AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

DEPARTMENT OF CSE

Data Warehousing and Data Mining

Summer 2022-2023

Section: A

Final Term Project On

Creating a KNN model from scratch
and implementing it on a dataset all using R language

Based On

Global Air Pollution Dataset

Supervised By:

Dr. Akinul Islam Jony

Associate Professor & Head (UG), Computer Science

Submitted By:

Name	ID
1. Muhammad Shahriar Zaman	20-41840-1
2. Ashraful Islam	20-42010-1

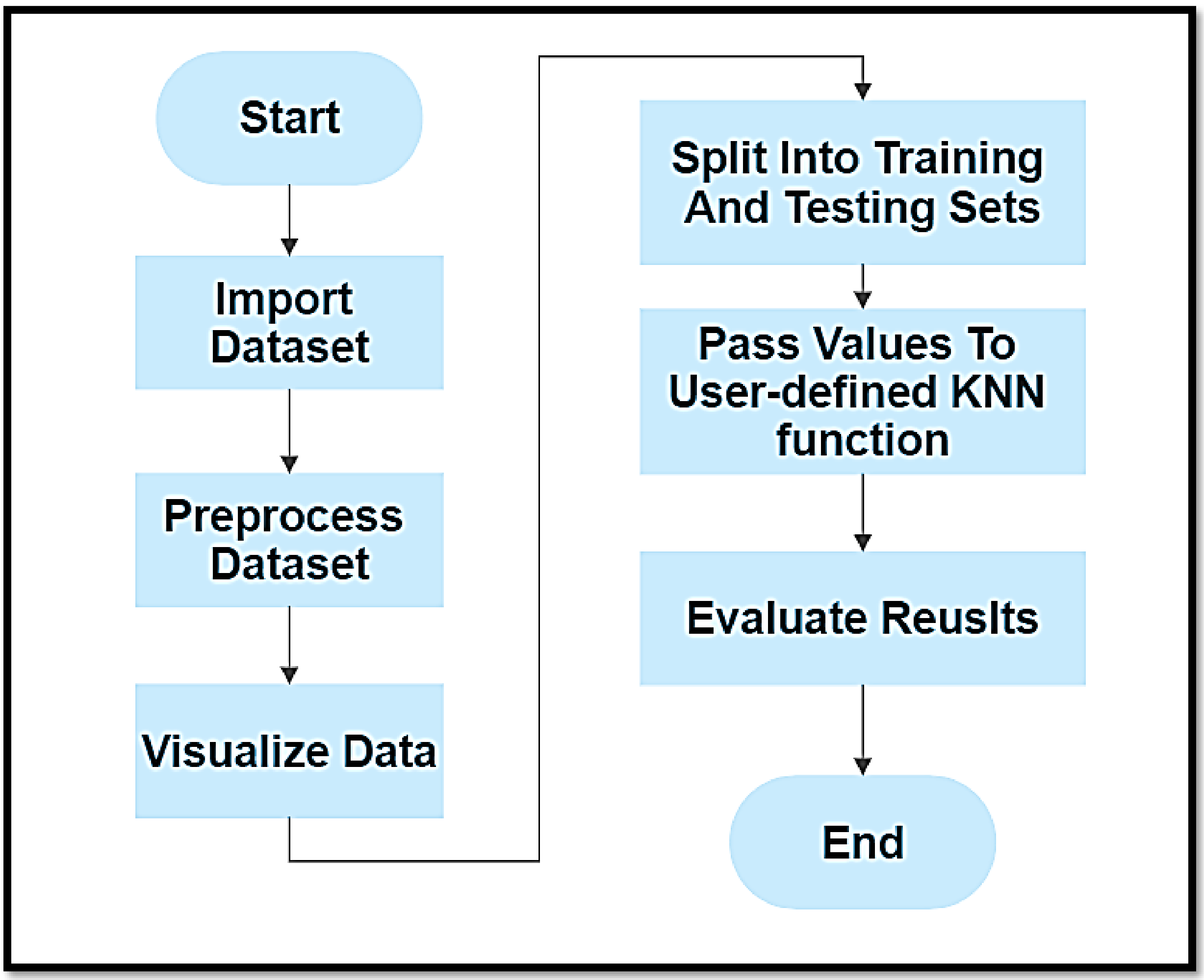
Date of Submission: **June 4th, 2023**

TABLE OF CONTENTS

<u>Project Overview:</u>

Air pollution is a major issue in the metropolitan life of modern times. Man-made causes are damaging air purity and in-tern harming the balance of our ecosystem. Effects of these can be noticed from the multitude of ailments caused by air pollution such as aggravated asthma, chronic obstructive pulmonary disease, lung cancer and many more.

Identifying the problem will be the first step in terms of solving it. This project will aim to create a machine learning model which can accurately predict air quality based on provided parameters.



<u>Fig. 1:</u> Course of actions of this project

<u>Dataset Overview:</u> Our dataset originally has 12 attributes and 23,463 records.

Dataset source->
https://www.kaggle.com/datasets/hasibalmuzdadid/global-air-pollution-dataset?fbclid=IwAR0gVSQlEEJG8zWPPSmnJyoKkjtw90Q0uxBz2oWOwDp2_hkw4QWh5Rl95PQ

1	Country	City	AQI_Value	AQI_Category	CO_Value	CO_Category	Ozone_Value	Ozone_Category	NO2_Value	NO2_Category	PM_2.5_Value	PM2.5 AQI Category
2	Russian Fe	Praskoveya	51	Moderate	1	Good	36	Good	0	Good	51	Moderate
3	Brazil	Presidente Dutra	41	Good	1	Good	5	Good	1	Good	41	Good
4	Italy	Priolo Gargallo	66	Moderate	1	Good	39	Good	2	Good	66	Moderate
5	Poland	Przasnysz	34	Good	1	Good	34	Good	0	Good	20	Good
6	France	Punaauia	22	Good	0	Good	22	Good	0	Good	6	Good
7	United Sta	Punta Gorda	54	Moderate	1	Good	14	Good	11	Good	54	Moderate
8	Germany	Puttlingen	62	Moderate	1	Good	35	Good	3	Good	62	Moderate
9	Belgium	Puurs	64	Moderate	1	Good	29	Good	7	Good	64	Moderate
10	Russian Fe	Pyatigorsk	54	Moderate	1	Good	41	Good	1	Good	54	Moderate
11	Egypt	Qalyub	142	Unhealthy for Se	3	Good	89	Moderate	9	Good	142	Unhealthy for Sensitive Groups
12	China	Qinzhou	68	Moderate	2	Good	68	Moderate	1	Good	58	Moderate
13	Netherlan	Raalte	41	Good	1	Good	24	Good	6	Good	41	Good
14	India	Radaur	158	Unhealthy	3	Good	139	Unhealthy for Se	1	Good	158	Unhealthy
15	Pakistan	Radhan	158	Unhealthy	1	Good	50	Good	1	Good	158	Unhealthy
16	Republic o	Radovis	83	Moderate	1	Good	46	Good	0	Good	83	Moderate
17	France	Raismes	59	Moderate	1	Good	30	Good	4	Good	59	Moderate
18	India	Rajgir	154	Unhealthy	3	Good	100	Unhealthy for Se	2	Good	154	Unhealthy
19	Italy	Ramacca	55	Moderate	1	Good	47	Good	0	Good	55	Moderate
20	United Sta	Phoenix	72	Moderate	1	Good	4	Good	23	Good	72	Moderate

