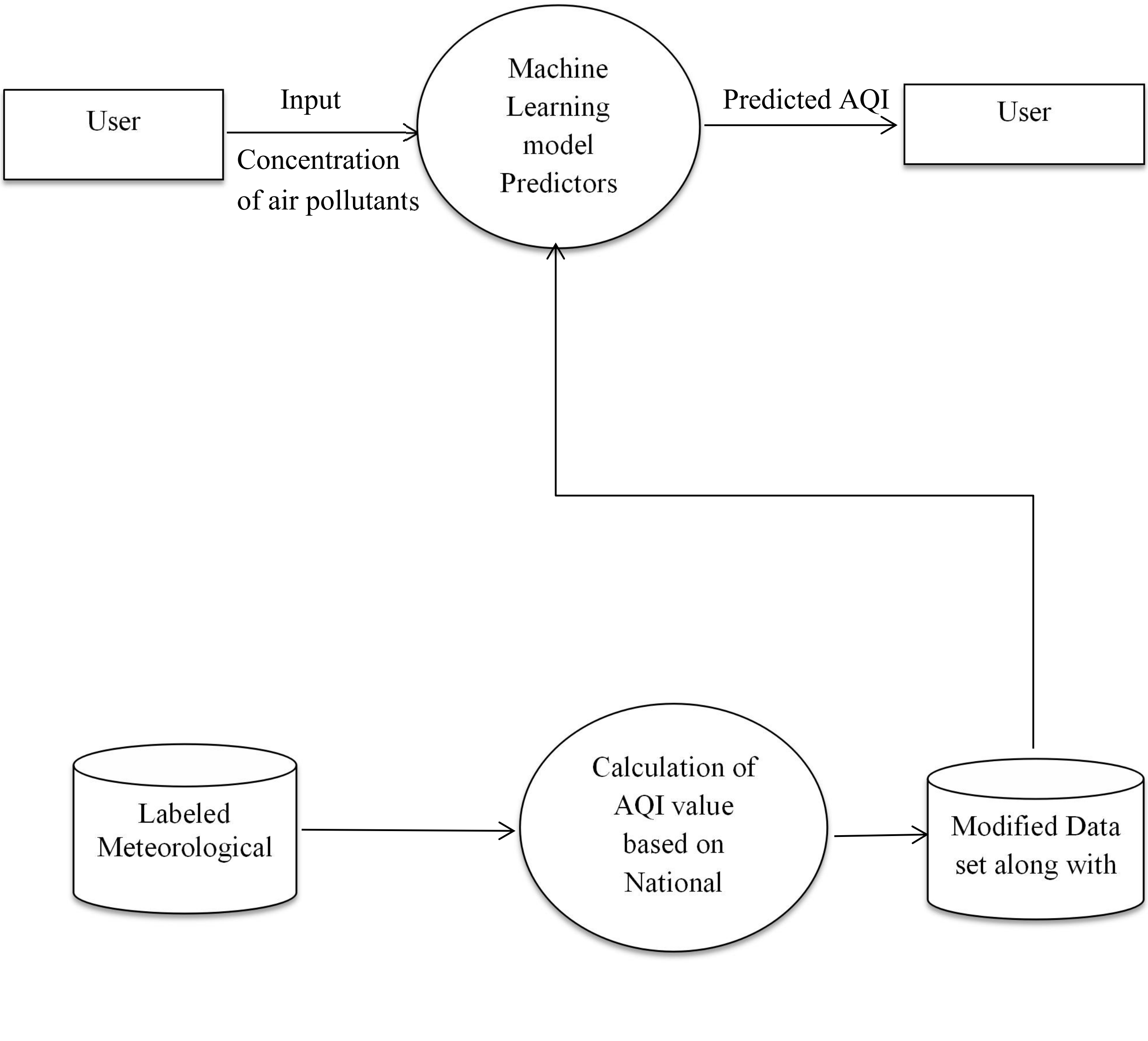
The split of data set occurred such that 80% is for training and 20% is for testing purposes.

