PROJECT REPORT

ANTHROPOGENIC AIR PREDICTION USING MACHINE LEARNING

Submitted towards the partial fulfillment of the criteria for award of Post Graduate In Data Analytics by Imarticus

Submitted By:

HARIHARASUDHAN P (IL031248)

Course and Batch: Post Graduate in Data Analytics & Feb-2022(PGA-22)

Abstract

Predicting air quality is necessary step to be taken by government as it is becoming the major concern among the health of human beings. Air quality Index measure the quality of air. Various air pollutants causing air pollution are Carbon dioxide, Nitrogen dioxide, carbon monoxide etc…that are released from burning of natural gas, coal and wood, industries, vehicles etc. Air Pollution can cause severe disease like lungs cancer, brain disease and even lead to death. Machine learning algorithms helps in determining the air quality index. Various research is being done in this field but still results are still not accurate. Dataset are available from Kaggle, air quality monitoring sites and divided into two Training and Testing. With help of Machine Learning algorithms.

Keywords:

Air quality index, Carbon dioxide, Nitrogen dioxide, Carbon monoxide Machine Learning algorithms.

Acknowledgements

We are using this opportunity to express my gratitude to everyone who supported us throughout the course of this individual project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, we were fortunate to have __________________ as our mentor. He/She has readily shared his immense knowledge in data analytics and guide us in a manner that the outcome resulted in enhancing our data skills.

We wish to thank, all the faculties, as this project utilized knowledge gained from every course that formed the PGA program.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Date: August 30, 2022 HARIHARASUDHAN P *(IL031248)*

Place: Chennai

Certificate of Completion

I hereby certify that the project titled "**ANTHROPOGENIC AIR PREDICTION USING MACHINE LEARNING**" was undertaken and completed under my supervision by HARIHARASUDHAN P from the batch of PGA (Feb-2022)

Mentor: Mentor Name (if any)

Date: August 30, 2022

Place – Chennai

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 AIR POLLUTION:

Air pollution is dangerous for human health and should be decrease fast in urban and rural areas so it is necessary to predict the quality of air accurately. There are many types of pollution like water pollution, air pollution, soil pollution etc but most important among these is air pollution which should be controlled immediately as humans inhale oxygen through air. There are various causes of air pollution. Outdoor air pollution caused by industries, factories, vehicles and Indoor air pollution is caused if air inside the house is contaminated by smokes, chemicals, smell. Two types of Pollutants that is causing air pollution are Primary Pollutants and Secondary Pollutants.

1.2 PRIMARY POLLUTANTS INCLUDE:

Carbon dioxide (CO2): Carbon dioxide is playing an important role in causing air pollution. It is also named as Greenhouse gas. Global warming a major concern caused by increase in carbon dioxide in air.CO2 is exhale by Human.CO2 is also released by burning of fossil fuels. Sulphur oxide (SOX): Sulphur dioxide (SO2) released by burning coal and petroleum. It is released by various industries. When react with Catalyst (NO2), results in H2SO4 causing acid rains that forms the major cause of Air Pollution. Nitrogen oxide (NOX): Most commonly Nitrogen dioxide (NO2) that is caused by thunderstorm, rise in temperature. Carbon monoxide (CO): -Carbon monoxide is caused by burning of coal and wood. It is released by Vehicles.It is odorless, colorless, toxic gas. It forms a smog in air and thus a primary pollutant in air pollution. Toxic metals –Example are Lead and Mercury Chlorofluorocarbons (CFC): -Chlorofluorocarbons released by air conditioners, refrigerators which react with other gases and damage the Ozone Layer. Therefore, Ultraviolet Rays reach the earth surface and thus cause harms to human beings. Garbage, Sewage and industrial Process also causes Air Pollution. Particles originating from dust storms, forest, volcanoes in the form of solid or liquid causing air pollution.

1.3 SECONDARY POLLUTANTS INCLUDE:

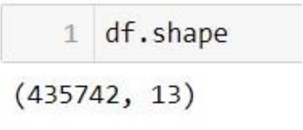
Ground Level Ozone: It is just above the earth surface and forms when Hydrocarbon react with Nitrogen Oxide in the sunlight presence. Acid Rain: When Sulphur dioxide react with nitrgendioxide, oxygen and water in air thus causing acid rain and fall on ground in dry or wet form. The difference between Primary Pollutants and Secondary Pollutants is Primary Pollutants are those which are released into air directly from Source whereas Secondary Pollutants are those which are formed by reacting with either primary pollutants or with other atmospheric component. There are various pollutants causing air pollution but PM 2.5 being the major air pollutant as proposed by the author (J. Angelin Jebamalar & A. Sasi Kumar,2019) and comes out with the best results in predicting level of PM 2.5 in their research [13] . Logistic regression and autoregression help in determining the level of PM2.5 [4]. The day wise prediction of pollutant level [1] was removed by various authors further by predicting hourly wise data using different algorithm. Benzene concentration can also account into air pollution and its concentration can be determined with CO [7] . These are the causes of air pollution.

Air pollution is causing harmful effects on human beings and plants. It causes the less threatening diseases like irritation in throat, nose. Headache to most severe disease like Respiratory Problems, shortness of breath, Lungs Cancer, brain disease, kidney disease and even leads to death. There are masks which protect us from increasing air pollution and various acts are there to control air pollution. It is also necessary to create awareness among human being about air pollution. It is necessary to predict the air quality accurately. Various traditional methods are there to measure it but results are not accurate and it involves a lot of mathematical calculations. Machine Learning a subset of Artificial Intelligence has an important role in predicting air quality. Various researches are being done on measuring Air quality Index by using Machine Learning algorithms. So, to control Air Pollution first necessary step is to measure accurately the Air Index Quality. Machine Learning algorithms plays an important role in measuring air quality index accurately. Various algorithm are compared on the basis of different condition in different areas and Neural Network comes out with best results.

2. IMPORTANCE OF PROJECT:

Having clean air to breathe is necessary for good health. Poor air quality reduces quality of life. Some air pollutants are irritants. Some smell bad. Some air pollutants can cause respiratory disease or even cancer. Air quality is important both indoors and outdoors at early stages with higher accuracy is an important task. Ground level ozone, particulate matter and allergens are common outdoor air pollutants. Secondhand smoke, mold and radon are common indoor air pollutants. Poor air quality may limit people’s ability or opportunity to be physically active. People with preexisting medical conditions such as asthma, emphysema or COPD are at greater risk from poor air quality. Good air quality is an important livability indicator for a healthy community. Collecting all the past data, analyzing it with the help of different algorithms, and comparing the results.

3. DATASET:

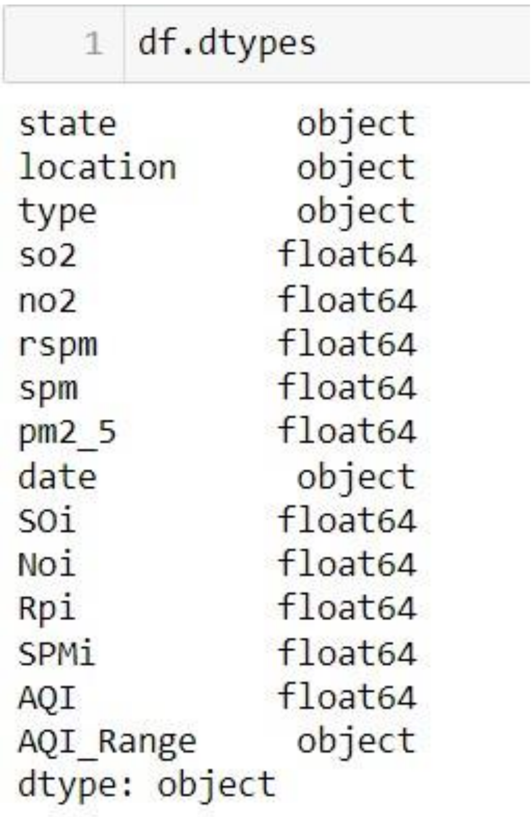


```
df.shape
```

```
(435742, 13)
```

We see the data frame has 435742 observations and 13 features, where 13 features are independent features and I add two columns they are target features is float and binary class as categorical.

1. We some float and object are there in dtypes



```
df.dtypes
```

```
state          object
location       object
type           object
so2           float64
no2           float64
rspm          float64
spm           float64
pm2_5         float64
date           object
SOi           float64
Noi           float64
Rpi           float64
SPMi          float64
AQI           float64
AQI_Range      object
dtype: object
```

As per data attribute information we have, “state”, ”Location”, ” type”, ” date” and ” AQI_Range” variables are object types and other features are float type but first 13 are independent features 14th and 15th both are dependent features for our project.

