Random Forest Algorithm- An application to enhance Air Quality

L Sreedhar[1], N.Aditya Sundar[2], Ch.D.Naidu[3], D.RaghaVendra[4]
[1]Associate Professor, [2]Assistant Professor, [3]Sr.Assistant Professor, [4]IV/IV B.Tech IT
[1]Department of Computer Science Engineering, [2,3]Department of Information Technology
[1]Raghu Engineering College – (REC)
[2.34]ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
Vishakapatnam (District), Andhra Pradesh, India

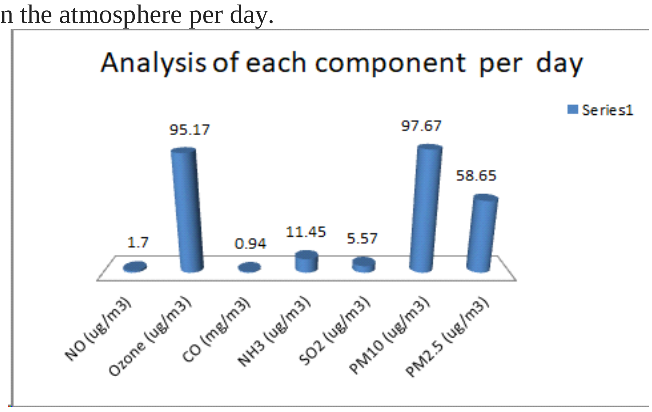
Abstract- Air quality is a vital necessity on the earth, it's helps us to live on this planet.Everyday air is polluted heavily because ofindustries releasing the gases in the atmosphere, burning of e-waste, polluted exhaust from vehicles.It's impact on public health causes diseases like cancer, asthma,lung diseases etc. A study analysis has been done,by observing the increase of day to day air pollution, considering the data from the pollution boardpertaining to two different years.Analysis was done on this data using Random Forest Algorithm implemented in R programming language,to observe the error rate between the two years of data so as to identify the component which impacts the environment and therefore public health.

I. INTRODUCTION

Air pollution is mixtures of various gases, particles and other activities. In this paper, we discuss about the breakpoints and the error rate among the data sets of the two different years. We apply the Random Forest algorithm [11] to calculate the error rate by applying the different formulas mentioned below using R programming language [2]. We calculate the breakpoint concentrations with the help of breakpoint table and different levels of AQI [6]. Air Quality Index is used to measure the air pollutants with in the region. There are many methods in calculating the concentrations and different methods that have been used by different countries in calculating the breakpoint concentration levels of each individual pollutants. We have taken the data observed around a 24 hrs range. By considering this,we get to know not only concentrations, but also the pollutants that cause health effects, such as toxic pollutants, dangerous gases etc.

II. METHODOLOGY

In order to understand the rise of air pollution in a particular region, a real time air quality monitoring was taken at the control board site for every region. In the present study we are going to discussed about radiations [9] that are causes due to the pollutants released in the atmosphere such as PM10, PM2.5, NO2, SO2, O3 and NH3.For this we have used the technique known as Beta Attenuation Monitoring (BAM). It is a widely used air monitoring technique for the absorption of beta radiation by solid particles extracted from the air.This technique [2] allows for the detection of PM10 and PM2.5.The main principle is based on Bouguer (Lambert–Beer) law: the amount by which the flow of beta radiation (electrons) is attenuated by a solid matter is exponentially dependent on its mass and not on any other feature (such as density, chemical composition or some optical or electrical properties) of this matter, and the component O3 is monitored based on the concentration that are released in the air .We take the index based on hourly or 8 hrs accurately[9]. For SO2 known as UV fluorescence, based on the emission of light by SO2 molecule observed by UV radiations is taken. While for NO2 the amount of nitrogen oxide in the air is measured by "Chemiluminescence Analyzer" [6]. It is a standard method for measuring NO concentration.Chemiluminescence relies on the reaction of NO with ozone. The chemical oxidation of NO by ozone yields nitrogen dioxide in an excited state. The calibration is undertaken by traceable standard reference gas method.In the below figure represent the pollutants consumes in the atmosphere per day.



Methods I:

In this method the Air Quality Index (AQI) calculated through the concept of arithmetic mean along with the concentration [1] of air pollutants to the standard value of those concerned pollutants such as PM2.5, PM10, SO2, NO2, CO, O3 and NH3. From the obtained value, say the average is multiplied by 100 to get the index value.Thus, Air Quality Index is compared with the rating scale [9]. We can calculate AQI for the pollutants individually by the following formula

$$\mathbf{AQI = (C/Cs) * 100}$$

Where AQI = Air Quality Index

C= the observed value of the air quality parameters pollutant (PM10, PM2.5, NO2 and SO2)

Cs= CPCB standard for residential Area

Method II:

In this method the Air Quality Index is calculated by observing the geometric mean to the ratio of the concentration of each pollutant to the standard value of that pollutants [2] such as PM10, PM2.5, SO2, NO2, CO and NH3.And the Air Quality Index is compared with the rating scale.

Method III:

Air Quality Index was done for combining qualitative measures with qualitative concept of the environment [5][7]. The individual air quality index here is calculated as follow:

$$AQi = (W * C)/Cs$$

Where AQI = Air Quality Index

W= Weighted of Pollutant

C= the observed value of the air quality parameters pollutant (PM10, PM2.5, NO2 and SO2) Cs= CPCB standard for residential Area

Method IV:

In this method, to calculate the AQI from the concentration of each individual pollutants were based on the breakpoint concentration table [8].The individual air quality index for a given pollutant concentration (Cs) as based on linear segmented principle is calculated as

$$Ip = [\{(Ihl - ILo)/(BhI - BLO) * (Cp - BLo)\}] + ILo$$

Where BhI= Breakpoint concentration greater or equal to given concentration

BLo=Breakpoint concentration smaller or equal to given concentration

Cp= The pollutant concentration

Ihl=AQI value corresponding to BhI

ILo=AQI value corresponding to BLo

Note: In order to calculate Index of the multiple pollutants, we have to calculate the index of eachpollutant, hence that pollutant with the highest value,or the "responsible pollutant/s" will determine the Air Quality Index and the category.

III. RESULT

The continuous monitoring of the air pollutants is undertaken from the GVMC corporation, Visakhapatnam city by the observation of Central Pollution Control Board during March to November 2017 with the reference to PM10, PM2.5, SO2, CO, NO2, O3 and NH3.Data obtained from the pollution board is used to calculate the air quality index (air pollution index) for critical parameter.

Air Quality Index (AQI):

In the present days every in the society must have the awareness on the current air pollution levels and they must compare the past and present levels of air pollution. Air Quality Index is a tool, used to know the status of daily air quality along with the basic standards proposed by the pollution board in order to control the pollution levels. It gives an idea, how the index rate is increasing daily and tells the public to understand how clean the air to breathe daily. In our country we use Central Pollution Control Board standards in calculating the Air Quality Index from the data present by them. The AQI of specific pollutant is derived mainly from the physical measurement of pollutant like PM10, PM2.5, NO2, CO, NH3, O3 and SO2 etc. In the present study, different methods were used to calculate ambient air quality index.