VI. DATA FLOW DIAGRAMS (DFD's)

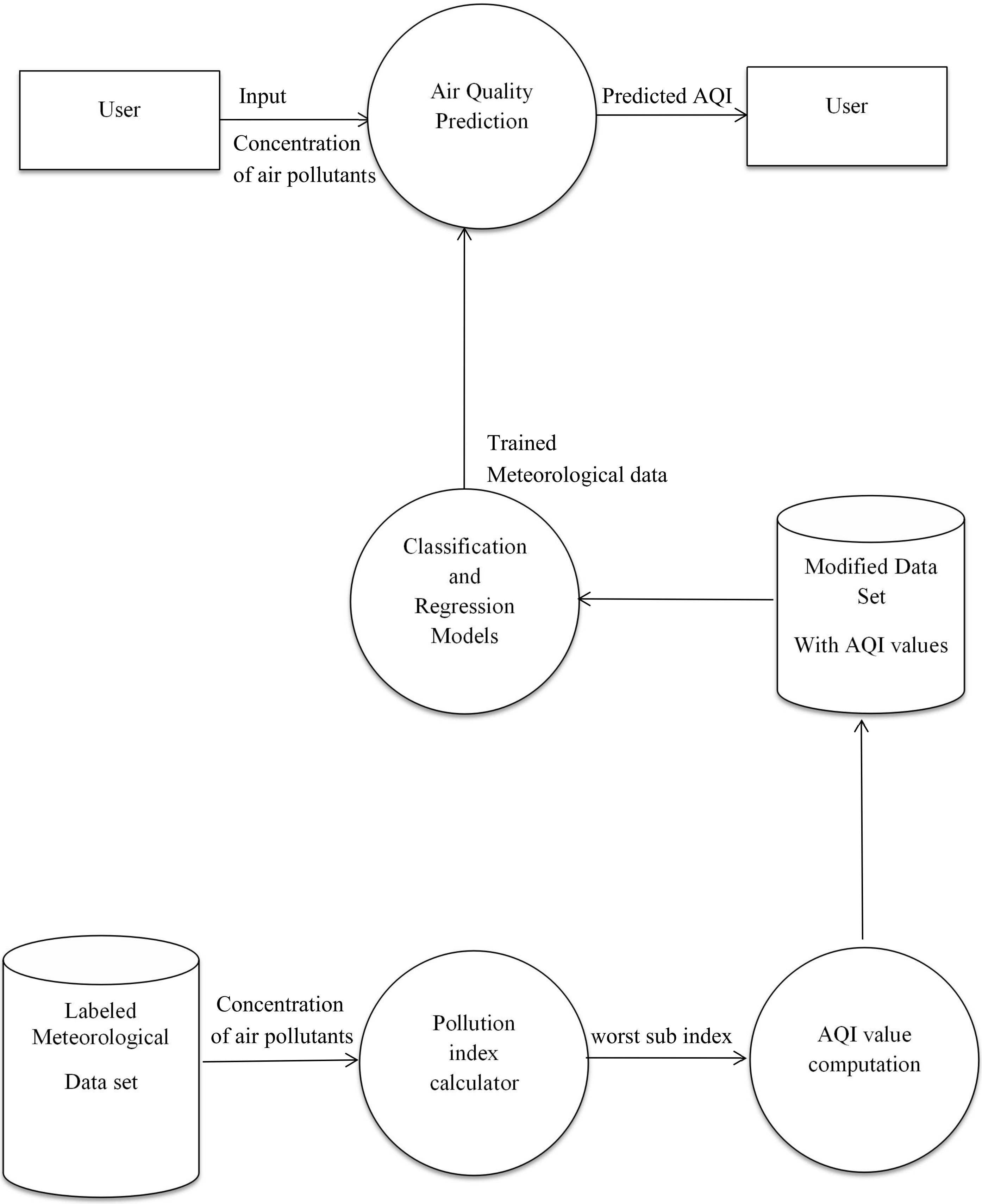
VI.1. Level 0 DFD:


User
Input
Concentration of air pollutants
Air Quality Prediction System
Predicted AQI
User