Independent Features description:

1. **State:** State of the country-(Categorical)

2. **Location:** Location of the state-(Categorical)

3. **Type:** Details of the area whether it’s a Residential, Rural and other areas (or) Industrial area-(categorical)

4. **so2:** Sulfur dioxide is a colorless, reactive air pollutant with a strong odor it’s a float continuous value-(Numerical)

5. **no2:** Nitrogen dioxide is a gaseous air pollutant composed of nitrogen and oxygen it's a mixture of gases is a float continuous value-(Numerical)

6. **Rspm:** Respirable Suspended Particulate Matter size of these pollutants is very tiny, which makes it virtually impossible to filter them out, which are less than 10 microns in diameter is a float continuous value-(Numerical)

7. **Spm:** Suspended Particulate Matter are finely divided solids or liquids that may be dispersed through the air from combustion processes, industrial activities or the natural sources is a float continuous value-(Numerical)

8. **P_M2.5 :** Fine Particulate Matter is an air pollutant that is a concern for people's health when levels in air are high is a float continuous value-(Numerical)

9. **Date:** Date of the air pollution around 1990-2015 is object value-(Numerical)

10. **SOI:** Sulfur dioxide as above mentioned separated in the range of as our convention is a float continuous value-(Numerical)

11. **NOI:** Nitrogen dioxide as above mentioned separated in the range of as our convention is a float continuous value-(Numerical)

12. **RPI:** Respirable Suspended Particulate Matter as above mentioned separated in the range of as our convention is a float continuous value-(Numerical)

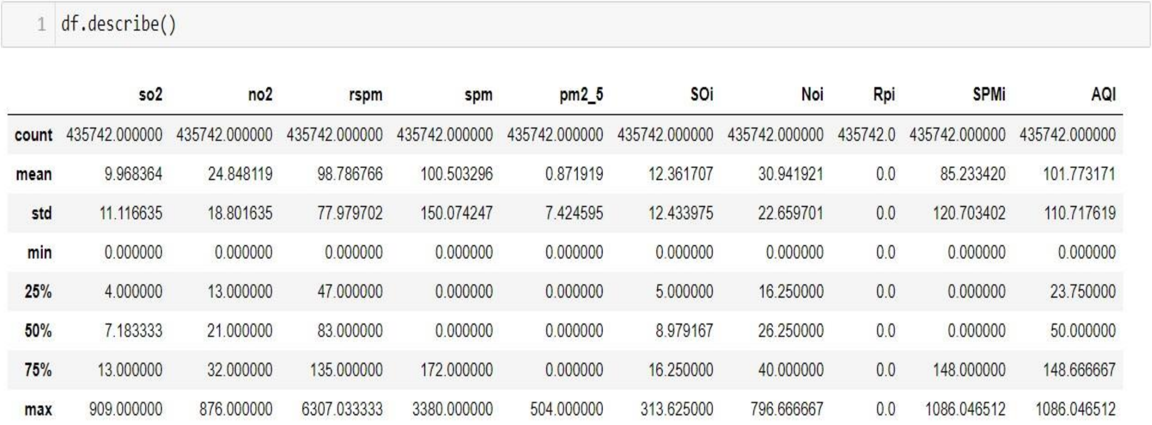
13. **SPMI:** Suspended Particulate Matter as above mentioned separated in the range of as our convention is a float continuous value-(Numerical)

Target Features description:

14. **AQI:** Air Quality Index gathered in all range of SOI, NOI, RPI, SPMI and then I take a value of NOI for AQI is a float continuous value-(Numerical)

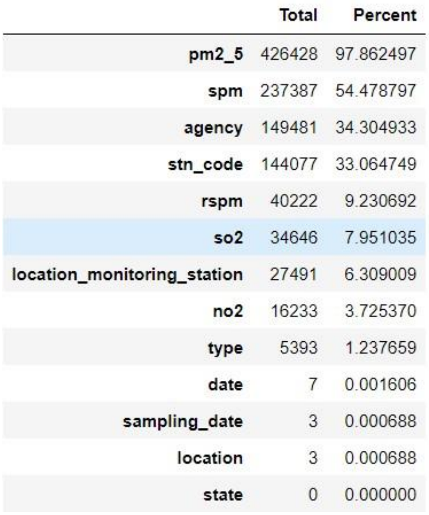
15. **AQI_Range:** Air Quality Index _ Range is a classify the value is object value-(Categorical)

4. SUMMARY OF DATA:



```
df.describe()
```

```
                so2            no2           rspm           spm            pm2_5          SOi            Noi            Rpi       SPMi           AQI
count  435742.000000  435742.000000  435742.000000  435742.000000  435742.000000  435742.000000  435742.000000  435742.0  435742.000000  435742.000000
mean        9.968364      24.848119      98.786766     100.503296       0.871919      12.361707      30.941921       0.0      85.233420     101.773171
std        11.116635      18.801635      77.979702     150.074247       7.424595      12.433975      22.659701       0.0     120.703402     110.717619
min         0.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.0       0.000000       0.000000
25%         4.000000      13.000000      47.000000       0.000000       0.000000       5.000000      16.250000       0.0       0.000000      23.750000
50%         7.183333      21.000000      83.000000       0.000000       0.000000       8.979167      26.250000       0.0       0.000000      50.000000
75%        13.000000      32.000000     135.000000     172.000000       0.000000      16.250000      40.000000       0.0     148.000000     148.666667
max       909.000000     876.000000    6307.033333    3380.000000     504.000000     313.625000     796.666667       0.0    1086.046512    1086.046512
```

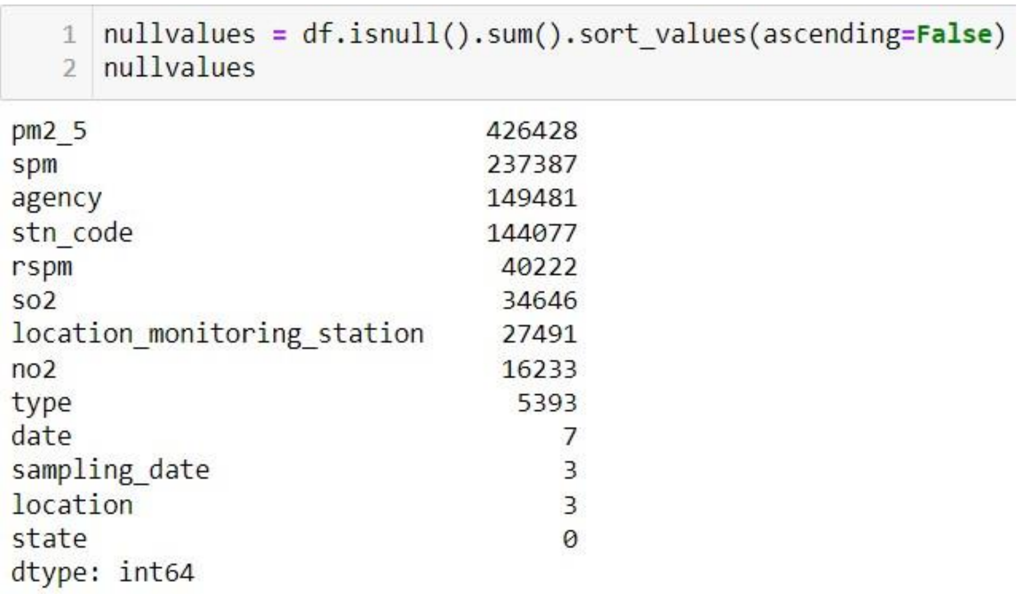


```
                              Total    Percent
                     pm2_5   426428  97.862497
                       spm   237387  54.478797
                    agency   149481  34.304933
                  stn_code   144077  33.064749
                      rspm    40222   9.230692
                       so2    34646   7.951035
location_monitoring_station   27491   6.309009
                       no2    16233   3.725370
                      type     5393   1.237659
                      date        7   0.001606
             sampling_date        3   0.000688
                  location        3   0.000688
                     state        0   0.000000
```

- As there are a total 435742 observations in a survey.
- The variable “pm2_5” and “spm” has more number of missing values, so may be removing these variables will be good option or else we can segregate with new features with reducing a missing value.
- The AQI values ranges from the normal value to max of 1086 range value. Which indicates outliers present in age, which need to be treated.
- The variables “agency”, ”stn_code”, ”rspm”, ”so2”, ”location_monitoring_station”, ”no2”, ”type” has few missing values which can be treated in later steps.