Different AQI were estimated for various months and varying results were observed ranging from good to unacceptable for the same set of data. This may be due to eclipsing effect of the values used in the formulas. The statistical theory behind these AQI makes it more prone to variations i.e the use of means from simple arithmetic to logarithmic and weighted averages [5] to use of breakpoint concentration as basis of estimation. The breakpoint concentration based AQI is more robust and can be used for decision making. Accordingly, the AQI values are calculated based on Breakpointconcentration [2] for 24 hourly averages for PM10, PM2.5, SO2, CO, O3 and NO2 concentrations and are categorized as satisfactory to moderate during the study

The below figures represent the analysis combining data of 2 months, in2018

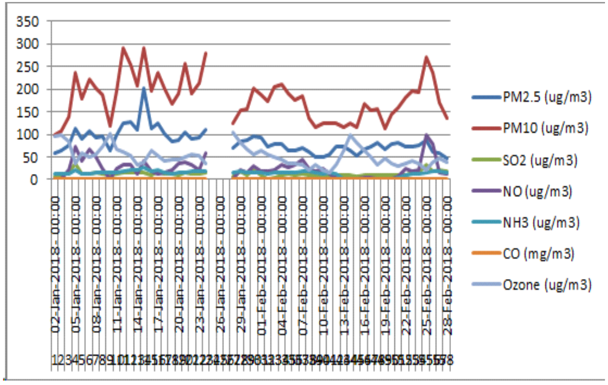
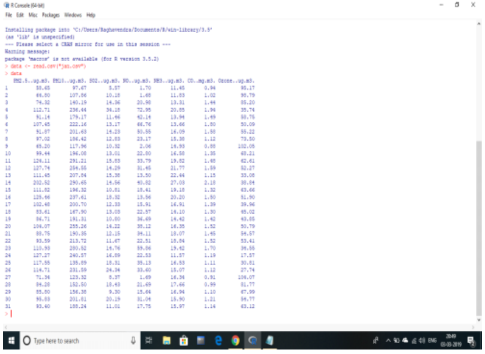


Fig.1: Represent the Jan-Feb ,2018

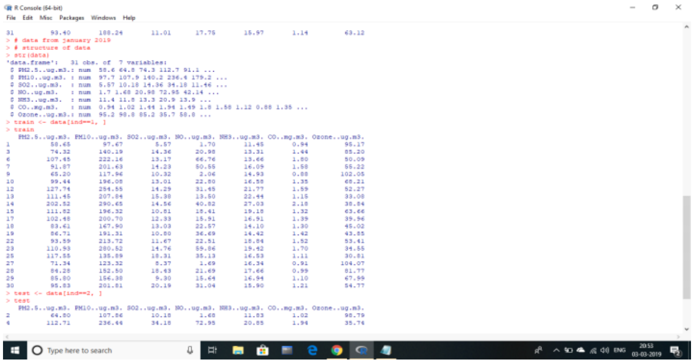
Below figures represent the analysis done through "Random Forest Algorithm" in R programming language.



```
Installing package into 'C:/Users/Raghavendra/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
Warning message:
package 'macros' is not available (for R version 3.5.2)
> data <- read.csv("jan.csv")
> data
   PM2.5..ug.m3. PM10..ug.m3. SO2..ug.m3. NO..ug.m3. NH3..ug.m3. CO..mg.m3. Ozone..ug.m3.
1          58.65        97.47        5.57       1.70       11.45       0.94         95.17
2          66.80       107.86       10.18       1.68       11.83       1.02         98.79
3          74.32       140.19       14.36      20.98       13.31       1.44         85.20
4         112.71       236.44       36.18      72.98       20.85       1.94         35.74
5          91.14       179.17       11.46      42.14       13.94       1.49         58.76
6         107.45       222.16       13.17      66.76       13.66       1.80         50.09
7          91.87       201.63       14.23      50.85       16.09       1.58         55.22
8          97.02       186.42       12.83      23.17       15.38       1.12         73.50
9          65.20       117.96       10.32       2.06       16.93       0.88        102.05
10         99.44       196.08       13.01      22.80       16.58       1.35         68.21
11        126.11       291.21       15.83      33.79       19.82       1.48         62.61
12        127.74       254.66       14.29      31.46       21.77       1.59         52.27
13        111.45       207.84       15.38      13.50       22.44       1.15         33.08
14        202.52       290.65       14.56      40.82       27.03       2.18         38.84
15        111.82       196.32       10.81      18.41       19.18       1.32         63.66
16        125.46       237.61       18.32      13.56       20.20       1.50         51.90
17        102.48       200.70       12.33      15.91       16.91       1.39         39.96
18         83.61       167.90       13.03      22.57       16.10       1.30         65.02
19         86.71       191.31       10.80      36.69       14.42       1.42         43.65
20        106.07       255.26       14.22      35.12       16.35       1.52         50.79
21         88.76       190.86       12.16      34.11       18.07       1.46         54.67
22         93.59       213.72       11.67      22.51       18.84       1.52         53.41
23        110.98       280.52       14.76      59.86       19.42       1.70         36.66
24        127.27       240.57       16.89      22.53       11.57       1.19         17.57
25        117.58       135.89       18.31      35.13       16.53       1.11         30.81
26        114.71       231.59       24.34      33.60       15.07       1.12         27.74
27         71.36       123.32        8.37       1.69       16.36       0.91        106.07
28         84.28       152.50       18.48      21.69       17.66       0.99         81.77
29         88.80       156.38        9.30      15.64       16.94       1.10         67.99
30         95.83       201.81       20.19      31.04       15.90       1.21         54.77
31         93.40       188.24       11.01      17.78       15.97       1.14         63.12
>
```