VI.2. Level 1 DFD:


User
Input
Concentration of air pollutants
Machine Learning model Predictors
Predicted AQI
User
Labeled Meteorological
Calculation of AQI value based on National
Modified Data set along with

VI.3. Level 2 DFD:


User
Input
Concentration of air pollutants
Air Quality Prediction
Predicted AQI
User
Trained Meteorological data
Classification and Regression Models
Modified Data Set
With AQI values
Labeled Meteorological
Data set
Concentration of air pollutants
Pollution index calculator
worst sub index
AQI value computation

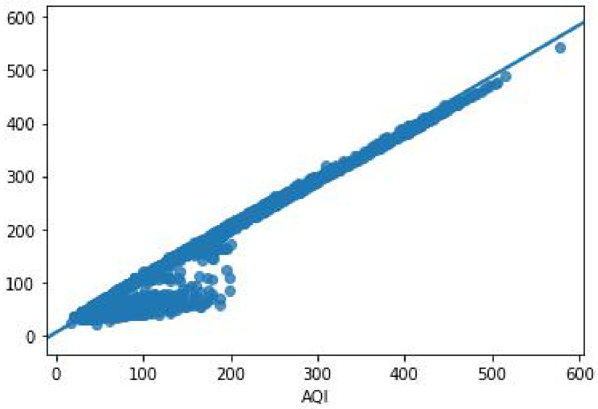
VII. RESULTS AND OBSERVATIONS

The Qualitative assessment of AQI value Prediction using various machine learning models is presented as follows:

Linear Regression model 1:

Intercept	14.6265	
Coefficients	SO_i	0.0164
	NOi	0.0284
	$RSPM_i$	0.0554
	SPM_i	0.9083
Accuracy score	0.964 or 96.4%	
R^2 value	0.96	
Mean square Error	17.31	

Regression plot of actual values vs predicted values:

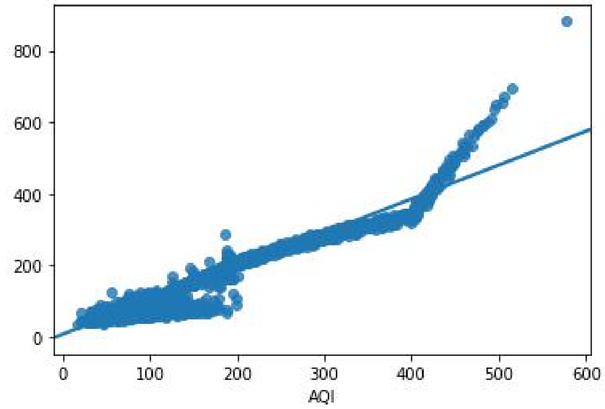


We can see that the model has a high accuracy and it is a good model for AQI prediction given the value of the individual pollution indices

Linear regression model 2:

Intercept	27.8448	
Coefficients	SO2	0.132
	NO2	0.1059
	RSPM	0.1250
	SPM	0.6486
Accuracy score	0.9515 or 95.15%	
R^2 value	0.95	
Mean square Error	20.11	

Regression plot of actual values vs predicted values:



We can see that the model has a high accuracy and is a good model for AQI prediction given the value of individual pollution indices, however the Linear Regression Model 1 is much better than the second model

The various other machine learning prediction models used and the corresponding accuracy scores are mentioned as follows:

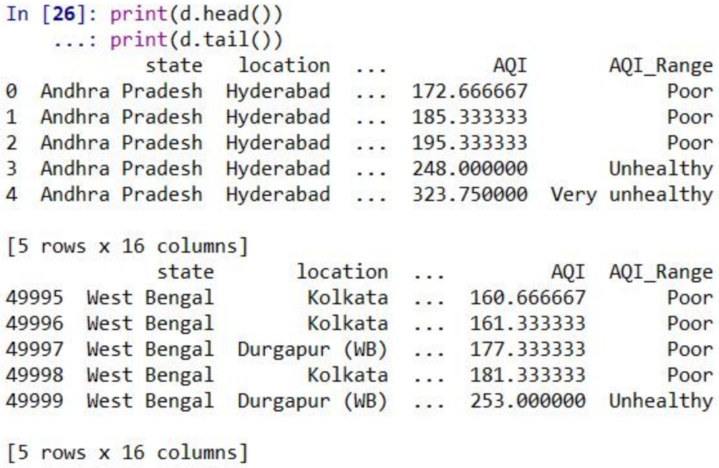
Model	Accuracy Score
Logistic Regression model 1	0.7572 or 75.72%