5. EXPLORATORY DATA ANALYSIS:

5.1 Treating missing data:

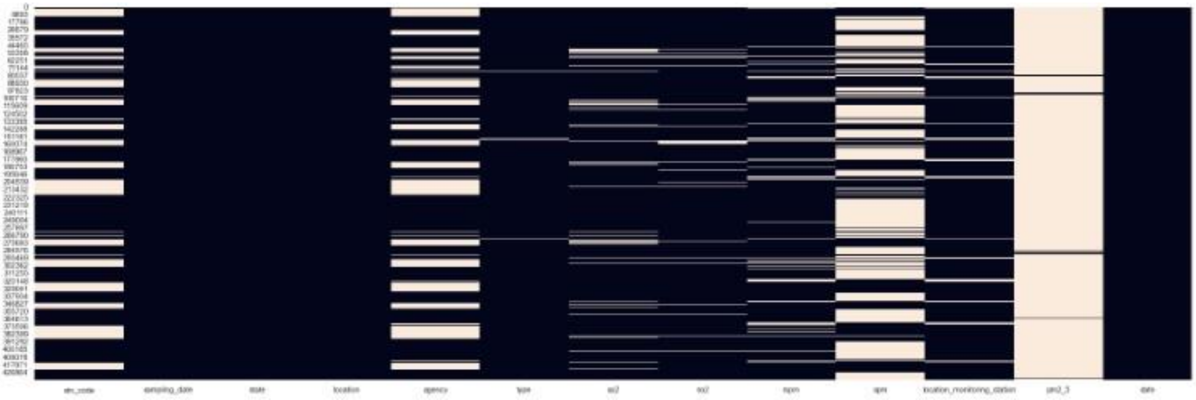


```
nullvalues = df.isnull().sum().sort_values(ascending=False)
nullvalues
```

```
pm2_5                          426428
spm                            237387
agency                         149481
stn_code                       144077
rspm                            40222
so2                             34646
location_monitoring_station     27491
no2                             16233
type                             5393
date                                7
sampling_date                       3
location                            3
state                               0
dtype: int64
```

- isnull().sum()”” returns the number of missing values in each variables

Visualization of missing values using heatmap:



There are horizontal lines in the heatmap for “stn_code”, “agency”, “so2”, “no2”, “rspm”, “spm”, “location_monitoring_station”, “pm2_5”. Which would correspond to probable missing values.

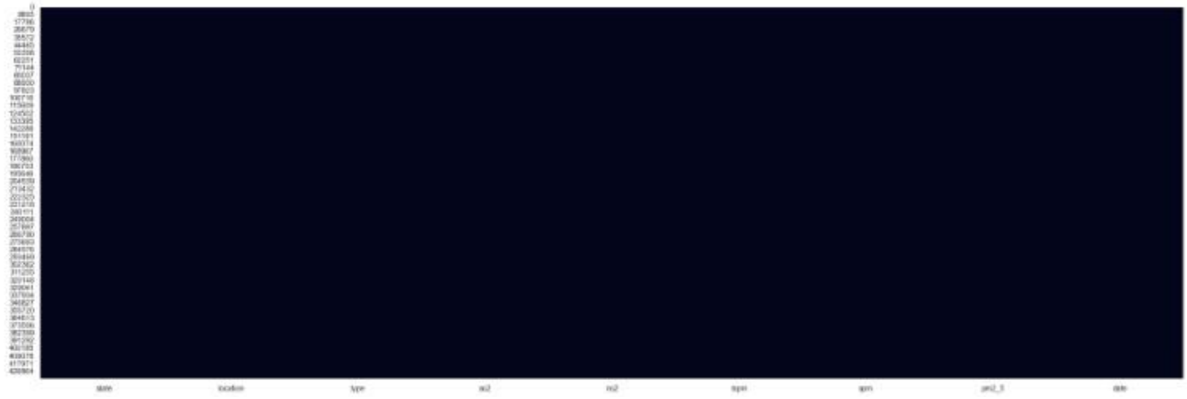
Why do you need to fill in the missing data?

1. Because most of the machine learning models that you want to use will provide an error if you pass NaN values into it.

Methods used to treat missing values:

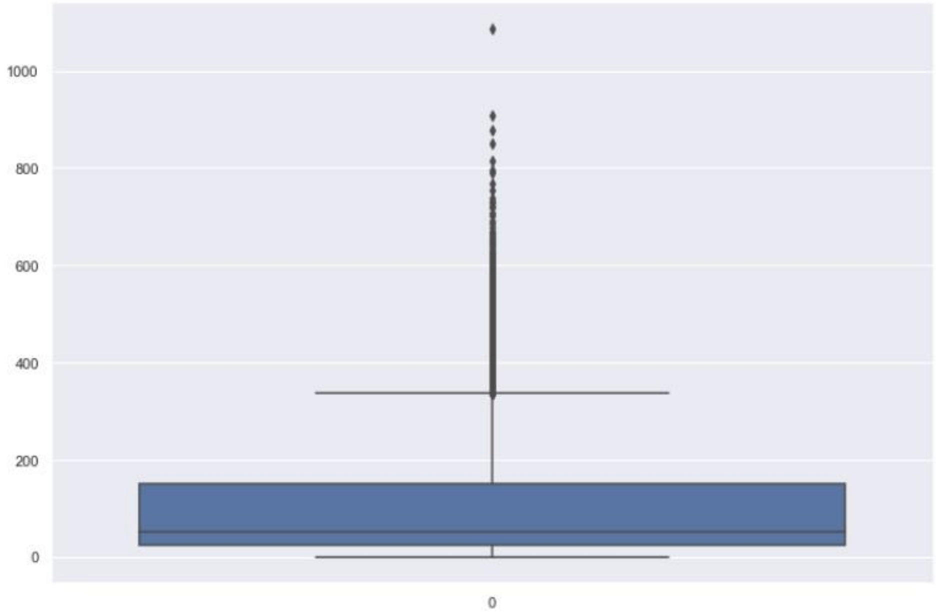
1. Fill the missing data with the mean or median value if it's a numerical variable.
2. Filling the missing data with mode if it's a categorical value.
3. Deleting the columns with complete missing values.

Visualization of missing values using the heatmap after treatment:



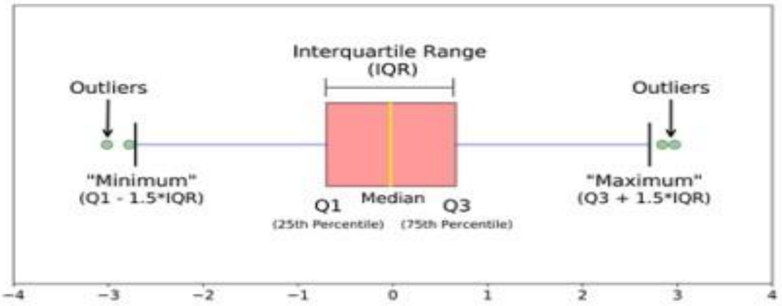
5.2 Discover Outliers for AQI using Box plot :

An Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.



Importance of treating outliers:

1. If the outliers are not removed, the model accuracy may decrease.



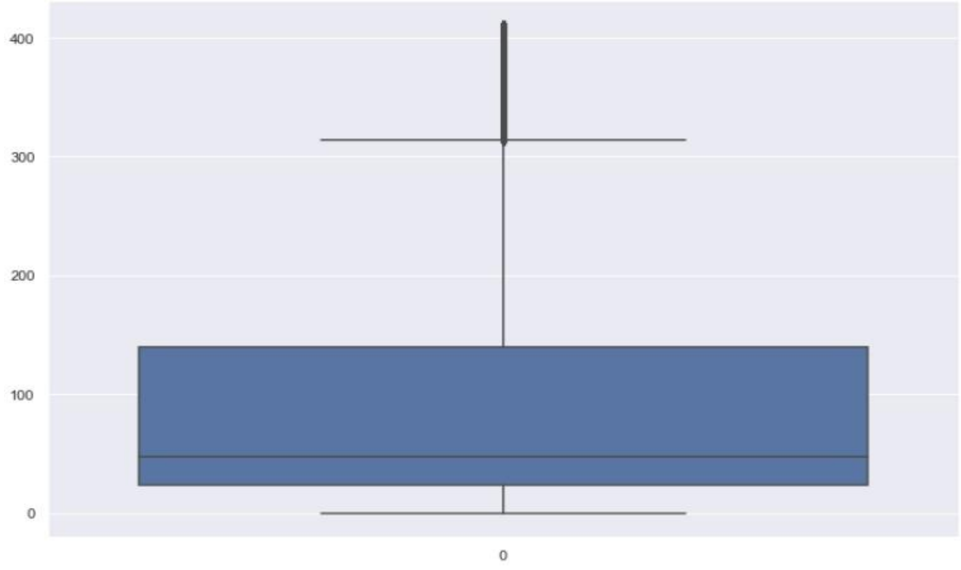
How to treat outliers:

1. Calculate the 1st and 3rd quartiles (Q1, Q3)

2. Compute IQR=Q3-Q1

3. Compute lower bound = (Q1–1.5*IQR), upper bound = (Q3+1.5*IQR)

Visualizing boxplot for AQI after treatment:

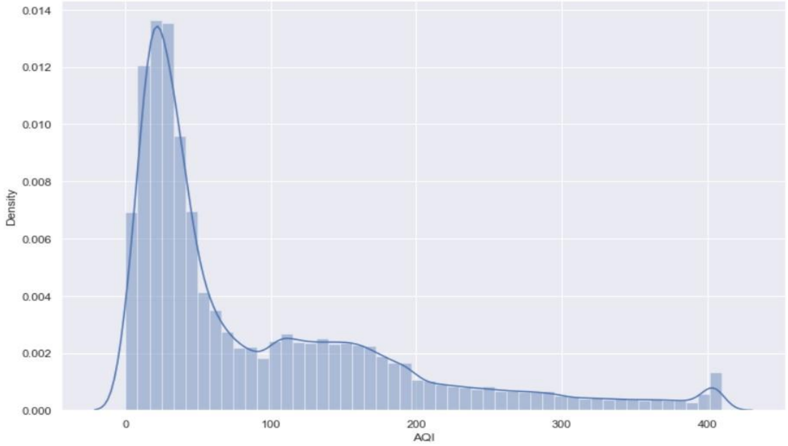


Here some of the outliers are there because I what to know the AQI_Range for classification model that’s why I didn’t remove the some of the outlier for feature purpose. Otherwise I remove the outliers for regression model for AQI.