<u>Fig. 2:</u> Dataset to be used in this project (Global Air pollution)

(P.T.O)

Dataset Overview (Cont'd):

We have described each of the columns of our dataset below:

- **Country:** Name of the country from which the air will be studied
- **City:** Name of the cities(unique); represents each row.
- **AQI Value:** Overall AQI**(Air Quality Index)** value of the city, the lesser the better
- **AQI Category:** Overall AQI category of the city
- **CO Value:** AQI value of Carbon Monoxide of the city
- **CO Category:** AQI category of Carbon Monoxide**(Pollutant)** of the city
- **Ozone Value:** AQI value of Ozone**(Pollutant)** of the city
- **Ozone Category:** AQI category of Ozone of the city
- **NO2 Value:** AQI value of Nitrogen Dioxide**(NO2, works as a pollutant)** of the city
- **NO2 Category:** AQI category of Nitrogen Dioxide of the city
- **PM2.5 Value:** AQI value of Particulate Matter with a diameter of 2.5**(Type of pollutant)** micrometers or less of the city.
- **PM2.5 Category:** AQI category of Particulate Matter with a diameter of 2.5 micrometers or less of the city.

AQI or Air Quality Index is a frequently occurring phrase in this project. AQI is a scale of air pollution and lower values are always preferred. It can be calculated using the following equation:

$$\boldsymbol{AQI_{pollutant}} = \frac{\boldsymbol{reading_{pollutant}}}{\boldsymbol{standard\ value}} \times \mathbf{100} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\boldsymbol{i})$$

Importing our dataset:

Code:

```
rm(list = ls()) # Clearing all previous variables
dataset <-
read.csv("C:/Users/Asus/Desktop/Data Mining Project/global air pollution dataset.csv")
# Importing Dataset
View(dataset)
```

Result:

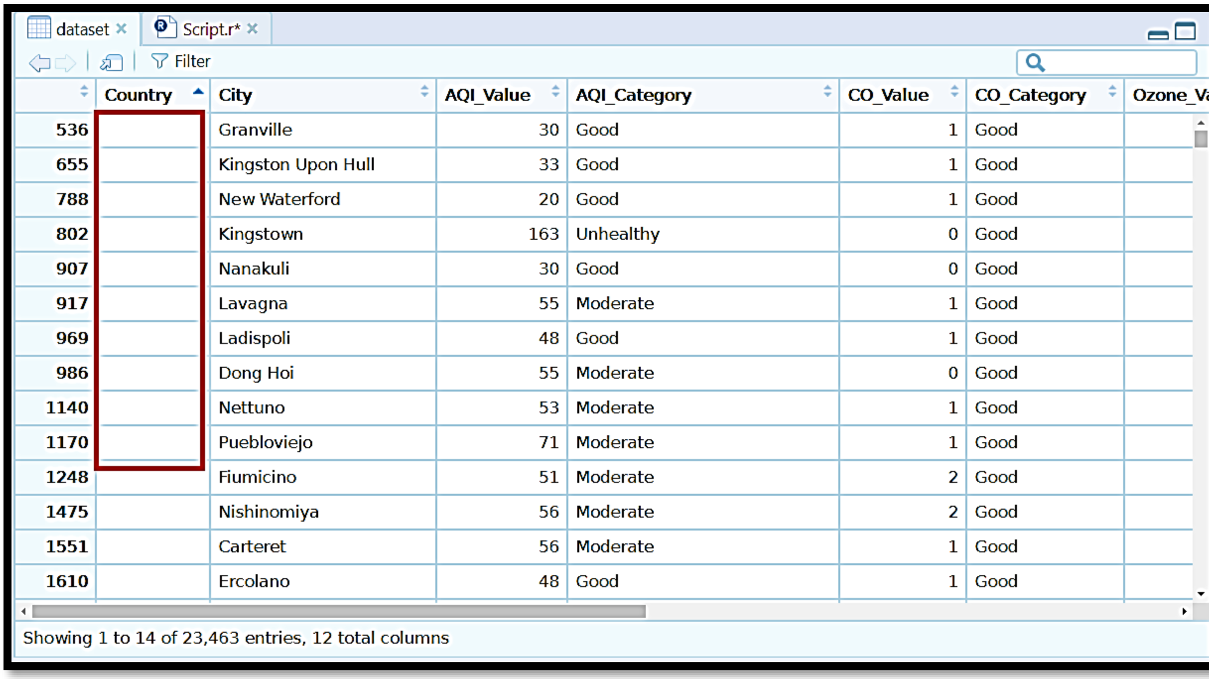
	Country	City	AQI_Value	AQI_Category	CO_Value
1	Russian Federation	Praskoveya	51	Moderate	
2	Brazil	Presidente Dutra	41	Good	
3	Italy	Priolo Gargallo	66	Moderate	
4	Poland	Przasnysz	34	Good	
5	France	Punaauia	22	Good	
6	United States of America	Punta Gorda	54	Moderate	
7	Germany	Puttlingen	62	Moderate	
8	Belgium	Puurs	64	Moderate	
9	Russian Federation	Pyatigorsk	54	Moderate	
10	Egypt	Qalyub	142	Unhealthy for Sensitive Groups	
11	China	Qinzhou	68	Moderate	
12	Netherlands	Raalte	41	Good	
13	India	Radaur	158	Unhealthy	
14	Pakistan	Radhan	158	Unhealthy	

Showing 1 to 14 of 23,463 entries, 12 total columns

Fig. 2: Dataset imported and stored in a data frame(all columns are not visible)

Dataset Preprocessing: For training our KNN machine learning model we have to work with a clean dataset, that is why we are preprocessing our data frame.