Fig.2: Shows the data set, Jan- Dec2018

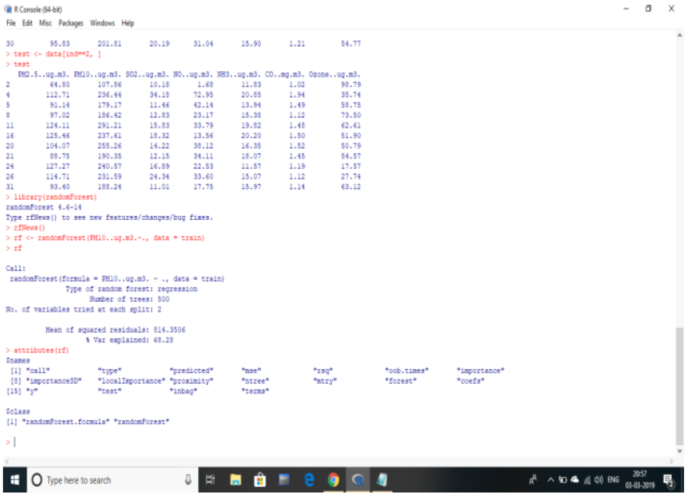
The above picture shows the data that is taken from the pollution board, represent the different components such as PM10, PM2.5, NO, CO, VSO2, NH3, OZONE.



```
31          93.40       188.24       11.01       17.75       15.97       1.14          63.12
> # data from january 2019
> # structure of data
> str(data)
'data.frame':   31 obs. of  7 variables:
 $ PM2.5..ug.m3.: num  58.6 64.8 74.3 112.7 91.1 ...
 $ PM10..ug.m3. : num  97.7 107.9 140.2 236.4 179.2 ...
 $ SO2..ug.m3.  : num  5.57 10.18 14.36 34.18 11.46 ...
 $ NO..ug.m3.   : num  1.7 1.68 20.98 72.95 42.14 ...
 $ NH3..ug.m3.  : num  11.4 11.8 13.3 20.9 13.9 ...
 $ CO..mg.m3.   : num  0.94 1.02 1.44 1.94 1.49 1.8 1.58 1.12 0.88 1.35 ...
 $ Ozone..ug.m3.: num  95.2 98.8 85.2 35.7 58.8 ...
> train <- data[ind==1, ]
> train
   PM2.5..ug.m3. PM10..ug.m3. SO2..ug.m3. NO..ug.m3. NH3..ug.m3. CO..mg.m3. Ozone..ug.m3.
1          58.65        97.67        5.57       1.70       11.45       0.94          95.17
3          74.32       140.19       14.36      20.98       13.31       1.44          85.20
6         107.45       222.16       13.17      66.76       13.66       1.80          50.09
7          91.87       201.63       14.23      50.55       16.09       1.58          55.22
9          65.20       117.96       10.32       2.06       14.93       0.88         102.05
10         99.44       196.08       13.01      22.80       16.58       1.35          68.21
12        127.74       254.55       14.29      31.45       21.77       1.59          52.27
13        111.45       207.84       15.38      13.50       22.44       1.15          33.08
14        202.52       290.65       14.56      40.82       27.03       2.18          38.84
15        111.82       196.32       10.81      18.41       19.18       1.32          63.66
17        102.48       200.70       12.33      15.91       16.91       1.39          39.96
18         83.61       167.90       13.03      22.57       14.10       1.30          45.02
19         86.71       191.31       10.80      36.69       14.42       1.42          43.85
22         93.59       213.72       11.67      22.51       18.84       1.52          53.41
23        110.93       280.52       14.76      59.86       19.42       1.70          34.55
25        117.55       135.89       18.31      35.13       16.53       1.11          30.81
27         71.34       123.32        8.37       1.69       16.34       0.91         104.07
28         84.28       152.50       18.43      21.69       17.66       0.99          81.77
29         85.80       156.38        9.30      15.64       16.94       1.10          67.99
30         95.83       201.81       20.19      31.04       15.90       1.21          54.77
> test <- data[ind==2, ]
> test
   PM2.5..ug.m3. PM10..ug.m3. SO2..ug.m3. NO..ug.m3. NH3..ug.m3. CO..mg.m3. Ozone..ug.m3.
2          64.80       107.86       10.18       1.68       11.83       1.02          98.79
4         112.71       236.44       34.18      72.95       20.85       1.94          35.74
```

Fig.3: Represent the structure and character of table

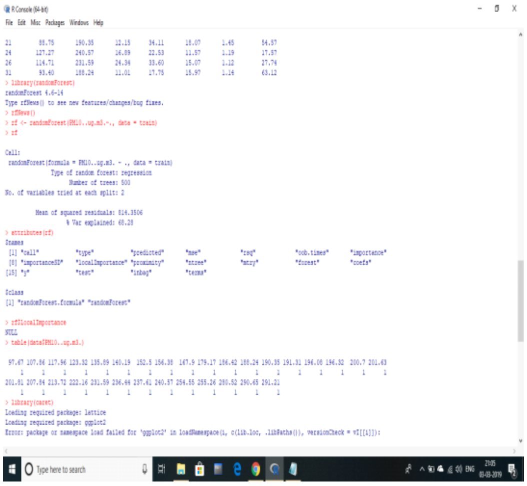
And the table represent the structure and type of components involved in the data. It tells the total number of observation and the variables in the table.



```
30          95.83       201.81       20.19       31.04       15.90       1.21          54.77
> test <- data[ind==2, ]
> test
   PM2.5..ug.m3. PM10..ug.m3. SO2..ug.m3. NO..ug.m3. NH3..ug.m3. CO..mg.m3. Ozone..ug.m3.
2          64.80       107.86       10.18       1.68       11.83       1.02          98.79
4         112.71       236.44       34.18      72.95       20.85       1.94          35.74
5          91.14       179.17       11.46      42.14       13.94       1.49          58.75
8          97.02       186.42       12.83      23.17       15.38       1.12          73.50
11        124.11       291.21       15.83      33.79       19.82       1.48          62.61
16        125.46       237.61       18.32      13.56       20.20       1.50          51.90
20        104.07       255.26       14.22      38.12       16.35       1.52          50.79
21         88.75       190.35       12.15      34.11       18.07       1.45          54.57
24        127.27       240.57       16.89      22.53       11.57       1.19          17.57
26        114.71       231.59       24.34      33.60       15.07       1.12          27.74
31         93.40       188.24       11.01      17.75       15.97       1.14          63.12
> library(randomForest)
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
> rfNews()
> rf <- randomForest(PM10..ug.m3.~., data = train)
> rf

Call:
 randomForest(formula = PM10..ug.m3. ~ ., data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 814.3506
                    % Var explained: 68.28
> attributes(rf)
$names
 [1] "call"            "type"            "predicted"       "mse"             "rsq"             "oob.times"       "importance"
 [8] "importanceSD"    "localImportance" "proximity"       "ntree"           "mtry"            "forest"          "coefs"
[15] "y"               "test"            "inbag"           "terms"

$class
[1] "randomForest.formula" "randomForest"

> |
```

Fig.4: Testing and Training data using Random Forest Model

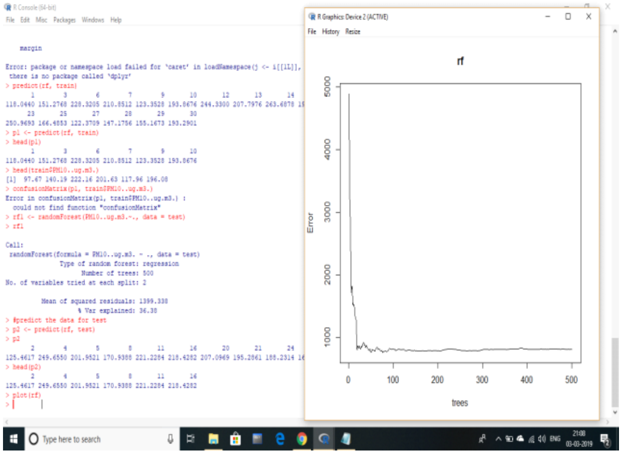
In order to use the random Forest model, we need two types of data sets. One is training data and the other one is testing data. We represent these data's by applying Random Forest Algorithm.



```
21         88.75       190.35       12.15      34.11       18.07       1.45          54.57
24        127.27       240.57       16.89      22.53       11.57       1.19          17.57
26        114.71       231.59       24.34      33.60       15.07       1.12          27.74
31         93.40       188.24       11.01      17.75       15.97       1.14          63.12
> library(randomForest)
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
> rfNews()
> rf <- randomForest(PM10..ug.m3.~., data = train)
> rf

Call:
 randomForest(formula = PM10..ug.m3. ~ ., data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 814.3506
                    % Var explained: 68.28
> attributes(rf)
$names
 [1] "call"            "type"            "predicted"       "mse"             "rsq"             "oob.times"       "importance"
 [8] "importanceSD"    "localImportance" "proximity"       "ntree"           "mtry"            "forest"          "coefs"
[15] "y"               "test"            "inbag"           "terms"

$class
[1] "randomForest.formula" "randomForest"

> rf$localImportance
NULL
> table(data$PM10..ug.m3.)

 97.67 107.86 117.96 123.32 135.89 140.19  152.5 156.38  167.9 179.17 186.42 188.24 190.35 191.31 196.08 196.32  200.7 201.63
     1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1
201.81 207.84 213.72 222.16 231.59 236.44 237.61 240.57 254.55 255.26 280.52 290.65 291.21
     1      1      1      1      1      1      1      1      1      1      1      1      1
> library(caret)
Loading required package: lattice
Loading required package: ggplot2
Error: package or namespace load failed for 'ggplot2' in loadNamespace(i, c(lib.loc, .libPaths()), versionCheck = vI[[i]]):
```