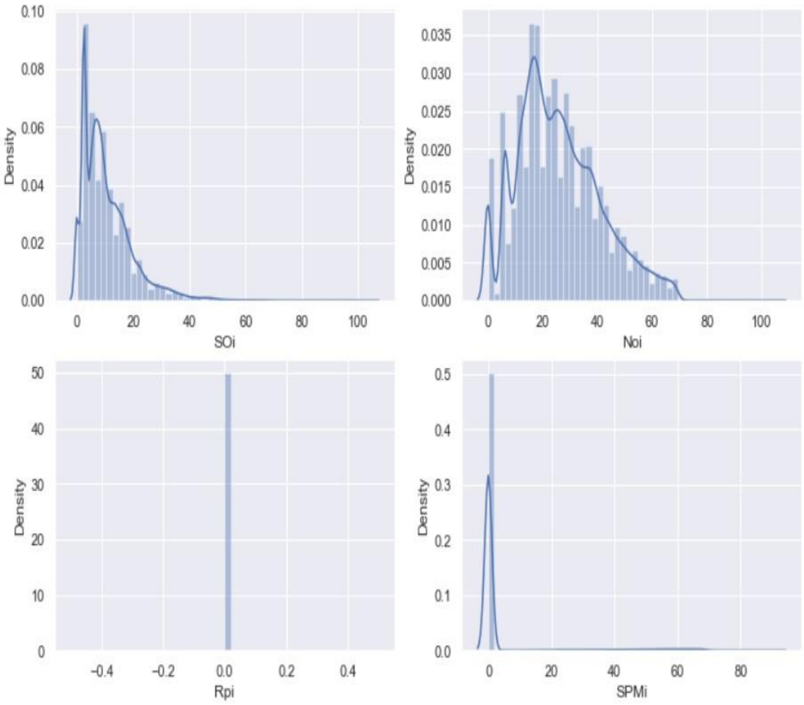
5.3. Distribution of Variables

Distribution of numeric independent variables using dist plot:

distplot() function is used to plot the distplot. The distplot represents the univariate distribution of data i.e., the data distribution of a variable against the density distribution.



Here variable "AQI" the range of 0 to 10 its slightly right skew but is near normally distributed.

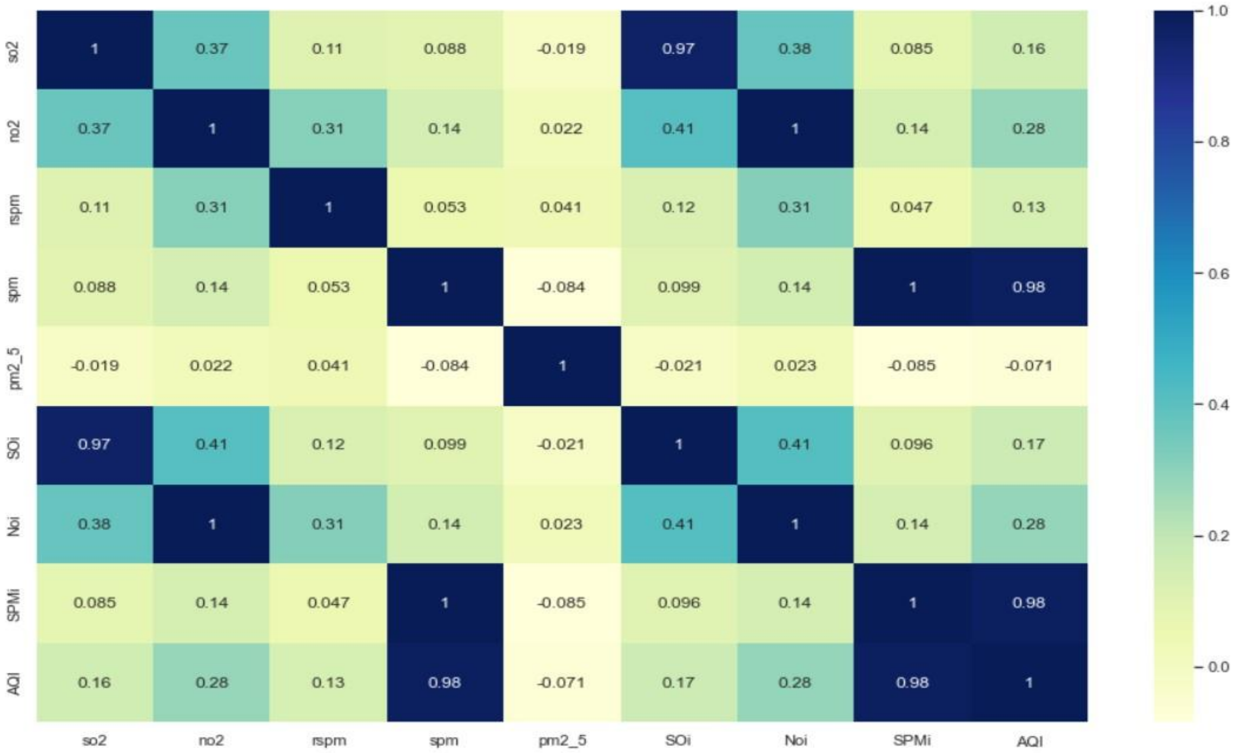


It can be seen that the variable "Soi", "Noi", "Rpi", and "SPMi" have slight right skew, but has a right tail. They are almost near normally distributed. But the Rpi doesn’t distributed so remove the feature.

5.4. Multivariate Analysis:

Correlation is a statistic that measures the degree to which two variables move with each other. A correlation coefficient near 1 indicates a strong relationship between them; a weak correlation indicates the extent to which one variable increases as the other decreases. Correlation among multiple variables can be represented in the form of a matrix. This allows us to see which variables are correlated.

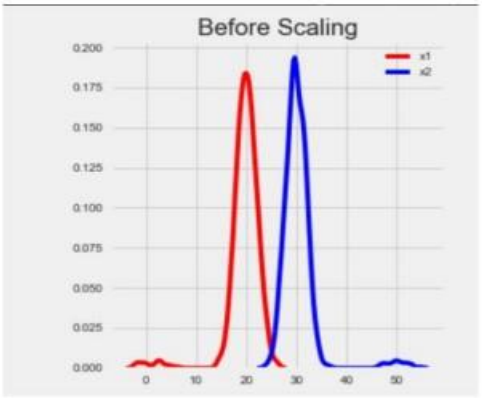
Heat map for the correlation matrix:

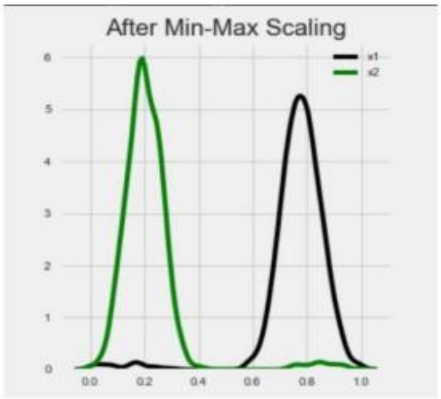


From the above heatmap, it can be seen that the continuous variables are not highly correlated but some of them are correlated but not it to be same row and column.

6. FEATURE SELECTION USING MINMAX SCALER:

There are many ways of data scaling, where the minimum feature is made equal to zero and the maximum feature equal to one. MinMax Scaler shrinks the data within the given range, usually from 0 to 1. It transforms data by scaling features to a given range. It scales the values to a specific value range without changing the shape of the original distribution.