Logistic Regression model 2	0.7601 or 76.01%
Random Forest Classifier model	0.9996 or 99.96%
Decision Tree Classifier model	0.9998 or 99.98%

It is clear that model 1 and model 2 of Logistic Regression are not the good models for classification in this case because of the low accuracy score. However, the Random forest classifier model and Decision tree classifier model are very good models (with latter being the best) for classification in this case because of high accuracy score

Model	R^2 value	Mean Square Error
K Nearest Neighbour	-49.51	649.24

AQI score and AQI Range :



```
In [26]: print(d.head())
    ...: print(d.tail())
            state   location  ...         AQI       AQI_Range
0  Andhra Pradesh  Hyderabad  ...  172.666667            Poor
1  Andhra Pradesh  Hyderabad  ...  185.333333            Poor
2  Andhra Pradesh  Hyderabad  ...  195.333333            Poor
3  Andhra Pradesh  Hyderabad  ...  248.000000       Unhealthy
4  Andhra Pradesh  Hyderabad  ...  323.750000  Very unhealthy

[5 rows x 16 columns]
             state       location  ...         AQI  AQI_Range
49995  West Bengal        Kolkata  ...  160.666667       Poor
49996  West Bengal        Kolkata  ...  161.333333       Poor
49997  West Bengal  Durgapur (WB)  ...  177.333333       Poor
49998  West Bengal        Kolkata  ...  181.333333       Poor
49999  West Bengal  Durgapur (WB)  ...  253.000000  Unhealthy

[5 rows x 16 columns]
```

VIII. CONCLUSION AND FUTURE WORK

The high degree of correlation between AQI and the pollutants such as SO_2, NO_2, RSPM, SPM has been emphasized significantly. The amount of AQI has been incremented considerably over the years. It is high time to take preventive measures against high AQI and shun the damage it causes on human health. The models used can be extended further to include more pollutants for the comparatively accurate prediction of AQI value. The data set can also be trained using various models of deep learning such as the artificial neural networks (ANN), auto regressive non linear neural network etc., in order to obtain high degree of effectiveness. The proposed system will be useful for meteorological department and common people for the prediction the level pollution in air and take the required action accordingly. This system will also help the people to form a source of data for small localities which are not considered usually compared to larger cities.

IX. REFERENCES

[1] Kostandina Veljanovska , Angel Dimoski : The prediction of air quality index using machine learning models

[2] National air quality index data in air quality by central pollution control board(CPCB) and ministry of environment , Foresis and climate change.

[3] Ziyue Guan, Richard O sinnott : Air pollution prediction using approaches of machine learning on cloud

[4] Rozita Jailani, Ruhizan liza Ahmad Shauri : Ambient air quality prediction on the basis of neural network technique.

[5] Nayana D. K, Aditya CR, Praveen Gandhi, Vidyavastu , Chandana R Deshmukh : Air pollution detection and prediction through machine learning models.

[6] Aeroquas : The air quality index and the associated benefits.

[7] Acciona Renewable energy and sustainable infrastructures : Adverse effects of air pollution on human health.

[8] United States Environmental protection Agency : Particulate matter pollution.