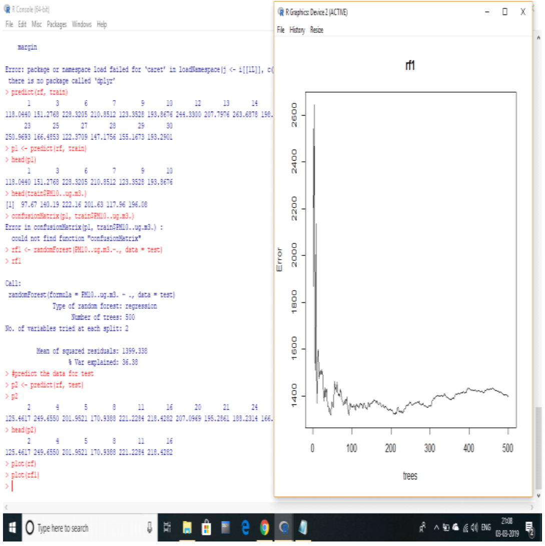
Fig.5: Applying Random Forest Algorithm

By considering the above-mentioned data's, testing and training data we calculate the error rate among the data sets by applying Random Forest Algorithm through R programming language.



```
   margin

Error: package or namespace load failed for 'caret' in loadNamespace(j <- i[[1L]],
 there is no package called 'dplyr'
> predict(rf, train)
       1        3        6        7        9       10       12       13       14
118.0440 151.2768 228.3205 210.8512 123.3528 193.8676 244.3300 207.7976 263.6878 1
      23       25       27       28       29       30
250.9693 166.4853 122.3709 147.1756 155.1673 193.2901
> p1 <- predict(rf, train)
> head(p1)
       1        3        6        7        9       10
118.0440 151.2768 228.3205 210.8512 123.3528 193.8676
> head(train$PM10..ug.m3.)
[1]  97.67 140.19 222.16 201.63 117.96 196.08
> confusionMatrix(p1, train$PM10..ug.m3.)
Error in confusionMatrix(p1, train$PM10..ug.m3.) :
  could not find function "confusionMatrix"
> rf1 <- randomForest(PM10..ug.m3.~., data = test)
> rf1

Call:
 randomForest(formula = PM10..ug.m3. ~ ., data = test)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 1399.338
                    % Var explained: 36.38
> #predict the data for test
> p2 <- predict(rf, test)
> p2
       2        4        5        8       11       16       20       21       24
125.4617 249.6550 201.9521 170.9388 221.2284 218.4282 207.0969 195.2861 188.2314 16
> head(p2)
       2        4        5        8       11       16
125.4617 249.6550 201.9521 170.9388 221.2284 218.4282
> plot(rf)
> |
```

Fig.6: Plotting the graph for the training data using RF Algorithm

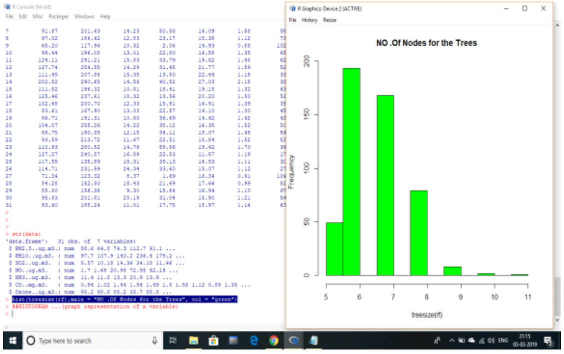


```
   margin

Error: package or namespace load failed for 'caret' in loadNamespace(j <- i[[1L]], c(
 there is no package called 'dplyr'
> predict(rf, train)
       1        3        6        7        9       10       12       13       14
118.0440 151.2768 228.3205 210.8512 123.3528 193.8676 244.3300 207.7976 263.6878 198.
      23       25       27       28       29       30
250.9693 166.4853 122.3709 147.1756 155.1673 193.2901
> p1 <- predict(rf, train)
> head(p1)
       1        3        6        7        9       10
118.0440 151.2768 228.3205 210.8512 123.3528 193.8676
> head(train$PM10..ug.m3.)
[1]  97.67 140.19 222.16 201.63 117.96 196.08
> confusionMatrix(p1, train$PM10..ug.m3.)
Error in confusionMatrix(p1, train$PM10..ug.m3.) :
  could not find function "confusionMatrix"
> rf1 <- randomForest(PM10..ug.m3.~., data = test)
> rf1

Call:
 randomForest(formula = PM10..ug.m3. ~ ., data = test)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 1399.338
                    % Var explained: 36.38
> #predict the data for test
> p2 <- predict(rf, test)
> p2
       2        4        5        8       11       16       20       21       24
125.4617 249.6550 201.9521 170.9388 221.2284 218.4282 207.0969 195.2861 188.2314 166.
> head(p2)
       2        4        5        8       11       16
125.4617 249.6550 201.9521 170.9388 221.2284 218.4282
> plot(rf)
> plot(rf1)
> |
```

6:

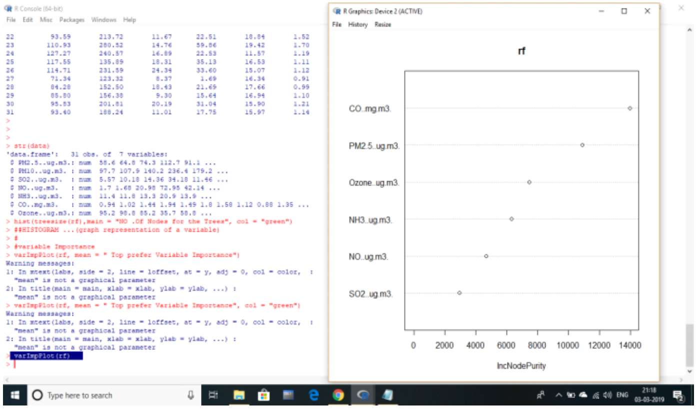
Fig.7: Plotting the graph for the testing data using RF Algorithm

Above shown graphs Fig 6 & Fig 7 are represented through R programming language (following command “plot (filename)” for both training and testing data sets, indicates the levels of each components for the two different years 2018 & 2019.