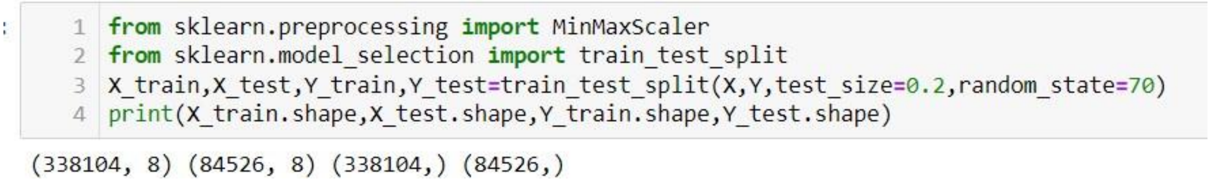




7. TRAINING AND TEST DATASET:

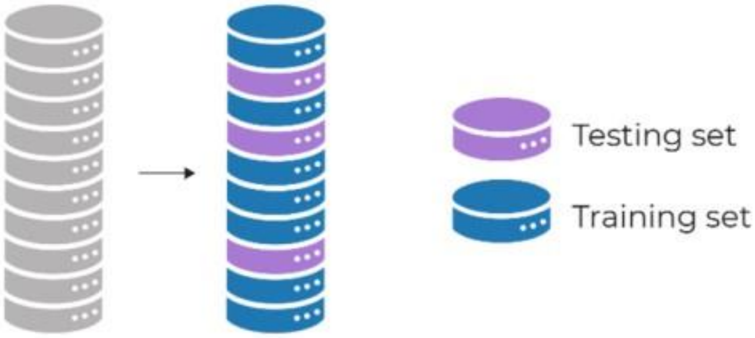
Creating the training dataset with 70% of original data and the remaining 30% as test data.

- **Train Dataset:** Used to fit the machine learning model.
- **Test Dataset:** Used to evaluate the fit machine learning model.



```
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=70)
print(X_train.shape,X_test.shape,Y_train.shape,Y_test.shape)
```

```
(338104, 8) (84526, 8) (338104,) (84526,)
```



The train_test_split function of the sklearn. Model_selection package in python splits arrays or matrices into random subsets for train and test data, respectively. To use the train_test_split function to slove classification problems the same way you do for regression analysis.

8. MODEL FOR REGRESSION:

8.1 LINEAR REGRESSION:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable. The linear regression model provides a sloped straight line representing the relationship between the variables.

8.2 DECISION TREE REGRESSOR:

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete.

8.3 RANDOM FOREST REGRESSOR:

Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, Commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Boostrap.

8.4 XGBOOST REGRESSOR:

Extreme Gradient Boosting, or XGBoost for short, is an efficient open-source implementation of the gradient boosting algorithm. As such, XGBoost is an algorithm, an open-source project, and a Python library.

Gradient boosting refers to a class of ensemble machine learning algorithms that can be used for classification or regression predictive modeling problems. Ensembles are constructed from decision tree models. Trees are added one at a time to the ensemble and fit to correct the prediction errors made by prior models. This is a type of ensemble machine learning model referred to as boosting. Models are fit using any arbitrary differentiable loss function and gradient descent optimization algorithm. This gives the technique its name, "*gradient boosting*," as the loss gradient is minimized as the model is fit, much like a neural network.

8.5 ADABOOST REGRESSOR:

Boosting refers to a class of machine learning ensemble algorithms where models are added sequentially and later models in the sequence correct the predictions made by earlier models in the sequence. Adaboost, short for "**Adaptive Boosting**," is a boosting ensemble machine learning algorithm, and was one of the first successful boosting approaches.

AdaBoost combines the predictions from short one-level decision trees, called decision stumps, although other algorithms can also be used. Decision stump algorithms are used as the AdaBoost algorithm seeks to use many weak models and correct their predictions by adding additional weak models. The training algorithm involves starting with one decision tree, finding those examples in the training dataset that were misclassified, and adding more weight to those examples. Another tree is trained on the same data, although now weighted by the misclassification errors. This process is repeated until a desired number of trees are added.

8.6 GRADIENT BOOSTING REGRESSOR:

Gradient Boosting algorithm is used to generate an ensemble model by combining the weak learners or weak predictive models. Gradient boosting algorithm can be used to train models for both regression and classification problem.

Gradient Boosting Regression algorithm is used to fit the model which predicts the continuous value. Gradient boosting builds an additive mode by using multiple decision trees of fixed size as weak learners or weak predictive models. The parameter, n_estimators, decides the number of decision trees which will be used in the boosting stages. Gradient boosting differs from AdaBoost in the manner that decision stumps (one node & two leaves) are used in AdaBoost whereas decision trees of fixed size are used in Gradient Boosting.

8.7 LASSO REGRESSION: (Lassocv & Lasso)

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of muticollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination. The acronym "LASSO" stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator.

Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) **doesn’t** result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge.

9. EVALUATION METRICS:

The essential step in any machine learning model is to evaluate the accuracy of the model. The Mean Squared Error, Mean absolute error, Root Mean Squared Error, and R-Squared or Coefficient of determination metrics are used to evaluate the performance of the model in regression analysis.

9.1 Mean absolute error:

- The Mean absolute error represents the average of the absolute difference between the actual and predicted values in the dataset. It measures the average of the residuals in the dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

Where,
$\hat{y}$ − *predicted value of y*
$\bar{y}$ − *mean value of y*

9.2 Mean squared error:

- Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2$$

9.3 Root mean squared error:

- Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2}$$

9.4 R-squared:

- The coefficient of determination or R-squared represents the proportion of the variance in the dependent variable which is explained by the linear regression model. It is a scale-free score i.e. irrespective of the values being small or large, the value of R square will be less than one.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

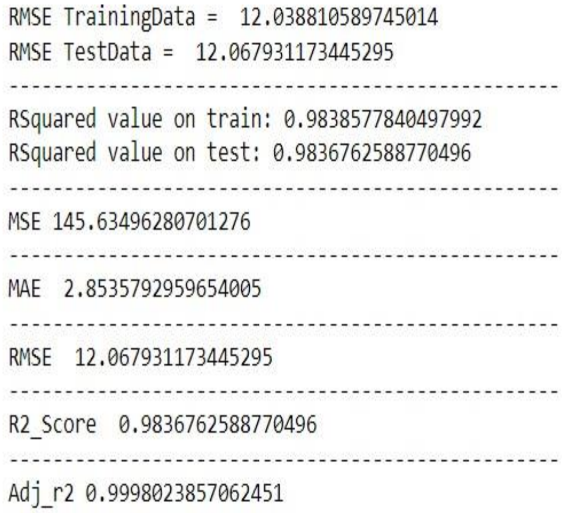
9.5 Adjusted R-squared:

- Adjusted R squared is a modified version of R square, and it is adjusted for the number of independent variables in the model, and it will always be less than or equal to R^2.In the formula below **n** is the number of observations in the data and **k** is the number of the independent variables in the data.

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

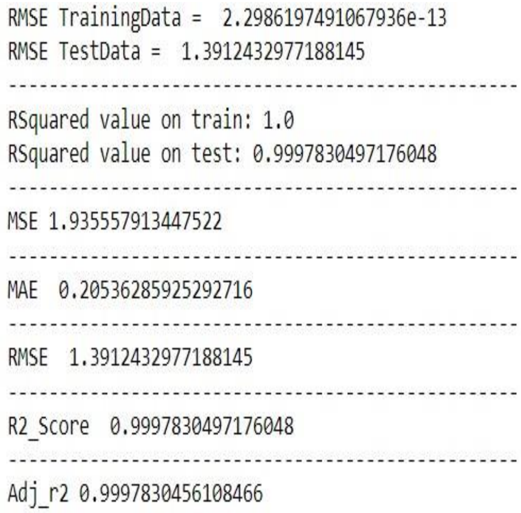
10. Comparing Model Results for Regression:

Linear Regression:



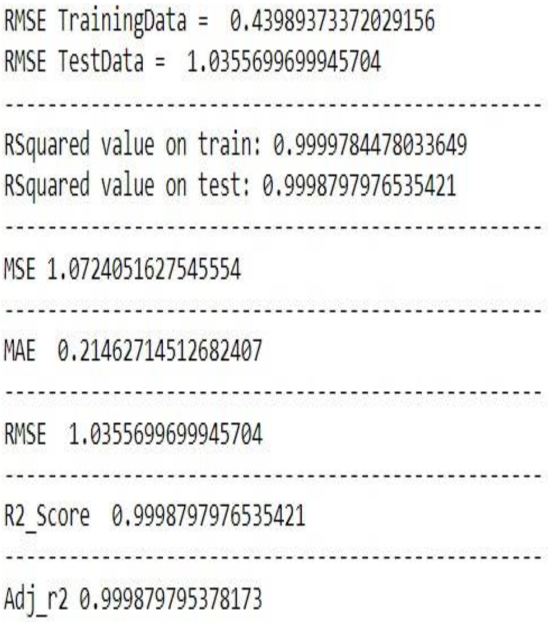
```
RMSE TrainingData =  12.038810589745014
RMSE TestData =  12.067931173445295
------------------------------------------------
RSquared value on train: 0.9838577840497992
RSquared value on test: 0.9836762588770496
------------------------------------------------
MSE 145.63496280701276
------------------------------------------------
MAE  2.8535792959654005
------------------------------------------------
RMSE  12.067931173445295
------------------------------------------------
R2_Score  0.9836762588770496
------------------------------------------------
Adj_r2 0.9998023857062451
```