1. Handling Missing Values: Our data frame has some missing values in the country and city columns.



dataset × Script.r* ×

Filter

	Country	City	AQI_Value	AQI_Category	CO_Value	CO_Category	Ozone_Va
536		Granville	30	Good	1	Good	
655		Kingston Upon Hull	33	Good	1	Good	
788		New Waterford	20	Good	1	Good	
802		Kingstown	163	Unhealthy	0	Good	
907		Nanakuli	30	Good	0	Good	
917		Lavagna	55	Moderate	1	Good	
969		Ladispoli	48	Good	1	Good	
986		Dong Hoi	55	Moderate	0	Good	
1140		Nettuno	53	Moderate	1	Good	
1170		Puebloviejo	71	Moderate	1	Good	
1248		Fiumicino	51	Moderate	2	Good	
1475		Nishinomiya	56	Moderate	2	Good	
1551		Carteret	56	Moderate	1	Good	
1610		Ercolano	48	Good	1	Good	

Showing 1 to 14 of 23,463 entries, 12 total columns

Fig.3: Dataset with missing values in country column

Deleting Rows with Missing Values:

Code:

```
# Deleting rows with blank values in Country/City column

dataset <- dataset[dataset$Country!="", ]
dataset <- dataset[dataset$City!="", ]
View(dataset)
```

(P.T.O)

Output:

	Country	City	AQI_Value	AQI_Category	CO_Valu
534	United States of America	Granger	57	Moderate	
535	United States of America	Grants Pass	29	Good	
537	Mexico	Guasave	35	Good	
538	Germany	Guben	37	Good	
539	Cameroon	Guider	76	Moderate	
540	China	Guilin	90	Moderate	
541	France	Guipavas	34	Good	
542	United States of America	Fort Walton Beach	65	Moderate	
543	France	Fosses	29	Good	
544	United States of America	Fountain Hills	37	Good	
545	United States of America	Fountain	72	Moderate	
546	Brazil	Francisco Morato	169	Unhealthy	
547	Germany	Freiberg	36	Good	
548	United States of America	Front Royal	46	Good	

Showing 533 to 547 of 23,035 entries, 12 total columns

Fig.4: Data frame with missing values removed

2. Data Reduction: Here we have 12 total rows, but 6 of them are redundant for our project. The country and city name columns cannot be used to determine air quality through KNN.

The attributes CO_Category, Ozone_Category, NO2_Category and PM_2.5_Category are also obsolete because they are categorical and their values are already related to other continuous variables.

So we will delete these 6 columns. In addition to this our data frame has around 23,000 rows which is a lot compared to our project requirement, so we shall take a sample of 5,000 rows from this.

(P.T.O)

Code:

```
# Deleting ("Country", "City", "CO_Category", "Ozone_Category", "NO2_Category"
# and "PM_2.5_Category columns

dataset <- dataset[ , ! colnames(dataset) %in% c("Country", "City", "CO_Category",
"Ozone_Category", "NO2_Category", "PM_2.5_Category")]


# Taking 5000 random samples from all the records

dataset<- dataset[sample(nrow(dataset), size=5000), ]
```

Output:

	AQI_Value	AQI_Category	CO_Value	Ozone_Value	NO2_Value	PM_2.5_Value
22296	71	Moderate	1	71	0	43
20301	80	Moderate	1	33	1	80
117	29	Good	1	29	1	7
8048	31	Good	0	10	1	31
9533	26	Good	1	26	1	15
20911	68	Moderate	2	38	0	68
10492	45	Good	1	45	0	17
8144	52	Moderate	1	15	5	52
5389	38	Good	1	15	0	38
17408	74	Moderate	1	74	1	37
4399	55	Moderate	1	32	5	55
14005	172	Unhealthy	1	71	0	172
17872	54	Moderate	1	27	1	54
11831	38	Good	1	38	1	32
15880	130	Unhealthy for Sensitive Groups	1	23	2	130

Showing 1 to 15 of 5,000 entries, 6 total columns

Fig.5: Data frame after column removal and sampling

Dataset Visualization: Let us visualize the relation between various pollutants and air quality.

Code: (AQI vs Ozone Value)

```
library(ggplot2) #Plotting tools belong to this library
ggplot(data = dataset, mapping = aes(x = Ozone_Value, y = AQI_Value)) +
geom_point(color = "orange", alpha = .7, size = 2)+ #Specifying color, opacity and size

geom_smooth( method = "lm")+ #Specifying a linear model to be fitted
ggtitle("Plot of overall Air Quality Index vs Ozone Value in air") +
#Title/heading of the plot
xlab("Ozone Value") + #Label of x axis
ylab("AQI Value") #Label of y axis
```

Output:

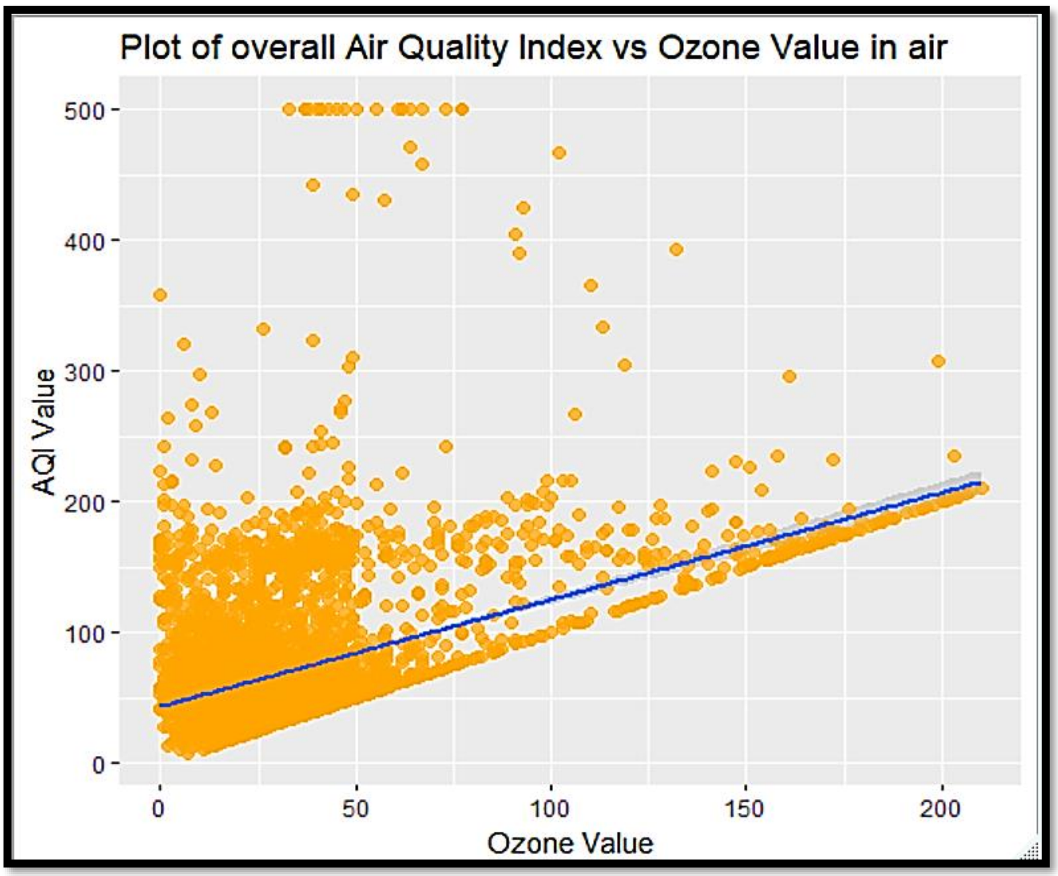


Fig.6: AQI vs Ozone value plot

Code: (AQI vs Carbon-Monixide Value)

```
library(ggplot2) #Plotting tools belong to this library
ggplot(data = dataset, mapping = aes(x = CO_Value, y = AQI_Value)) +
geom_point(color = "green", alpha = 1, size = 2)+ #Specifying color, opacity and size
geom_smooth(method = "lm")+ #Specifying a linear model to be fitted
ggtitle("Plot of overall Air Quality Index vs Carbon-Monoxide Value in air") +
#Title/heading of the plot
xlab("Carbon Monoxide Value") + #Label of x axis
ylab("AQI Value") #Label of y axis
```

Output:

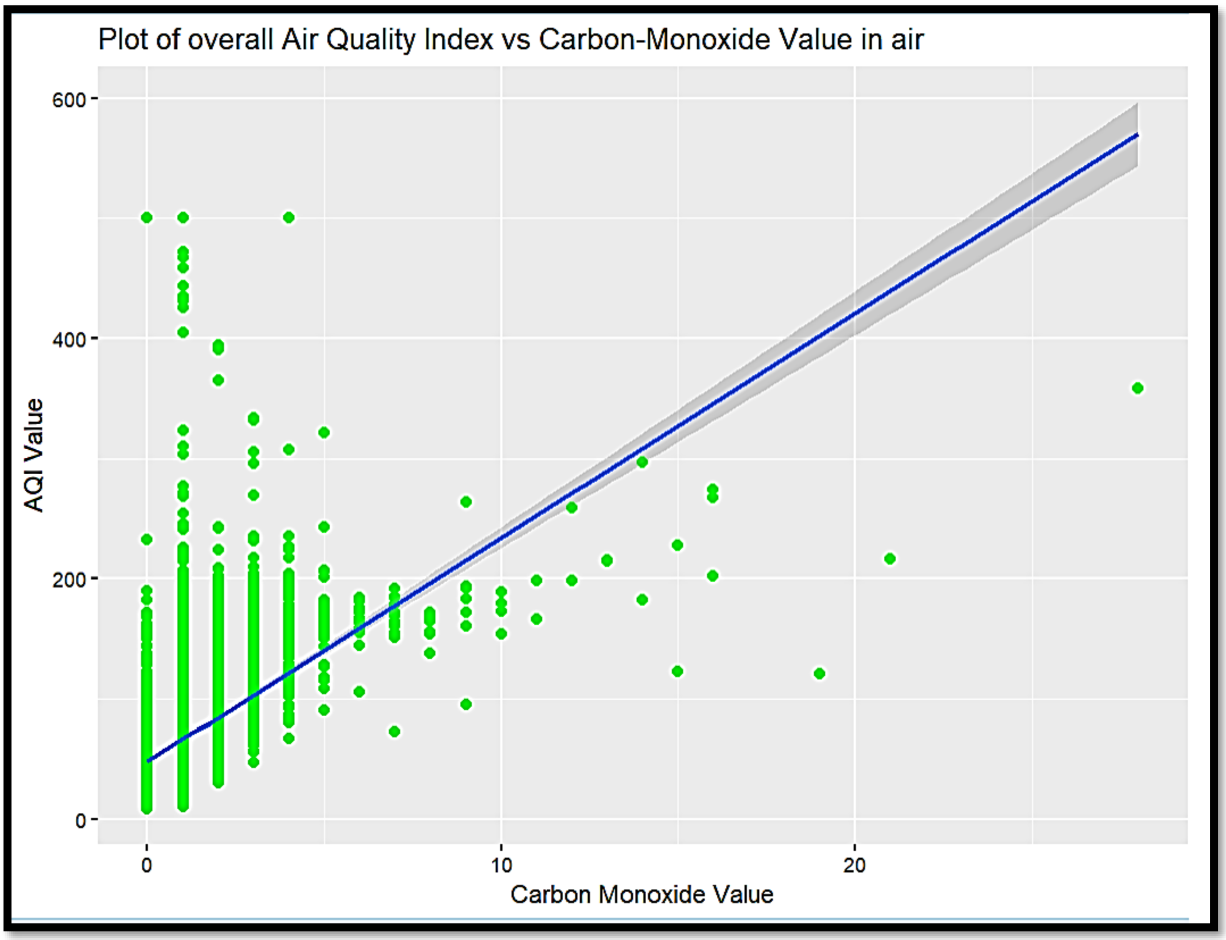


Fig.7: AQI vs Carbon Monoxide value plot

Code: (AQI vs Nitrous Oxide Value)

```
library(ggplot2) #Plotting tools belong to this library
ggplot(data = dataset, mapping = aes(x = NO2_Value, y = AQI_Value)) +
geom_point(color = "brown", alpha = 1, size = 2)+ #Specifying color, opacity and size
geom_smooth(method = "lm")+ #Specifying a linear model to be fitted
ggtitle("Plot of overall Air Quality Index vs Nitrous Oxide Value in air") +
#Title/heading of the plot
xlab("Nitrous Oxide Value") + #Label of x axis
ylab("AQI Value") #Label of y axis
```

Output:

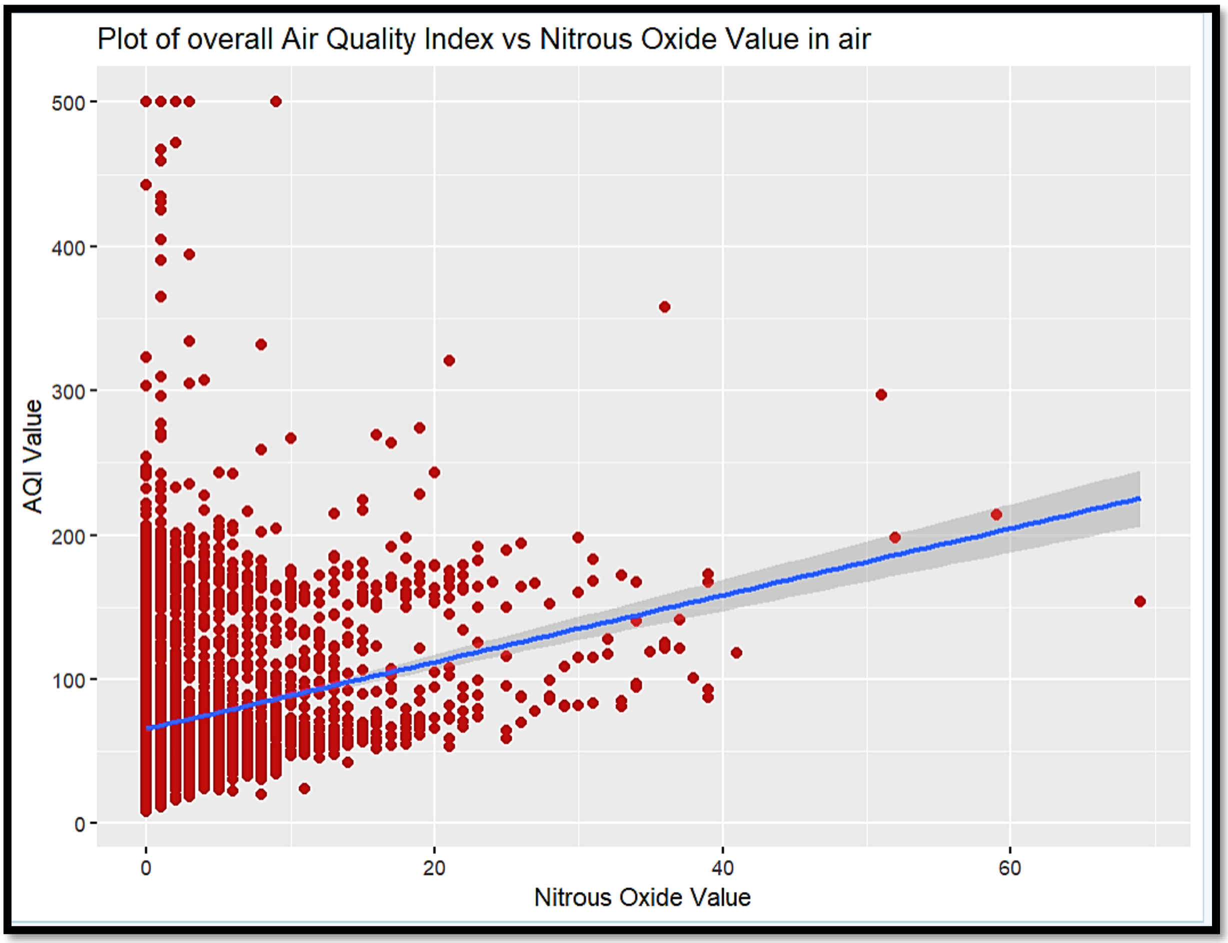


Fig.8: AQI vs Nitrous Oxide value plot

Code: (AQI vs PM 2.5 Value)

```
library(ggplot2) #Plotting tools belong to this library
ggplot(data = dataset, mapping = aes(x = PM_2.5_Value, y = AQI_Value)) +
geom_point(color = "black", alpha = 1, size = 2)+ #Specifying color, opacity and size
geom_smooth(method = "lm")+ #Specifying a linear model to be fitted
ggtitle("Plot of overall Air Quality Index vs value of 2.5 µm particulates in air") +
#Title/heading of the plot
xlab("2.5 µm particulates value") + #Label of x axis
ylab("AQI Value") #Label of y axis
```

Output:

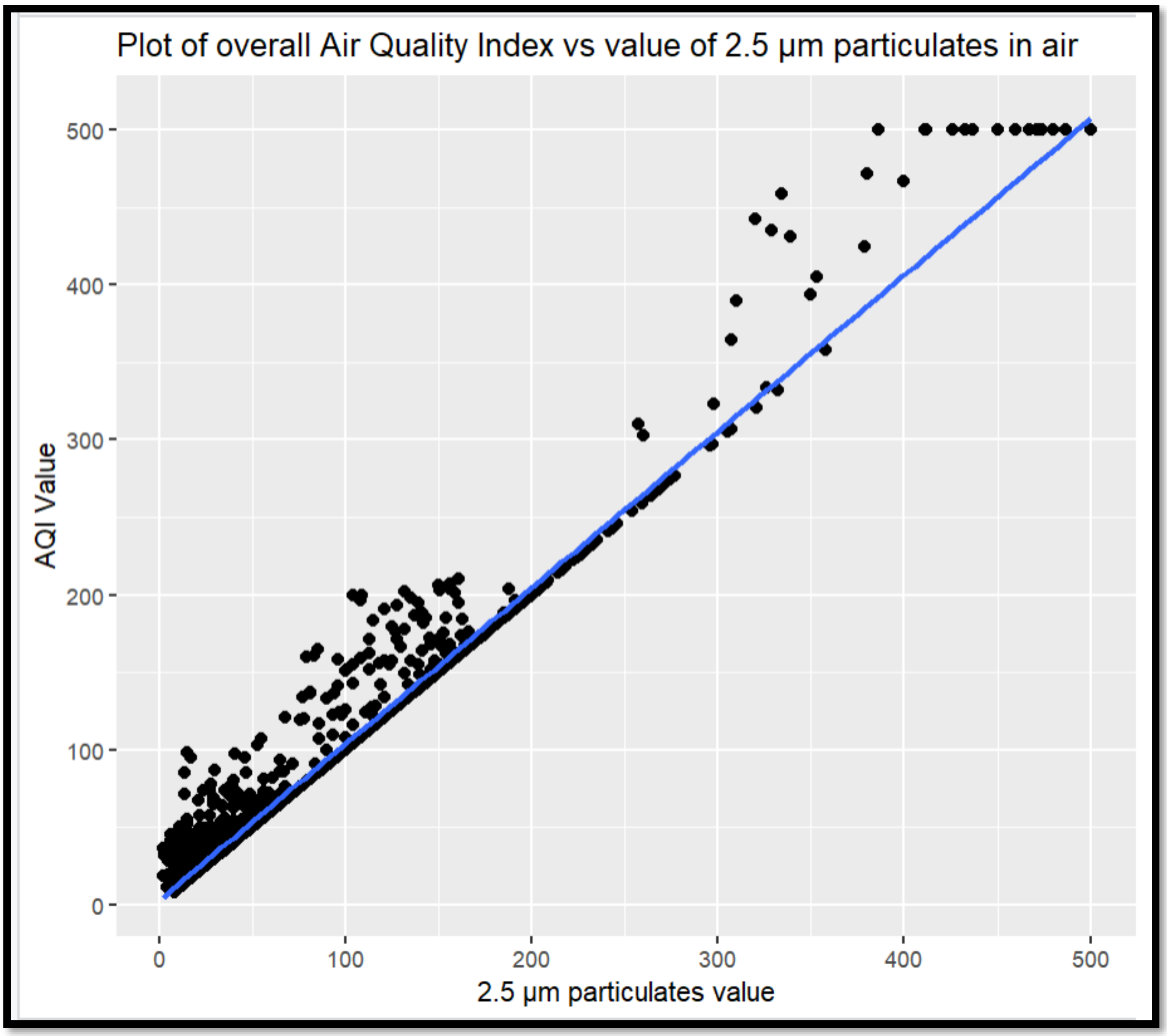


Fig.9: AQI vs 2.5 µm particulates value plot

Now, if we consider AQI_Category as target label then let us check wich class/category has most records. Here each record represents a city.