APPENDIX

```
data = pd.read_csv('datam.csv',encoding='unicode_escape') #import data

print(data.head(10)) #print first 10 rows

print(data.tail(10)) #printing last 10 rows

print(data.columns) #print the columns/features of the data

print(data.describe()) #basic info of the dataset

print(data.shape) #dimensions of the data

data[['so2','state']].groupby(["state"]).mean().sort_values(by='so2').head(20).plot.bar(color='b')

plt.show()

data[['no2','state']].groupby(["state"]).mean().sort_values(by='no2').plot.bar(color='g')

plt.show()

data[['spm','state']].groupby(["state"]).mean().sort_values(by='spm').plot.bar(color='b')

plt.show()

data[['rspm','state']].groupby(["state"]).mean().sort_values(by='rspm').plot.bar(color='r')

plt.show()

data.isna().sum() #print the sum of null values for each columns
data.drop(['stn_code','agency','sampling_date','location_monitoring_station'],
axis=1,inplace=True)

print(data.head(10))

total = data.isnull().sum().sort_values(ascending=False)

print(total.head())

percent =
(data.isnull().sum()/data.isnull().count()*100).sort_values(ascending=False)

missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])

missing_data.head()

sns.barplot(x=missing_data.index, y=missing_data['Percent'])

plt.xlabel('Features', fontsize=20)

plt.ylabel('Percent of missing values', fontsize=20)

plt.title('Percent missing data by feature', fontsize=20)

data.groupby('state')[['spm','pm2_5','rspm','so2','no2']].mean()


plt.hist(data.spm,range=(0.0,4000)) #spm

plt.hist(data.so2,range=(0,1000)) #so2

plt.hist(data.no2,range=(0,1000)) #no2

plt.hist(data.rspm,range=(0,7000)) #rspm

plt.hist(data.pm2_5,range=(0,1000)) #pm2_5

grp_state = data.groupby('state')
```

```
def impute_mean_by_state(series):
    return series.fillna(series.mean())
data['rspm']=grp_state['rspm'].transform(impute_mean_by_state)   #fill value with mean value group by state
data['so2']=grp_state['so2'].transform(impute_mean_by_state)
data['no2']=grp_state['no2'].transform(impute_mean_by_state)
data['spm']=grp_state['spm'].transform(impute_mean_by_state)
data['pm2_5']=grp_state['pm2_5'].transform(impute_mean_by_state)
print(data.describe())
data.isna().sum() #some null value remains since some state have one value(i.e NaN only) and no mean to replace them
plt.hist(data.so2,range=(0,1000))
plt.hist(data.rspm,range=(0,7000))
plt.hist(data.no2,range=(0.0,1000))
plt.hist(data.spm,range=(0.0,4000)) #spm
print(data.tail(10))
def cal_SOi(so2):
   si=0
    if (so2<=40):
     si= so2*(50/40)
    elif (so2>40 and so2<=80):
     si= 50+(so2-40)*(50/40)

    elif (so2>80 and so2<=380):

     si= 100+(so2-80)*(100/300)

    elif (so2>380 and so2<=800):

     si= 200+(so2-380)*(100/420)

    elif (so2>800 and so2<=1600):

     si= 300+(so2-800)*(100/800)

    elif (so2>1600):

     si= 400+(so2-1600)*(100/800)

    return si

data['SOi']=data['so2'].apply(cal_SOi)

df= data[['so2','SOi']]

df.head()


def cal_Noi(no2):

    ni=0

    if(no2<=40):

     ni= no2*50/40

    elif(no2>40 and no2<=80):

     ni= 50+(no2-40)*(50/40)
    elif(no2>80 and no2<=180):
     ni= 100+(no2-80)*(100/100)
    elif(no2>180 and no2<=280):
     ni= 200+(no2-180)*(100/100)
    elif(no2>280 and no2<=400):
     ni= 300+(no2-280)*(100/120)
    else:
     ni= 400+(no2-400)*(100/120)
    return ni
data['Noi']=data['no2'].apply(cal_Noi)
df= data[['no2','Noi']]
df.head()
def cal_RSPMi(rspm):
    rpi=0
    if(rspm<=100):
     rpi = rspm
    elif(rspm>=101 and rspm<=150):
     rpi= 101+(rspm-101)*((200-101)/(150-101))
    elif(rspm>=151 and rspm<=350):
     ni= 201+(rspm-151)*((300-201)/(350-151))
    elif(rspm>=351 and rspm<=420):
     ni= 301+(rspm-351)*((400-301)/(420-351))
    elif(rspm>420):
     ni= 401+(rspm-420)*((500-401)/(420-351))
    return rpi
data['RSPMi']=data['rspm'].apply(cal_RSPMi)
df= data[['rspm','RSPMi']]
df.head()
def cal_SPMi(spm):
    spi=0
    if(spm<=50):
     spi=spm*50/50
    elif(spm>50 and spm<=100):
     spi=50+(spm-50)*(50/50)
    elif(spm>100 and spm<=250):
     spi= 100+(spm-100)*(100/150)
    elif(spm>250 and spm<=350):
     spi=200+(spm-250)*(100/100)
    elif(spm>350 and spm<=430):
     spi=300+(spm-350)*(100/80)
    else:
     spi=400+(spm-430)*(100/430)
    return spi
data['SPMi']=data['spm'].apply(cal_SPMi)
df= data[['spm','SPMi']]
df.head()
def cal_pmi(pm2_5):
    pmi=0
    if(pm2_5<=50):
     pmi=pm2_5*(50/50)
    elif(pm2_5>50 and pm2_5<=100):
     pmi=50+(pm2_5-50)*(50/50)
    elif(pm2_5>100 and pm2_5<=250):
     pmi= 100+(pm2_5-100)*(100/150)
    elif(pm2_5>250 and pm2_5<=350):
     pmi=200+(pm2_5-250)*(100/100)
    elif(pm2_5>350 and pm2_5<=450):
     pmi=300+(pm2_5-350)*(100/100)
    else:
     pmi=400+(pm2_5-430)*(100/80)
    return pmi

data['PMi']=data['pm2_5'].apply(cal_pmi)

df= data[['pm2_5','PMi']]

df.head()

type(data['PMi'])

def cal_aqi(si,ni,rspmi,spmi):

    aqi=0

    if(si>ni and si>rspmi and si>spmi):

     aqi=si

    if(ni>si and ni>rspmi and ni>spmi ):

     aqi=ni

    if(rspmi>si and rspmi>ni and rspmi>spmi ):

     aqi=rspmi

    if(spmi>si and spmi>ni and spmi>rspmi):

     aqi=spmi

    return aqi

data['AQI']=data.apply(lambda
x:cal_aqi(x['SOi'],x['Noi'],x['RSPMi'],x['SPMi']),axis=1)

df= data[['state','SOi','Noi','RSPMi','SPMi','AQI']]

print(df.head())

print(data.head())
```

```
def AQI_Range(x):
    if x<=50:
        return "Good"
    elif x>50 and x<=100:
        return "Moderate"
    elif x>100 and x<=200:
        return "Poor"
    elif x>200 and x<=300:
        return "Unhealthy"
    elif x>300 and x<=400:
        return "Very unhealthy"
    elif x>400:
        return "Hazardous"


data['AQI_Range'] = data['AQI'] .apply(AQI_Range)
data.head()
d=data #saving data in new value
d.head()
data=data.dropna(subset=['spm']) #spm
data=data.dropna(subset=['pm2_5']) #spm
```

```
data.isna().sum() #all null values removed

"""

Linear Regression

"""

X = data[['SOi','Noi','RSPMi','SPMi']]

y = data['AQI']

y.head()

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,random_state=101)

X_train.head()

LR = LinearRegression()

LR.fit(X_train, y_train)

print('Linear Regression')

print('Intercept',LR.intercept_)

print('Coefficients',LR.coef_)

prediction = LR.predict(X_test)

plt.scatter(y_test,prediction) #scatter plot for actual and predicted values

plt.xlabel('Y Test')

plt.ylabel('Predicted Y')

LR.predict([[9.1,16.3,67,179]])

sns.regplot(y_test,prediction) #regression plot
```

```
y_test_np= np.array(y_test)

prediction_np = np.array(prediction)

print(LR.score(X_test,y_test)) #accuracy score 76.82%