Decision Tree Regressor:



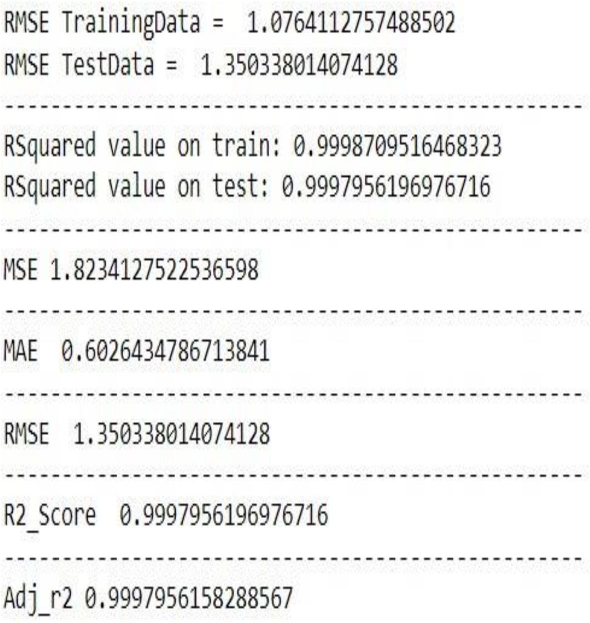
```
RMSE TrainingData =  2.2986197491067936e-13
RMSE TestData =  1.3912432977188145
------------------------------------------------
RSquared value on train: 1.0
RSquared value on test: 0.9997830497176048
------------------------------------------------
MSE 1.935557913447522
------------------------------------------------
MAE  0.20536285925292716
------------------------------------------------
RMSE  1.3912432977188145
------------------------------------------------
R2_Score  0.9997830497176048
------------------------------------------------
Adj_r2 0.9997830456108466
```

Random Forest Regressor:



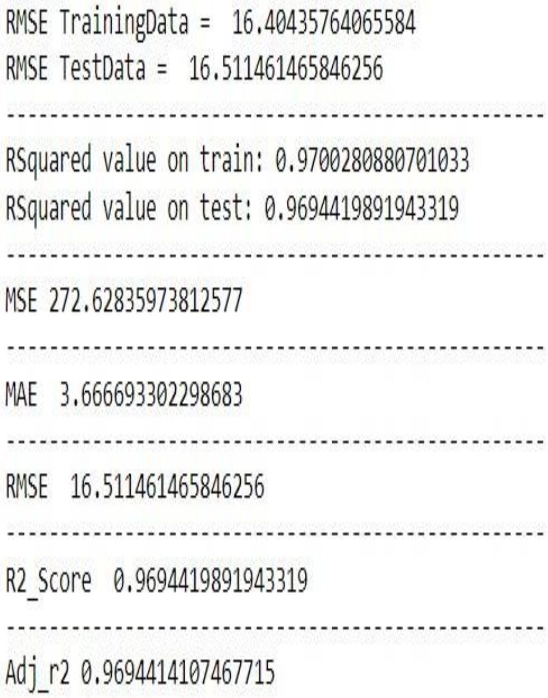
```
RMSE TrainingData =  0.43989373372029156
RMSE TestData =  1.0355699699945704
--------------------------------------------------
RSquared value on train: 0.9999784478033649
RSquared value on test: 0.9998797976535421
--------------------------------------------------
MSE 1.0724051627545554
--------------------------------------------------
MAE  0.21462714512682407
--------------------------------------------------
RMSE  1.0355699699945704
--------------------------------------------------
R2_Score  0.9998797976535421
--------------------------------------------------
Adj_r2 0.999879795378173
```

XGBoost Regressor:



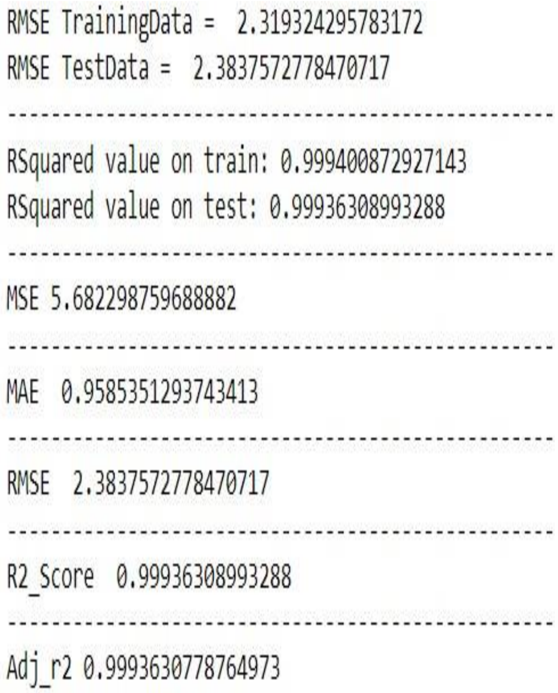
```
RMSE TrainingData =  1.0764112757488502
RMSE TestData =  1.350338014074128
--------------------------------------------------
RSquared value on train: 0.9998709516468323
RSquared value on test: 0.9997956196976716
--------------------------------------------------
MSE 1.8234127522536598
--------------------------------------------------
MAE  0.6026434786713841
--------------------------------------------------
RMSE  1.350338014074128
--------------------------------------------------
R2_Score  0.9997956196976716
--------------------------------------------------
Adj_r2 0.9997956158288567
```

Adaboost Regressor:



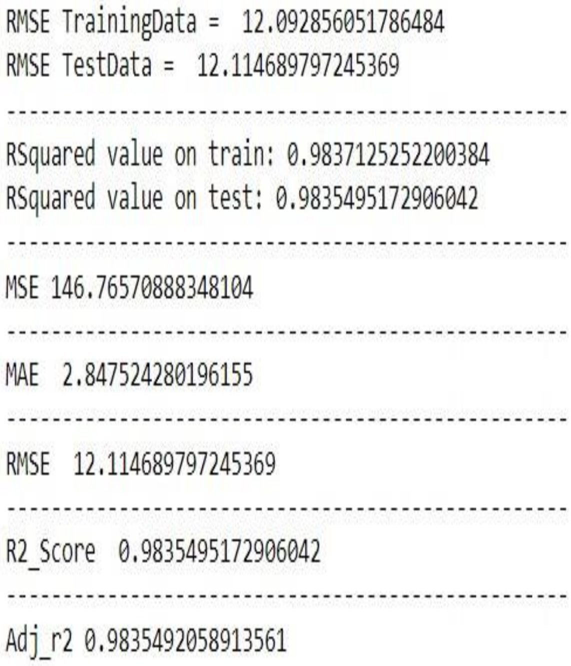
```
RMSE TrainingData =  16.40435764065584
RMSE TestData =  16.511461465846256
--------------------------------------------------
RSquared value on train: 0.9700280880701033
RSquared value on test: 0.9694419891943319
--------------------------------------------------
MSE 272.62835973812577
--------------------------------------------------
MAE  3.666693302298683
--------------------------------------------------
RMSE  16.511461465846256
--------------------------------------------------
R2_Score  0.9694419891943319
--------------------------------------------------
Adj_r2 0.9694414107467715
```

Gradient Boost Regressor:



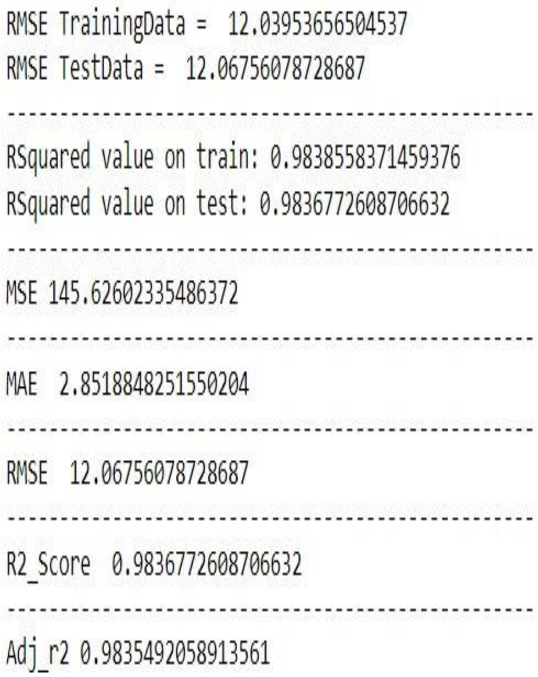
```
RMSE TrainingData =  2.319324295783172
RMSE TestData =  2.3837572778470717
--------------------------------------------------
RSquared value on train: 0.999400872927143
RSquared value on test: 0.99936308993288
--------------------------------------------------
MSE 5.682298759688882
--------------------------------------------------
MAE  0.9585351293743413
--------------------------------------------------
RMSE  2.3837572778470717
--------------------------------------------------
R2_Score  0.99936308993288
--------------------------------------------------
Adj_r2 0.9993630778764973
```