R Console (64-bit)
File Edit Misc Packages Windows Help
R Graphics: Device 2 (ACTIVE)
File History Resize
NO .Of Nodes for the Trees
Frequency
0
50
100
150
200
5
6
7
8
9
10
11
treesize(rf)

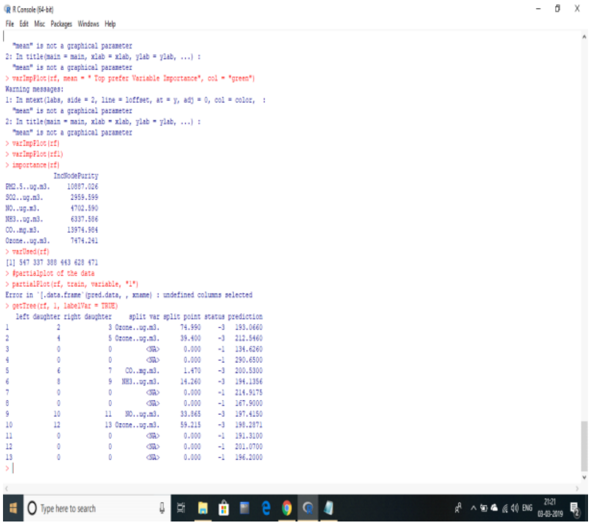
```
7          91.87     201.63     14.23     50.55     16.09     1.58     58
8          97.02     186.42     12.83     23.17     15.38     1.12     73
9          65.20     117.96     10.32      2.06     14.93     0.88    102
10         99.44     196.08     13.01     22.80     16.58     1.35     68
11        124.11     291.21     15.83     33.79     19.82     1.48     62
12        127.74     254.55     14.29     31.45     21.77     1.59     52
13        111.45     207.84     15.38     13.50     22.44     1.15     33
14        202.52     290.65     14.56     40.82     27.03     2.18     38
15        111.82     196.32     10.81     18.41     19.18     1.32     63
16        125.46     237.61     18.32     13.56     20.20     1.50     51
17        102.48     200.70     12.33     15.91     16.91     1.39     35
18         83.61     167.90     13.03     22.57     14.10     1.30     45
19         86.71     191.31     10.80     36.69     14.42     1.42     43
20        104.07     255.26     14.22     38.12     16.35     1.52     50
21         88.75     190.35     12.15     34.11     18.07     1.65     54
22         93.59     213.72     11.67     22.51     18.84     1.52     53
23        110.93     280.52     14.76     59.86     19.42     1.70     34
24        127.27     240.57     16.89     22.53     11.57     1.19     17
25        117.55     135.89     18.31     35.13     16.53     1.11     30
26        114.71     231.59     24.34     33.60     15.07     1.12     27
27         71.34     123.32      8.37      1.69     16.34     0.91    104
28         84.28     152.50     18.43     21.69     17.66     0.99     81
29         85.80     156.38      9.30     15.64     16.94     1.10     67
30         95.83     201.81     20.19     31.04     15.90     1.21     54
31         93.40     188.24     11.01     17.75     15.97     1.14     63
>
>
>
> str(data)
'data.frame':   31 obs. of  7 variables:
 $ PM2.5..ug.m3.: num  58.6 64.8 74.3 112.7 91.1 ...
 $ PM10..ug.m3. : num  97.7 107.9 140.2 236.4 179.2 ...
 $ SO2..ug.m3.  : num  5.57 10.18 14.36 34.18 11.46 ...
 $ NO..ug.m3.   : num  1.7 1.68 20.98 72.95 42.14 ...
 $ NH3..ug.m3.  : num  11.4 11.8 13.3 20.9 13.9 ...
 $ CO..mg.m3.   : num  0.94 1.02 1.44 1.94 1.49 1.8 1.58 1.12 0.88 1.35 ...
 $ Ozone..ug.m3.: num  95.2 98.8 85.2 35.7 58.8 ...
> hist(treesize(rf),main = "NO .Of Nodes for the Trees", col = "green")
> ##HISTOGRAM ...(graph representation of a variable)
>
```

Fig 8: Plotting the Histogram graph for the training data using RF Algorithm, Jan-2018..



```
22         93.59     213.72     11.67     22.51     18.84     1.52
23        110.93     280.52     14.76     59.86     19.42     1.70
24        127.27     240.57     16.89     22.53     11.57     1.19
25        117.55     135.89     18.31     35.13     16.53     1.11
26        114.71     231.59     24.34     33.60     15.07     1.12
27         71.34     123.32      8.37      1.69     16.34     0.91
28         84.28     152.50     18.43     21.69     17.66     0.99
29         85.80     156.38      9.30     15.64     16.94     1.10
30         95.83     201.81     20.19     31.04     15.90     1.21
31         93.40     188.24     11.01     17.75     15.97     1.14
>
>
>
> str(data)
'data.frame':   31 obs. of  7 variables:
 $ PM2.5..ug.m3.: num  58.6 64.8 74.3 112.7 91.1 ...
 $ PM10..ug.m3. : num  97.7 107.9 140.2 236.4 179.2 ...
 $ SO2..ug.m3.  : num  5.57 10.18 14.36 34.18 11.46 ...
 $ NO..ug.m3.   : num  1.7 1.68 20.98 72.95 42.14 ...
 $ NH3..ug.m3.  : num  11.4 11.8 13.3 20.9 13.9 ...
 $ CO..mg.m3.   : num  0.94 1.02 1.44 1.94 1.49 1.8 1.58 1.12 0.88 1.35 ...
 $ Ozone..ug.m3.: num  95.2 98.8 85.2 35.7 58.8 ...
> hist(treesize(rf),main = "NO .Of Nodes for the Trees", col = "green")
> ##HISTOGRAM ...(graph representation of a variable)
> #
> #variable Importance
> varImpPlot(rf, mean = " Top prefer Variable Importance")
Warning messages:
1: In mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color,  :
  "mean" is not a graphical parameter
2: In title(main = main, xlab = xlab, ylab = ylab, ...) :
  "mean" is not a graphical parameter
> varImpPlot(rf, mean = " Top prefer Variable Importance", col = "green")
Warning messages:
1: In mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color,  :
  "mean" is not a graphical parameter
2: In title(main = main, xlab = xlab, ylab = ylab, ...) :
  "mean" is not a graphical parameter
> varImpPlot(rf)
>
```

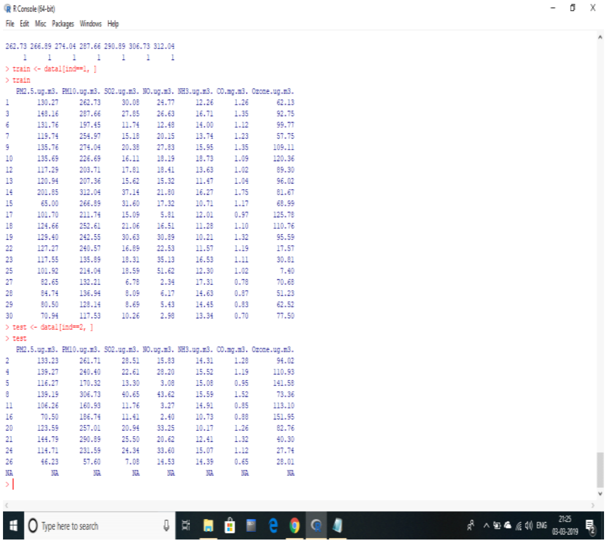
Fig.9: Representing the Variable Importance,2018 Jan. data



```
  "mean" is not a graphical parameter
2: In title(main = main, xlab = xlab, ylab = ylab, ...) :
  "mean" is not a graphical parameter
> varImpPlot(rf, mean = " Top prefer Variable Importance", col = "green")
Warning messages:
1: In mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color,  :
  "mean" is not a graphical parameter
2: In title(main = main, xlab = xlab, ylab = ylab, ...) :
  "mean" is not a graphical parameter
> varImpPlot(rf)
> varImpPlot(rf1)
> importance(rf)
              IncNodePurity
PM2.5..ug.m3.     10887.026
SO2..ug.m3.        2959.599
NO..ug.m3.         4702.590
NH3..ug.m3.        6337.586
CO..mg.m3.        13974.984
Ozone..ug.m3.      7474.241
> varUsed(rf)
[1] 547 337 388 443 628 471
> #partialplot of the data
> partialPlot(rf, train, variable, "1")
Error in `[.data.frame`(pred.data, , xname) : undefined columns selected
> getTree(rf, 1, labelVar = TRUE)
   left daughter right daughter     split var split point status prediction
1              2              3 Ozone..ug.m3.      74.990     -3   193.0660
2              4              5 Ozone..ug.m3.      39.400     -3   212.5460
3              0              0          <NA>       0.000     -1   134.6260
4              0              0          <NA>       0.000     -1   290.6500
5              6              7    CO..mg.m3.       1.470     -3   200.5300
6              8              9   NH3..ug.m3.      14.260     -3   194.1356
7              0              0          <NA>       0.000     -1   214.9175
8              0              0          <NA>       0.000     -1   167.9000
9             10             11    NO..ug.m3.      33.865     -3   197.4150
10            12             13 Ozone..ug.m3.      59.215     -3   198.2871
11             0              0          <NA>       0.000     -1   191.3100
12             0              0          <NA>       0.000     -1   201.0700
13             0              0          <NA>       0.000     -1   196.2000
>
```