Code: (Barchart for each class of label)

```
# Factoring our label from 'Good' to 'Hazardous'
AQI_Category_list <-
factor(dataset$AQI_Category, levels=c('Good', 'Moderate', 'Unhealthy for Sensitive Groups',
'Unhealthy', 'Very Unhealthy', 'Hazardous'))

# Making a barplot

barplot(table(AQI_Category_list),
        main= "Number of cities divided by quality of air",
        xlab= "Grade",
        ylab= "Count",
        border= "red",
        col=c( "green" , "yellow" , "orange" , "red" , "brown" ,  "black" ))
```

Output:

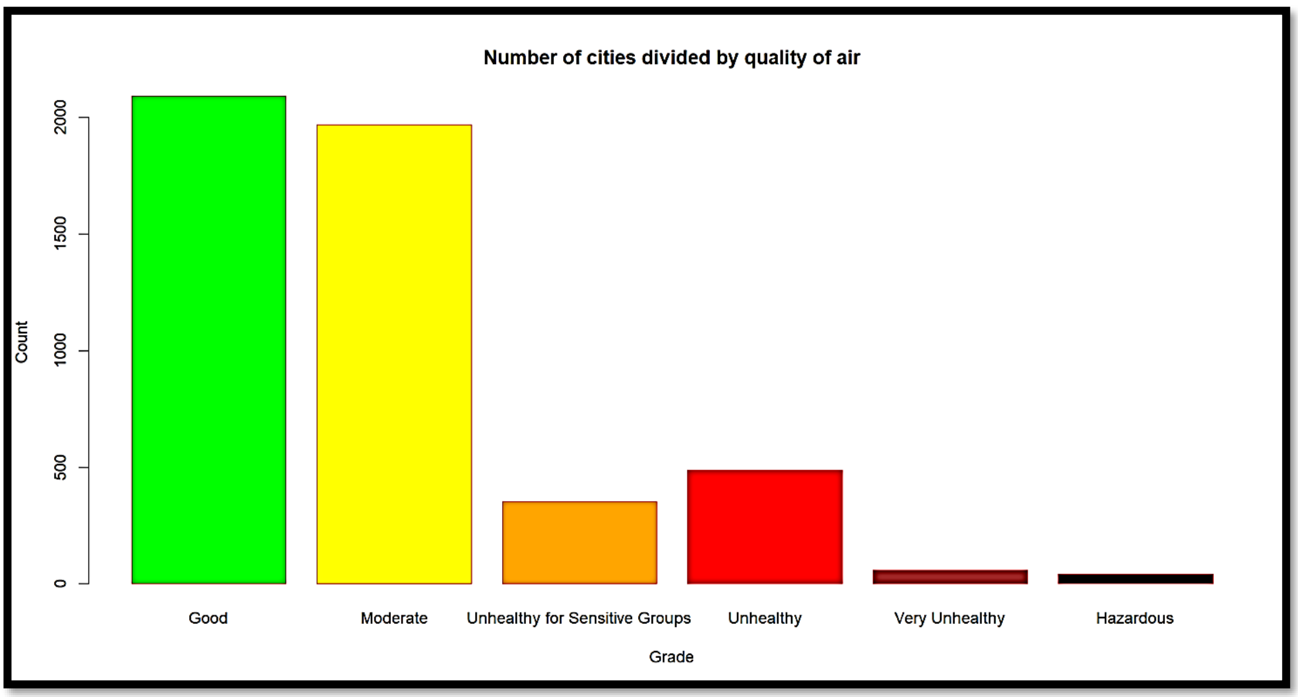


Fig.10: Quality or air among cities of the world

KNN Evaluation:

Our KNN function shall work in the following way:

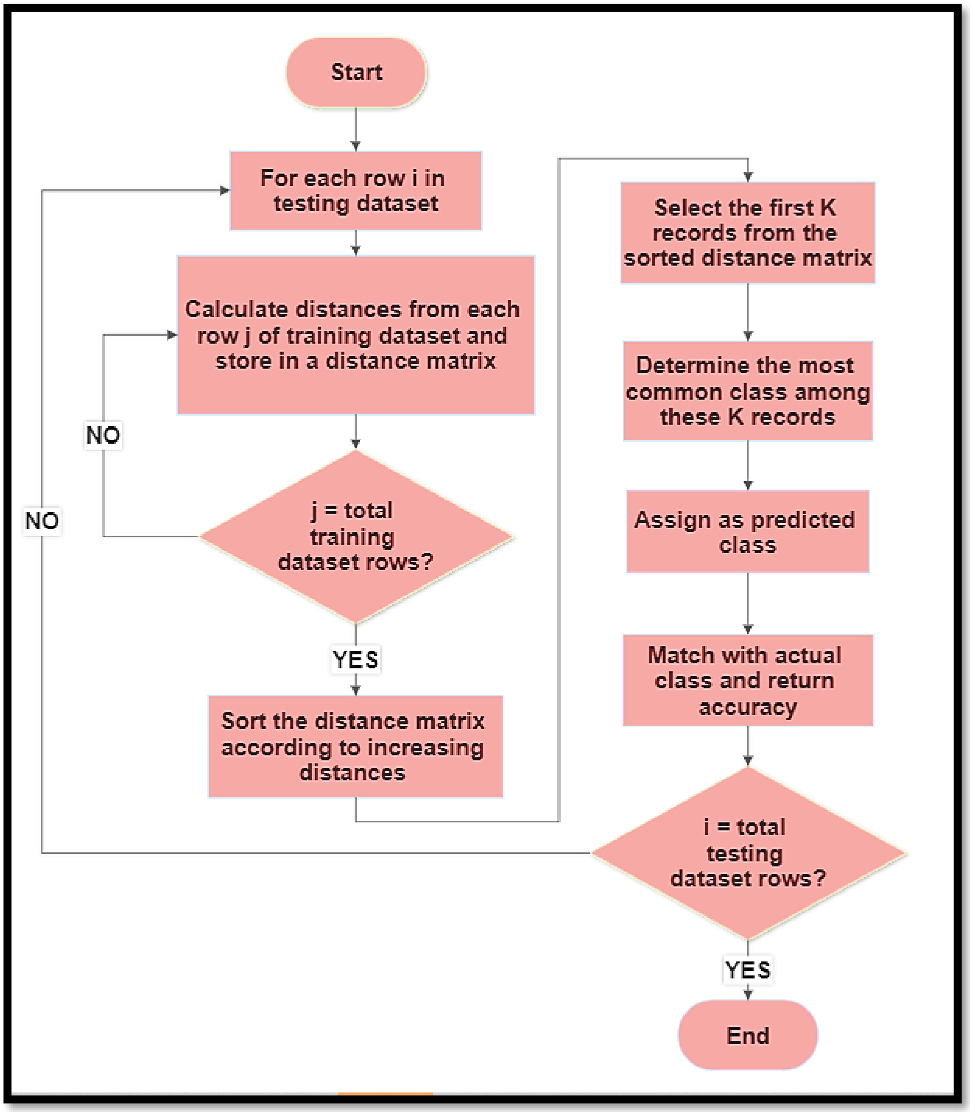


Fig.11: KNN algorithm flowchart for this project

Now, let us create some necessary functions.

Code:

```
# Splitter function
splitter = function(dataset, training_ratio)
{ # This function will split the dataset...
  # ...into training and testing sets and then return them in a list

  if(training_ratio>=1 | training_ratio<=0)
    {return("Training ratio has to be a fraction value i range: 0<x<1")

  }else
    {training_row_count<-round(training_ratio*nrow(dataset)) # No. Of rows
      testing_row_count<-round((1-training_ratio)*nrow(dataset))
      training_dataset<-head(dataset, training_row_count) # Splitting by head()
      testing_dataset<-tail(dataset, testing_row_count)  # Splitting by tail()
      return_vales <- list(training_dataset, testing_dataset) # Binding in a list
      return(return_vales)  # Returning the list
  }

  }
```

```
# Manhattan distance function
manhattan_distance=function(training_row, testing_row, label_name)
{ # This function will return manhattan distance
  distance=0

  for(attribute in colnames(testing_row))
  { if(attribute %in% c(label_name)){next}
   # Target label is irrelevant to calculating distance
  # Sometimes row count ‘X’ is also imported; which is also to be discarded
    distance=distance+ abs(testing_row[attribute] - training_row[attribute])
  }

  distance<-as.numeric(distance)
  return(distance)
}

# Euclidean distance function
euclidean_distance=function(training_row, testing_row, label_name)
{ # This function will return euclidean distance
  distance=0
  for(attribute in colnames(testing_row))
  { if(attribute %in% c(label_name)){next}
   # Target label is irrelevant to calculating distance
  # Sometimes row count ‘X’ is also imported; which is also to be discarded
    distance=distance+ (testing_row[attribute] - training_row[attribute])^2
  }
  distance<-as.numeric(distance)
  distance<-sqrt(distance)
  return(distance)}
```

```
# Function to work as KNN model
evaluate_KNN<-function(training_dataset, testing_dataset, k_value, label_name,
distance_type ){
 # This function will predict target label using KNN and show accuracy rates
 match_count=0
 accuracy=0
 for(x in 1:nrow(testing_dataset)){
  testing_row=testing_dataset[x,]
  #creating distance matrix with 0 rows and 3 columns
  distance_matrix <- data.frame(matrix(ncol = 3, nrow = 0))
  #providing column names
  colnames(distance_matrix) <- c('distance', 'predicted_label', 'actual_label')

  for(i in 1:nrow(training_dataset))
  { # Training
   training_row=training_dataset[i,]
   # Specifying type of distance
   if(distance_type=="Euclidean")
   {distance=euclidean_distance(training_row, testing_row, label_name)
   }
   else{ distance=manhattan_distance(training_row, testing_row, label_name)}

   predicted_label= training_row[label_name]

   actual_label = testing_row[label_name]

   distance_matrix[nrow(distance_matrix) + 1,] <- c(distance,predicted_label,actual_label)

  }
  ordered_distance_matrix <- distance_matrix[order(distance_matrix$distance),]
  distance_matrix<-ordered_distance_matrix
```

```
    distance_matrix<-head(distance_matrix, k_value) # Only considering k rows
    unique_values=(unique(distance_matrix$predicted_label))

    tabulated_values=tabulate(match(distance_matrix$predicted_label, unique_values))

    predicted_label=unique_values[tabulated_values==max(tabulated_values)]
    # Choosing the label which occurs the most among k records

    # Testing

    if(predicted_label==actual_label)
    {
      match_count=match_count+1

    }
}
  accuracy=round(match_count/nrow(testing_dataset), 2)*100 # Calculating Accuracy

  return(accuracy)}
```

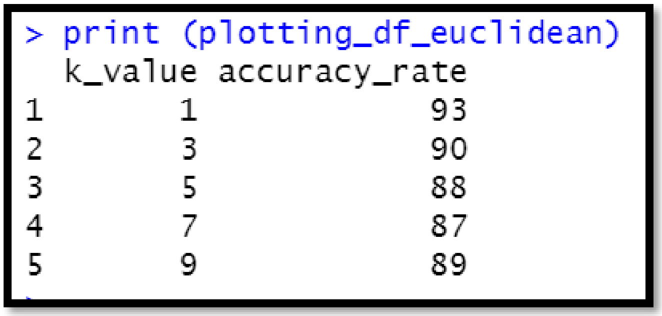
Now, let us evaluate the KNN model with our data

Code:

```
# Dropping AQI_Value column along with the X column, which is sometimes encountered
dataset= subset(dataset, select = -c(AQI_Value) )

# Making our dataset smaller in-case the model takes too long to execute, this part is optional
dataset<- dataset[sample(nrow(dataset), size=500), ]
# Splitting our dataset into 80% - 20% format
train_set<-data.frame(splitter(dataset,0.8)[1])
test_set<-data.frame(splitter(dataset,0.8)[2])

# Evaluating through KNN and visualizing the results (Euclidean distancing)
k_value <- c(1, 3, 5, 7, 9) #  Show accuracy for these k values
accuracy_rate <- c(evaluate_KNN(train_set, test_set, 1, 'AQI_Category', 'Euclidean'),
                   evaluate_KNN(train_set, test_set, 3, 'AQI_Category', 'Euclidean'),
                   evaluate_KNN(train_set, test_set, 5, 'AQI_Category', 'Euclidean'),
                   evaluate_KNN(train_set, test_set, 7, 'AQI_Category', 'Euclidean'),
                   evaluate_KNN(train_set, test_set, 9, 'AQI_Category', 'Euclidean'))
                    #  Distances will be calculated in Euclidean Method
                   #  'AQI_Category' is the target label
plotting_df_euclidean <- data.frame(k_value, accuracy_rate)
print (plotting_df_euclidean)
plotting_df_euclidean$k_value<-as.character(plotting_df_euclidean$k_value)

library(ggplot2)
ggplot(plotting_df_euclidean, aes(x = k_value, y = accuracy_rate )) +
geom_bar(stat = "identity",color="black",  fill=c("purple", "pink", "red", "cyan", "violet"))+
ggtitle("Accuracy of KNN model with Euclidean distance and different K values") +
#Title/heading of the plot
xlab("Values of K nearest neighbours") + #Label of x axis
ylab("Accuracy rate") #Label of y axis
```