Lassocv:



```
RMSE TrainingData =  12.092856051786484
RMSE TestData =  12.114689797245369
--------------------------------------------------
RSquared value on train: 0.9837125252200384
RSquared value on test: 0.9835495172906042
--------------------------------------------------
MSE 146.76570888348104
--------------------------------------------------
MAE  2.847524280196155
--------------------------------------------------
RMSE  12.114689797245369
--------------------------------------------------
R2_Score  0.9835495172906042
--------------------------------------------------
Adj_r2 0.9835492058913561
```

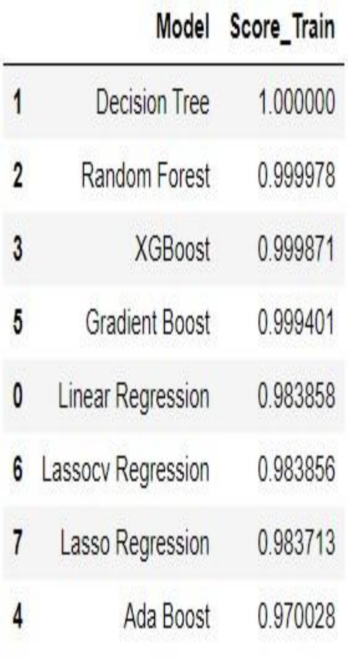
Lasso:



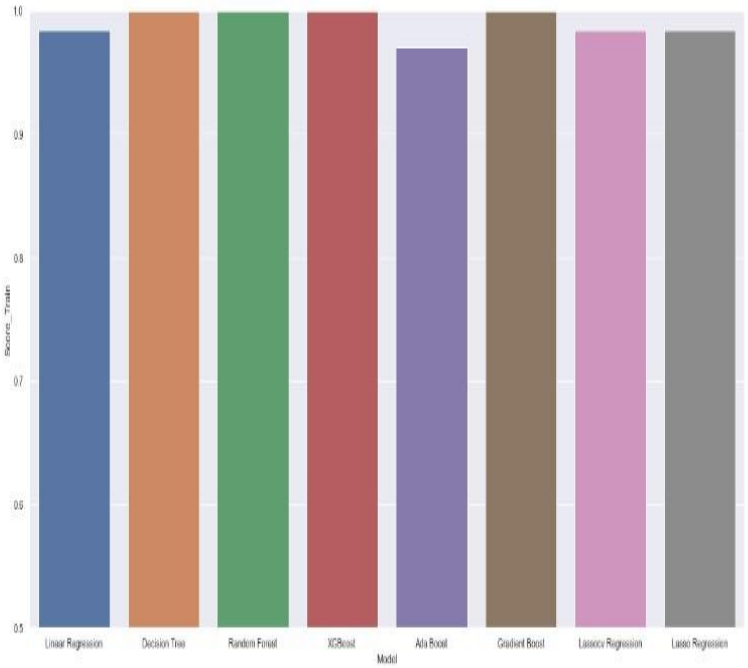
```
RMSE TrainingData =  12.03953656504537
RMSE TestData =  12.06756078728687
--------------------------------------------------
RSquared value on train: 0.9838558371459376
RSquared value on test: 0.9836772608706632
--------------------------------------------------
MSE 145.62602335486372
--------------------------------------------------
MAE  2.8518848251550204
--------------------------------------------------
RMSE  12.06756078728687
--------------------------------------------------
R2_Score  0.9836772608706632
--------------------------------------------------
Adj_r2 0.9835492058913561
```

11. Overall model result tabulation for regression both train & test:

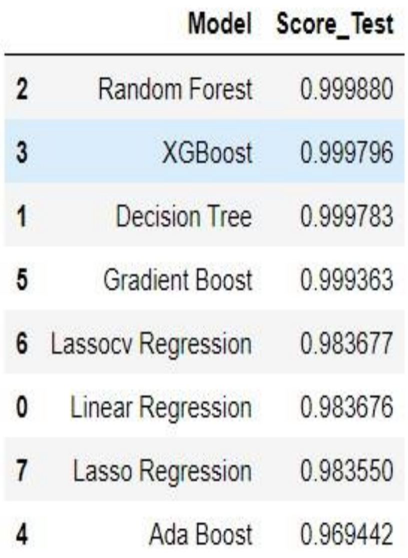
Train:



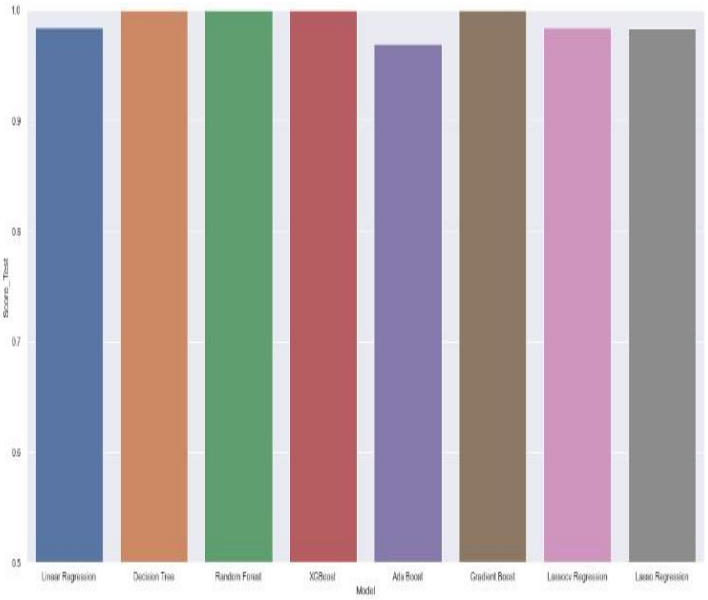
	Model	Score_Train
1	Decision Tree	1.000000
2	Random Forest	0.999978
3	XGBoost	0.999871
5	Gradient Boost	0.999401
0	Linear Regression	0.983858
6	Lassocv Regression	0.983856
7	Lasso Regression	0.983713
4	Ada Boost	0.970028



Test:



	Model	Score_Test
2	Random Forest	0.999880
3	XGBoost	0.999796
1	Decision Tree	0.999783
5	Gradient Boost	0.999363
6	Lassocv Regression	0.983677
0	Linear Regression	0.983676
7	Lasso Regression	0.983550
4	Ada Boost	0.969442



12. MODEL FOR CLASSIFICATION:

12.1 LOGISTIC REGRESSION:

Logistic regression is another powerful supervised ML algorithm used for both multiple classification and binary classification problems (when the target is categorical). It is a predictive analytic technique that is based on the probability idea. The goal of Logistic Regression is to discover a link between characteristics and the likelihood of a specific outcome.

The Logistic Regression utilizes a more sophisticated cost function, which is known as the “Sigmoid function” or “logistic function” instead of a linear function.

12.2 DECISION TREE CLASSIFIER:

A Decision Tree is a non-parametric supervised learning method. It builds a regression model in the form of a tree structure. It breaks down a data set into smaller subsets, which is called splitting.

The final result is a tree with a decision and leaf nodes. A decision node has two or more branches. The leaf node represents a class or decision. The topmost decision node in a tree that corresponds to the best predictor is called the 'root node'. The decision tree is built using different criteria like Gini index, and entropy. To build the decision tree, we used the criterion of 'entropy'. Entropy is one of the criteria used to build the decision tree. It calculates the homogeneity of the sample. The entropy is zero if the sample is completely homogeneous, and it is equal to 1 if the sample is equally divided.

12.3 RANDOM FOREST CLASSIFIER:

Random forest is a commonly-used machine learning algorithm, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Generates a random subset of features, which ensures low correlation among decision trees. This is a key difference between decision trees and random forests. While decision trees consider all the possible feature splits, random forests only select a subset of those features.

12.4 KNN CLASSIFIER:

K-Nearest Neighbours, or KNN for short, is one of the simplest machine learning algorithms and is used in a wide array of institutions. KNN is a non-parametric, lazy learning algorithm. When we say a technique is non-parametric, it means that it does not make any assumptions about the underlying data. Being a lazy learning algorithm implies that there is little to no training phase. Therefore, we can immediately classify new data points as they present themselves.

12.5 GRADIENT BOOSTING CLASSIFIER:

Gradient boosting classifier are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.

12.6 NAIVE BAYES CLASSIFIER:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in *text classification* that includes a high-dimensional training dataset.

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

12.7 SUPPORT VECTOR CLASSIFIER:

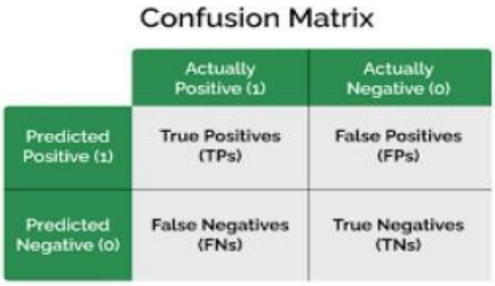
Support Vector Machine or SVM is a supervised and linear Machine Learning algorithm most commonly used for solving classification problems and is also referred to as Support

Vector Classification. SVM also supports the kernel method also called the kernel SVM which allows us to tackle non-linearity.