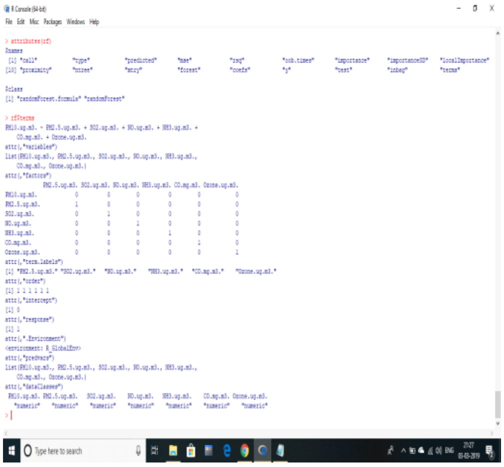
Fig.10: Showing the tree formation for training data, Jan-Dec2018

January 2019 data



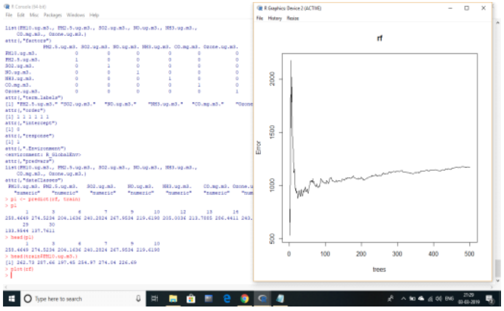
```
262.73 266.89 274.04 287.66 290.89 306.73 312.04
     1      1      1      1      1      1      1
> train <- data1[ind==1, ]
> train
   PM2.5.ug.m3. PM10.ug.m3. SO2.ug.m3. NO.ug.m3. NH3.ug.m3. CO.mg.m3. Ozone.ug.m3.
1        130.27      262.73      30.08     24.77      12.26      1.26        62.13
3        148.16      287.66      27.85     26.63      16.71      1.35        92.75
6        131.76      197.45      11.74     12.48      14.00      1.12        99.77
7        119.74      254.97      15.18     20.15      13.74      1.23        57.75
9        135.76      274.04      20.38     27.83      15.95      1.35       109.11
10       135.69      226.69      16.11     18.19      18.73      1.09       120.36
12       117.29      203.71      17.81     18.41      13.63      1.02        89.30
13       120.94      207.36      15.62     15.32      11.47      1.04        96.02
14       201.85      312.04      37.14     21.80      16.27      1.75        81.67
15        65.00      266.89      31.60     17.32      10.71      1.17        68.99
17       101.70      211.74      15.09      5.81      12.01      0.97       125.78
18       124.66      252.61      21.06     16.51      11.28      1.10       110.76
19       129.40      242.55      30.63     30.89      10.21      1.32        95.59
22       127.27      240.57      16.89     22.53      11.57      1.19        17.57
23       117.55      135.89      18.31     35.13      16.53      1.11        30.81
25       101.92      214.04      18.59     51.62      12.30      1.02         7.40
27        82.65      132.21       6.78      2.34      17.31      0.78        70.68
28        84.74      136.94       8.09      6.17      14.63      0.87        51.23
29        80.50      128.14       8.69      5.43      14.45      0.83        62.52
30        70.94      117.53      10.26      2.98      13.34      0.70        77.50
> test <- data1[ind==2, ]
> test
   PM2.5.ug.m3. PM10.ug.m3. SO2.ug.m3. NO.ug.m3. NH3.ug.m3. CO.mg.m3. Ozone.ug.m3.
2        133.23      261.71      28.51     15.83      14.31      1.28        94.02
4        139.27      240.40      22.61     28.20      15.52      1.19       110.93
5        116.27      170.32      13.30      3.08      15.08      0.95       141.58
8        139.19      306.73      40.65     43.62      15.59      1.52        73.36
11       106.26      160.93      11.76      3.27      14.91      0.85       113.10
16        70.50      186.74      11.41      2.40      10.73      0.88       151.95
20       123.59      257.01      20.94     33.25      10.17      1.26        82.76
21       144.79      290.89      25.50     20.62      12.41      1.32        40.30
24       114.71      231.59      24.34     33.60      15.07      1.12        27.74
26        46.23       57.60       7.08     14.53      14.39      0.65        28.01
NA           NA          NA         NA        NA         NA        NA           NA
>
```

Fig.11: Data representation for “jan-2019 data”



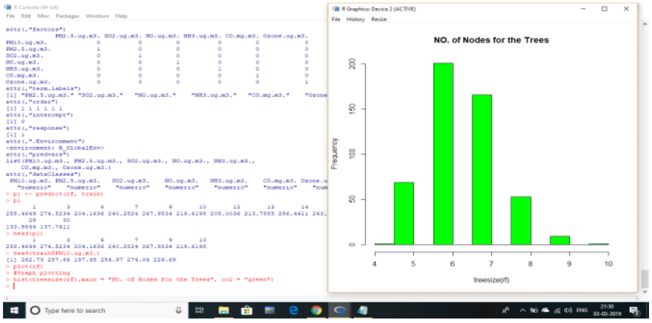
```
> attributes(rf)
$names
 [1] "call"            "type"            "predicted"       "mse"             "rsq"             "oob.times"       "importance"      "importanceSD"    "localImportance"
[10] "proximity"       "ntree"           "mtry"            "forest"          "coefs"           "y"               "test"            "inbag"           "terms"

$class
[1] "randomForest.formula" "randomForest"

> rf$terms
PM10.ug.m3. ~ PM2.5.ug.m3. + SO2.ug.m3. + NO.ug.m3. + NH3.ug.m3. +
    CO.mg.m3. + Ozone.ug.m3.
attr(,"variables")
list(PM10.ug.m3., PM2.5.ug.m3., SO2.ug.m3., NO.ug.m3., NH3.ug.m3.,
    CO.mg.m3., Ozone.ug.m3.)
attr(,"factors")
             PM2.5.ug.m3. SO2.ug.m3. NO.ug.m3. NH3.ug.m3. CO.mg.m3. Ozone.ug.m3.
PM10.ug.m3.             0          0         0          0         0            0
PM2.5.ug.m3.            1          0         0          0         0            0
SO2.ug.m3.              0          1         0          0         0            0
NO.ug.m3.               0          0         1          0         0            0
NH3.ug.m3.              0          0         0          1         0            0
CO.mg.m3.               0          0         0          0         1            0
Ozone.ug.m3.            0          0         0          0         0            1
attr(,"term.labels")
[1] "PM2.5.ug.m3." "SO2.ug.m3."   "NO.ug.m3."    "NH3.ug.m3."   "CO.mg.m3."    "Ozone.ug.m3."
attr(,"order")
[1] 1 1 1 1 1 1
attr(,"intercept")
[1] 0
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(PM10.ug.m3., PM2.5.ug.m3., SO2.ug.m3., NO.ug.m3., NH3.ug.m3.,
    CO.mg.m3., Ozone.ug.m3.)
attr(,"dataClasses")
 PM10.ug.m3. PM2.5.ug.m3.   SO2.ug.m3.    NO.ug.m3.   NH3.ug.m3.    CO.mg.m3. Ozone.ug.m3.
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
>
```