print('R^2_Square:%.2f '% r2_score(y_test, prediction))

print('MSE:%.2f '% np.sqrt(mean_squared_error(y_test, prediction)))

X1= data[['so2','no2','rspm','spm']]

y1 = data['AQI']

y.tail()

X_train1, X_test1, y_train1, y_test1 = train_test_split(X1,y1, test_size=0.2,random_state=101)

X_train1.head()

LR1 = LinearRegression()

LR1.fit(X_train1, y_train1)

print('Intercept',LR1.intercept_)

print('Coefficients',LR1.coef_)

prediction1 = LR1.predict(X_test1)

plt.scatter(y_test1,prediction1) #scatter plot for actual and predicted values

plt.xlabel('Y Test')

plt.ylabel('Predicted Y')

LR1.predict([[9.1,16.3,67,179]])

sns.regplot(y_test1,prediction1) #regression plot
```

```
y_test1_np= np.array(y_test1)

prediction1_np = np.array(prediction1)

print(LR1.score(X_test1,y_test1))

print('R^2_Square:%.2f '% r2_score(y_test1, prediction1))

print('MSE:%.2f '% np.sqrt(mean_squared_error(y_test1, prediction1)))




"""

Logistic Regression

"""

X2 = data[['SOi','Noi','RSPMi','SPMi']]

y2 = data['AQI_Range']

X_train2, X_test2, y_train2, y_test2 = train_test_split(X2, y2, test_size=0.33, random_state=42)

logmodel = LogisticRegression()

logmodel.fit(X_train2,y_train2)

predictions = logmodel.predict(X_test)

print(logmodel.score(X_test2,y_test2))

new = pd.DataFrame(d)

file1 = 'new1.csv'

new.to_csv(file1,index=True)
```

```
d.tail()

print(logmodel.predict([[77.4,147.7,78.182,100]])) #correct

print(logmodel.predict([[32.7,35,78.182,203]])) #correct

print(logmodel.predict([[100,182.2,78.182,400]])) #correct

X3 = data[['so2','no2','rspm','spm']]

y3 = data['AQI_Range']

X_train3, X_test3, y_train3, y_test3 = train_test_split(X3, y3, test_size=0.33,
random_state=42)

logmodel2 = LogisticRegression()

logmodel2.fit(X_train3,y_train3)

logmodel2.score(X_test3,y_test3) #very low accuracy score

print(logmodel2.predict([[4.8,17.4,78.48,200]])) #correct

print(logmodel2.predict([[67.4,127.7,78.48,215]])) #correct

print(logmodel2.predict([[2.059,8.94,102,256]])) #wrong

"""

Random Forest

"""

model = RandomForestClassifier(n_estimators=10)

model.fit(X_train3,y_train3)

model.score(X_test3,y_test3) #high accuracy score of 99.97 %

X_train3.head()
```

```
print(model.predict([[2.059,8.94,102,256]]))#correct

"""

Decision Tree

"""

model2 = tree.DecisionTreeClassifier()

model2.fit(X_train3,y_train3)

model2.score(X_test3,y_test3) #high accuracy score of 99.98%

print(model2.predict([[9,31,51,205.25]])) # correct

print(model2.predict([[2,5.8,17,36]])) # correct

print(model2.predict([[18.6,48.3,142,285]])) # correct

print(model2.predict([[6,11,109,84.41]])) # correct

print(model2.predict([[10,16,156,372.66]]))# correct

"""

KNN

"""

X0 = data[['SOi','Noi','RSPMi','SPMi']]

y0 = data['AQI']

y0.head()

X0_train, X0_test, y0_train, y0_test = train_test_split(X0,y0, test_size=0.2,random_state=101)

lab_enc = preprocessing.LabelEncoder()
```

```
y_encoded = lab_enc.fit_transform(y0_train)

knn = KNeighborsClassifier(n_neighbors=3,leaf_size=30)

knn.fit(X0_train,y_encoded)

y0_pred = knn.predict(X0_test)

predictions0=knn.predict(X0_test)

print('R^2_Square:%.2f '% r2_score(y0_test, predictions0))

print('MSE:%.2f '% np.sqrt(mean_squared_error(y0_test, predictions0)))

print(d.head())

print(d.tail())
```