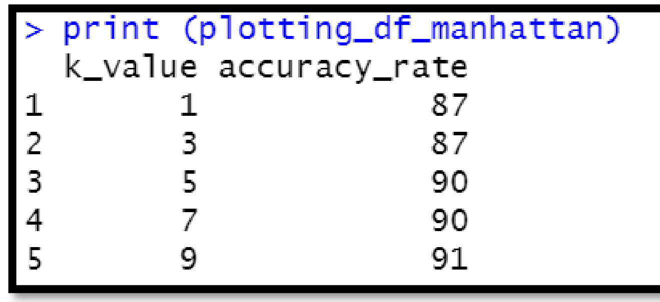
```
# Evaluating through KNN and visualizing the results (Manhattan distancing)
k_value <- c(1, 3, 5, 7, 9) #  Show accuracy for these k values
accuracy_rate <- c(evaluate_KNN(train_set, test_set, 1, 'AQI_Category', 'Manhattan'),
                   evaluate_KNN(train_set, test_set, 3, 'AQI_Category', 'Manhattan'),
                   evaluate_KNN(train_set, test_set, 5, 'AQI_Category', 'Manhattan'),
                   evaluate_KNN(train_set, test_set, 7, 'AQI_Category', 'Manhattan'),
                   evaluate_KNN(train_set, test_set, 9, 'AQI_Category', 'Manhattan'))
                    #  Distances will be calculated in Manhattan Method
                   #  'AQI_Category' is the target label

plotting_df_manhattan <- data.frame(k_value, accuracy_rate)

print (plotting_df_manhattan)

plotting_df_manhattan$k_value<-as.character(plotting_df_manhattan$k_value)

library(ggplot2)
ggplot(plotting_df_manhattan, aes(x = k_value, y = accuracy_rate )) +
geom_bar(stat = "identity" ,color="black",  fill=c("orange", "yellow", "green", "brown",
"salmon"))+ # Setting colors
ggtitle("Accuracy of KNN model with Manhattan distance and different K values") +
# Title/heading of the plot
xlab("Values of K nearest neighbours") + # Label of x axis
ylab("Accuracy rate") # Label of y axis
```

Result: On execution we get results like the following.



```
> print (plotting_df_euclidean)
  k_value accuracy_rate
1       1            93
2       3            90
3       5            88
4       7            87
5       9            89
```

Fig.12: Result of KNN with Euclidean distance



```
> print (plotting_df_manhattan)
  k_value accuracy_rate
1       1            87
2       3            87
3       5            90
4       7            90
5       9            91
```

Fig.13: Result of KNN with Manhattan distance

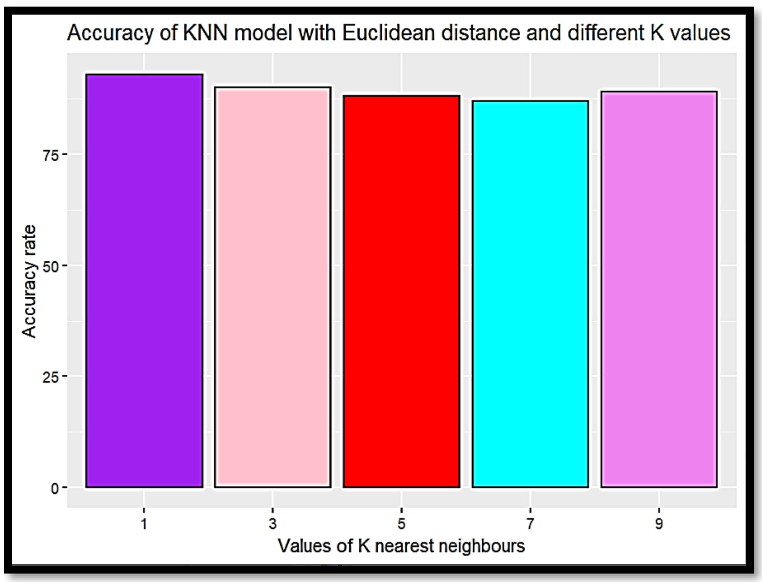


Fig.14: KNN accuracy values with Euclidean distance

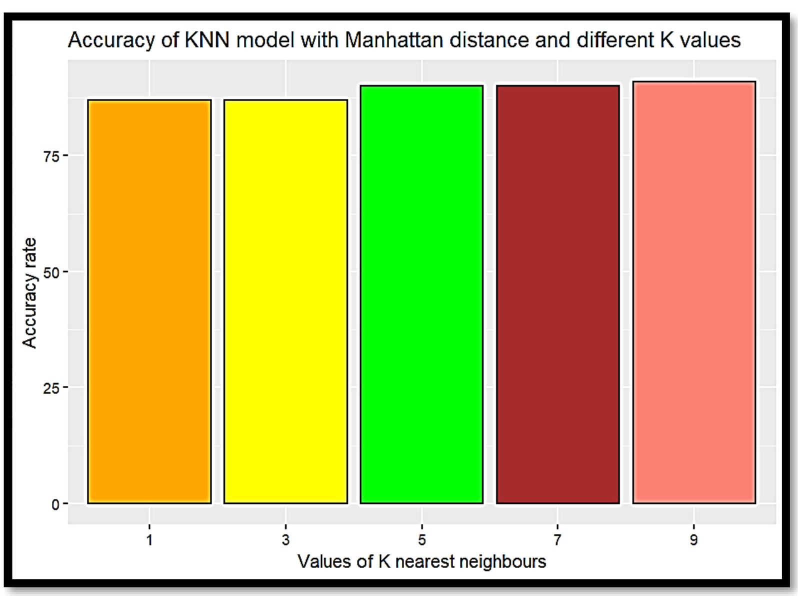


Fig.15: KNN accuracy values with Manhattan distance

<u>Discussion and Conclusion:</u> From the results we can see that Euclidean distancing yielded slightly more accurate results than Manhattan distancing. So for datasets like this this distancing method can be used. Our dataset can also be made smaller by taking samples for faster training and testing. We believe our efforts have satisfied the project requirements and it is a success.

Note: Our project can be downloaded from the following link:
https://drive.google.com/drive/u/0/folders/1dVGvh-KjkfrtcuT92or4C_hpwd54I4yo