Fig.12: Identifying the Attributes for the data



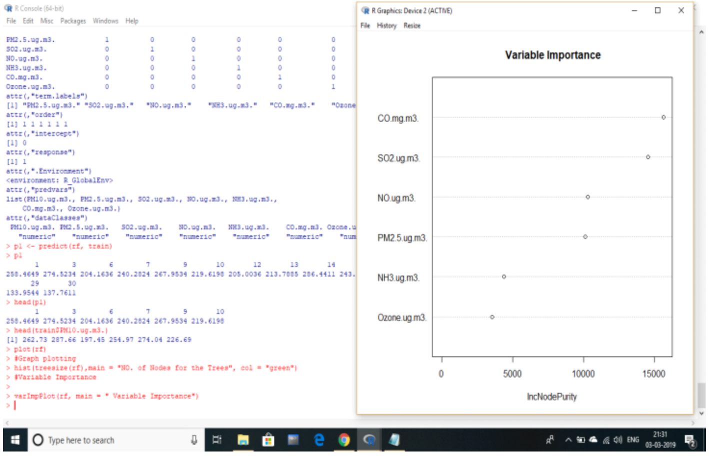
```
list(PM10.ug.m3., PM2.5.ug.m3., SO2.ug.m3., NO.ug.m3., NH3.ug.m3.,
    CO.mg.m3., Ozone.ug.m3.)
attr(,"factors")
             PM2.5.ug.m3. SO2.ug.m3. NO.ug.m3. NH3.ug.m3. CO.mg.m3. Ozone.ug.m3.
PM10.ug.m3.             0          0         0          0         0            0
PM2.5.ug.m3.            1          0         0          0         0            0
SO2.ug.m3.              0          1         0          0         0            0
NO.ug.m3.               0          0         1          0         0            0
NH3.ug.m3.              0          0         0          1         0            0
CO.mg.m3.               0          0         0          0         1            0
Ozone.ug.m3.            0          0         0          0         0            1
attr(,"term.labels")
[1] "PM2.5.ug.m3." "SO2.ug.m3."   "NO.ug.m3."    "NH3.ug.m3."   "CO.mg.m3."    "Ozone
attr(,"order")
[1] 1 1 1 1 1 1
attr(,"intercept")
[1] 0
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(PM10.ug.m3., PM2.5.ug.m3., SO2.ug.m3., NO.ug.m3., NH3.ug.m3.,
    CO.mg.m3., Ozone.ug.m3.)
attr(,"dataClasses")
 PM10.ug.m3. PM2.5.ug.m3.   SO2.ug.m3.    NO.ug.m3.   NH3.ug.m3.    CO.mg.m3. Ozone.u
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "num
> p1 <- predict(rf, train)
> p1
       1        3        6        7        9       10       12       13       14
258.4649 274.5234 204.1636 240.2824 267.9534 219.6190 205.0036 213.7885 206.4411 243.
      29       30
133.9544 137.7611
> head(p1)
       1        3        6        7        9       10
258.4649 274.5234 204.1636 240.2824 267.9534 219.6190
> head(train$PM10.ug.m3.)
[1] 262.73 287.66 197.45 254.97 274.04 226.69
> plot(rf)
>
```

Fig.13: Graph plotting for tree data set ,2019 data set



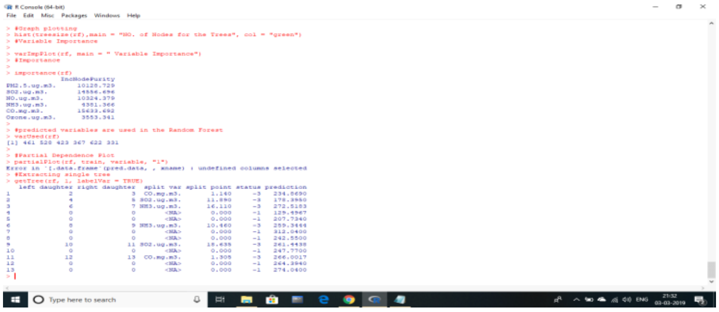
```
attr(,"factors")
              PM2.5.ug.m3. SO2.ug.m3. NO.ug.m3. NH3.ug.m3. CO.mg.m3. Ozone.ug.m3.
PM10.ug.m3.              0          0         0          0         0            0
PM2.5.ug.m3.             1          0         0          0         0            0
SO2.ug.m3.               0          1         0          0         0            0
NO.ug.m3.                0          0         1          0         0            0
NH3.ug.m3.               0          0         0          1         0            0
CO.mg.m3.                0          0         0          0         1            0
Ozone.ug.m3.             0          0         0          0         0            1
attr(,"term.labels")
[1] "PM2.5.ug.m3." "SO2.ug.m3."   "NO.ug.m3."    "NH3.ug.m3."   "CO.mg.m3."    "Ozone
attr(,"order")
[1] 1 1 1 1 1 1
attr(,"intercept")
[1] 0
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(PM10.ug.m3., PM2.5.ug.m3., SO2.ug.m3., NO.ug.m3., NH3.ug.m3.,
    CO.mg.m3., Ozone.ug.m3.)
attr(,"dataClasses")
 PM10.ug.m3. PM2.5.ug.m3.   SO2.ug.m3.    NO.ug.m3.   NH3.ug.m3.    CO.mg.m3. Ozone.u
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "num
> pl <- predict(rf, train)
> pl
       1        3        6        7        9       10       12       13       14
258.4649 274.5234 204.1636 240.2824 267.9534 219.6198 205.0036 213.7885 286.4411 243.
      29       30
133.9544 137.7611
> head(pl)
       1        3        6        7        9       10
258.4649 274.5234 204.1636 240.2824 267.9534 219.6198
> head(train$PM10.ug.m3.)
[1] 262.73 287.66 197.45 254.97 274.04 226.69
> plot(rf)
> #Graph plotting
> hist(treesize(rf),main = "NO. of Nodes for the Trees", col = "green")
> |
```

Fig.14: Plotting the Histogram graph for the training data using RF Algorithm, Jan-2019



```
PM2.5.ug.m3.             1          0         0          0         0            0
SO2.ug.m3.               0          1         0          0         0            0
NO.ug.m3.                0          0         1          0         0            0
NH3.ug.m3.               0          0         0          1         0            0
CO.mg.m3.                0          0         0          0         1            0
Ozone.ug.m3.             0          0         0          0         0            1
attr(,"term.labels")
[1] "PM2.5.ug.m3." "SO2.ug.m3."   "NO.ug.m3."    "NH3.ug.m3."   "CO.mg.m3."    "Ozone
attr(,"order")
[1] 1 1 1 1 1 1
attr(,"intercept")
[1] 0
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(PM10.ug.m3., PM2.5.ug.m3., SO2.ug.m3., NO.ug.m3., NH3.ug.m3.,
    CO.mg.m3., Ozone.ug.m3.)
attr(,"dataClasses")
 PM10.ug.m3. PM2.5.ug.m3.   SO2.ug.m3.    NO.ug.m3.   NH3.ug.m3.    CO.mg.m3. Ozone.u
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "num
> pl <- predict(rf, train)
> pl
       1        3        6        7        9       10       12       13       14
258.4649 274.5234 204.1636 240.2824 267.9534 219.6198 205.0036 213.7885 286.4411 243.
      29       30
133.9544 137.7611
> head(pl)
       1        3        6        7        9       10
258.4649 274.5234 204.1636 240.2824 267.9534 219.6198
> head(train$PM10.ug.m3.)
[1] 262.73 287.66 197.45 254.97 274.04 226.69
> plot(rf)
> #Graph plotting
> hist(treesize(rf),main = "NO. of Nodes for the Trees", col = "green")
> #Variable Importance
>
> varImpPlot(rf, main = " Variable Importance")
> |
```

Fig.15: Representing the Variable Importance,2019- Jan data



```
> #Graph plotting
> hist(treesize(rf),main = "NO. of Nodes for the Trees", col = "green")
> #Variable Importance
>
> varImpPlot(rf, main = " Variable Importance")
> #Importance
>
> importance(rf)
             IncNodePurity
PM2.5.ug.m3.     10128.729
SO2.ug.m3.       14556.696
NO.ug.m3.        10324.379
NH3.ug.m3.        4381.366
CO.mg.m3.        15633.692
Ozone.ug.m3.      3553.341
>
> #predicted variables are used in the Random Forest
> varUsed(rf)
[1] 461 520 423 367 622 331
>
> #Partial Dependence Plot
> partialPlot(rf, train, variable, "1")
Error in `[.data.frame`(pred.data, , xname) : undefined columns selected
> #Extracting single tree
> getTree(rf, 1, labelVar = TRUE)
   left daughter right daughter  split var split point status prediction
1              2              3  CO.mg.m3.       1.140     -3   234.8690
2              4              5 SO2.ug.m3.      11.890     -3   178.3950
3              6              7 NH3.ug.m3.      16.110     -3   272.5183
4              0              0       <NA>       0.000     -1   129.4967
5              0              0       <NA>       0.000     -1   207.7340
6              8              9 NH3.ug.m3.      10.460     -3   259.3444
7              0              0       <NA>       0.000     -1   312.0400
8              0              0       <NA>       0.000     -1   242.5500
9             10             11 SO2.ug.m3.      18.638     -3   261.4438
10             0              0       <NA>       0.000     -1   247.7700
11            12             13  CO.mg.m3.       1.305     -3   266.0017
12             0              0       <NA>       0.000     -1   264.3940
13             0              0       <NA>       0.000     -1   274.0400
> |
```

Fig .16: Showing the tree formation for training data, Jan-2019

Table 1: Breakpoints table for the AQI

TABLE 2—BREAKPOINTS FOR THE AQI

These breakpoints							Equal these AQI's	
O_3(ppm) 8-hour	O_3(ppm) 1-hour[1]	$PM_{2.5}$($\mu g/m^3$) 24-hour	PM_{10}($\mu g/m^3$) 24-hour	CO (ppm) 8-hour	SO_2(ppb) 1-hour	NO_2(ppb) 1-hour	AQI	Category
0.000-0.059		0.0-12.0	0-54	0.0-4.4	0-35	0-53	0-50	Good.
0.060-0.075		12.1-35.4	55-154	4.5-9.4	36-75	54-100	51-100	Moderate.
0.076-0.095	0.125-0.164	35.5-55.4	155-254	9.5-12.4	76-185	101-360	101-150	Unhealthy for Sensitive Groups.
0.096-0.115	0.165-0.204	[3] 55.5-150.4	255-354	12.5-15.4	[4] 186-304	361-649	151-200	Unhealthy.
0.116-0.374	0.205-0.404	[3] 150.5-250.4	355-424	15.5-30.4	[4] 305-604	650-1249	201-300	Very Unhealthy.
(2)	0.405-0.504	[3] 250.5-350.4	425-504	30.5-40.4	[4] 605-804	1250-1649	301-400	Hazardous.
(2)	0.505-0.604	[3] 350.5-500.4	505-604	40.5-50.4	[4] 805-1004	1650-2049	401-500	

The overall result will be showing the pollutants in a graph with month intervals and identifying the causes.

While choosing the breakpoint table there are three cases to be followed.

1. we consider areas large in number and the AQI values based on the 8 hrs ozone values. However, for smaller number of areas AQI based on 1 hrs ozone value, that would be more precautionary. Inthese cases, both 8 hrs and 1 hrs index value may be calculated and maximum two values are reported.

2 .NO2 has no short-term NAAQS (National Ambient Air Quality Standards) and can generate an AQI only above an AQI value of 200.

3. When 8-hour O3 concentrations exceed 0.374ppm, AQI values of 301 or higher must be calculated with 1-hour O3 concentrations.

IV. CONCLUSION

Air quality is an important aspect on the earth, because it is linked to our healthy survival. Everyday air is getting polluted heavily and it leads to health issues hence in an attempt to control it, a proper monitoring and analysis can be done by considering every two different years data set obtained from the Central Pollution Control Board. Making use of suchdata,and by applying Random Forest Algorithm in R language, error rate has been identified and the result has been observed as provided in the above figures. As the quality of air is measurable its study and analysis can significantly impact public health.

V. REFERENCES

[1]. Ranjana Waman Gore, Deepa S. Deshpande, “An Approach for Classification of Health Risks Based on Air Quality Levels”, International Conference on Intelligent Systems and Information Management (ICISIM), 2017, IEEE.

[2]. Random Forest Algorithm for the Relationship between Negative Air Ions and Environmental Factors in an Urban Park ,2018, 9(12), 463.

[3]. Costa, D., Dreher, K. 1997. Bioavailable Transition Metals in Particulate Matter http://home.ijasca.com/data/documents/ID-45_Pg140-153_Air-Quality-Analysis-Based-On-MapReduce.pdf

[4]. Mediate Cardiopulmonary Injury in Healthy and Compromised Animal Models. Environ. Health Perspectives 105, Suppl. 5: 97.

[5]. *(PDF) Using Factor Analysis to Attribute Health Impacts to Particulate Pollution Sources*. Available from https://www.researchgate.net/publication/8502837_Using_Factor_Analysis_to_Attribute_Health_Impacts_to_Particulate_Pollution_Sources [accessed Jan 03 2019].

[6]. Y.-L. Zhang, F. Cao, Environmental Pollution, 2015, 202, 217–219.

[7]. World Health Organization, “Air Quality Guidelines: Global Update 2005: Particulate Matter, Ozone, Nitrogen Dioxide, and Sulphur Dioxide”, 2006

[8]. Attri, A. K., Kumar, U. and Jain, V.K. (2001). Formation of Ozone by Fireworks. Nature.

[9]. Cogliani, E. (2001). Air Pollution Forecast in Cities by an Air Pollution Index Highly Correlated with the Meteorological Variables. Atmos. Environ. 35: 2871-2877.

[10]. IJCSN - International Journal of Computer Science and Network,Volume7,Issue1,January-2018"Air Data Analysis for Predicting Health Risks " http://ijcsn.org/IJCSN-2018/7-1/Air-Data-Analysis-for-Predicting-Health-Risks.pdf

[11]. Building a Random Forest from Scratch & Understanding Real-World Data Products (ML for Programmers),2016, Vol. 26, No. 2, pp. 467–478.