The objective of SVM is to draw a line that best separates the two classes of data points. SVM generates a line that can cleanly separate the two classes. How clean, you may ask. There are many possible ways of drawing a line that separates the two classes, however, in SVM, it is determined by the margins and the support vectors. The SVM then generates a hyper plane that has the maximum margin. In the case of more than 2 features and multiple dimensions, the line is replaced by a hyper plane that separates multidimensional spaces.

13. MODEL EVALUATION:

13.1Confusion matrix:



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

The confusion matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multiclass classification problems.

TP (True Positive):

Model has given prediction No, and the real or actual value was also No.

TN (True Negative):

The model has predicted yes, and the actual value was also true.

FP (False Positive):

The model has predicted Yes, but the actual value was No. It is also called a **Type-I error.**

FN (False Negative):

The model has predicted no, but the actual value was yes, it is also called as **Type-II error**.

Performance metrics of an algorithm are accuracy, precision, recall, and F1 score, which are calculated based on the above-stated TP, TN, FP, and FN.

13.2 Accuracy:

It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:
Accuracy = (TP+TN) / (TP+FP+FN+TN)

13.3 Precision:

It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:
Precision=TP / (TP+FP)

13.4 Recall:

It is defined as the out of total positive classes, how our model predicted correctly. The recall must be as high as possible.

Recall=TP / (TP+FN)

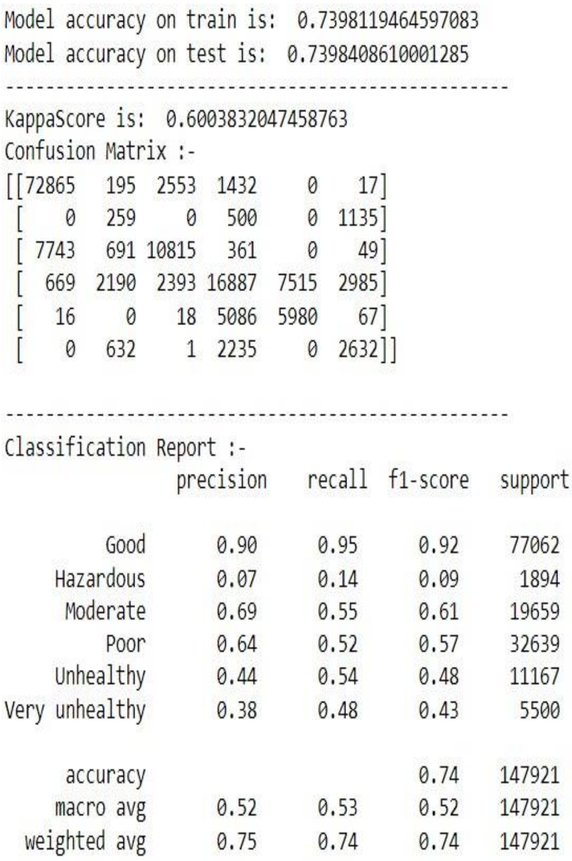
13.5 F1 score:

If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

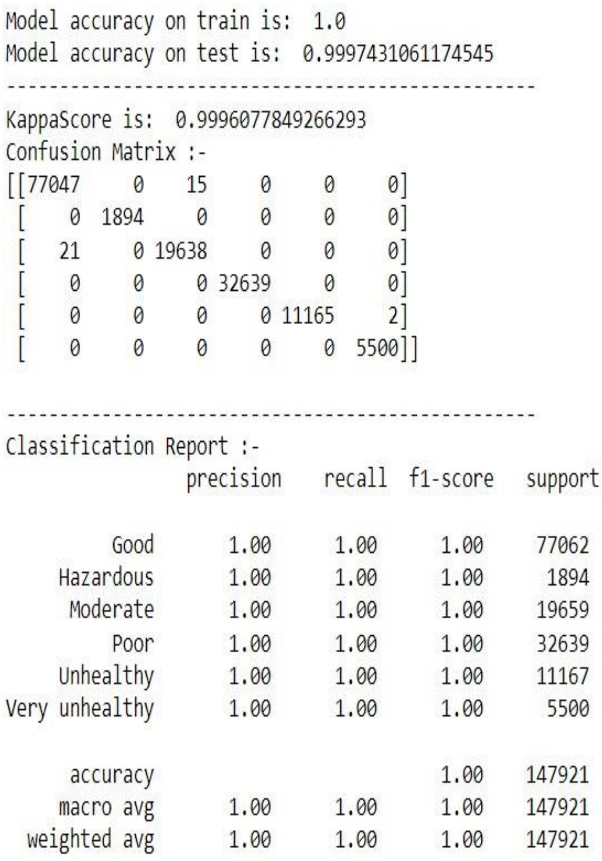
14. Comparing Model Results for Classification:

Logistic Regression:



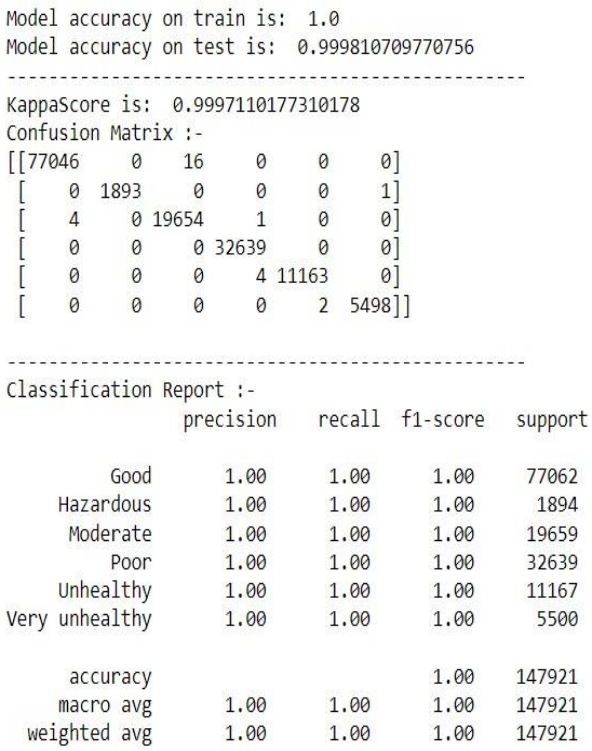
```
Model accuracy on train is:  0.7398119464597083
Model accuracy on test is:  0.7398408610001285
------------------------------------------------
KappaScore is:  0.6003832047458763
Confusion Matrix :-
[[72865   195  2553  1432     0    17]
 [    0   259     0   500     0  1135]
 [ 7743   691 10815   361     0    49]
 [  669  2190  2393 16887  7515  2985]
 [   16     0    18  5086  5980    67]
 [    0   632     1  2235     0  2632]]

------------------------------------------------
Classification Report :-
                precision    recall  f1-score   support

          Good       0.90      0.95      0.92     77062
     Hazardous       0.07      0.14      0.09      1894
      Moderate       0.69      0.55      0.61     19659
          Poor       0.64      0.52      0.57     32639
     Unhealthy       0.44      0.54      0.48     11167
Very unhealthy       0.38      0.48      0.43      5500

      accuracy                           0.74    147921
     macro avg       0.52      0.53      0.52    147921
  weighted avg       0.75      0.74      0.74    147921
```

Decision Tree Classifier:



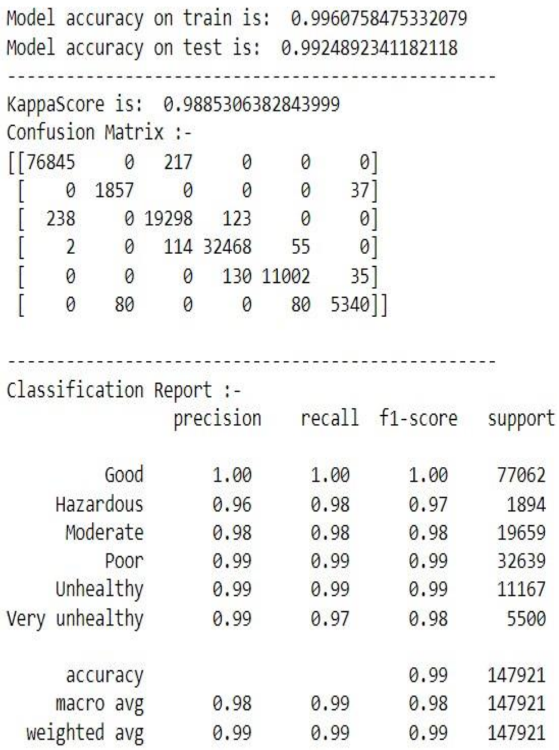
```
Model accuracy on train is:  1.0
Model accuracy on test is:  0.9997431061174545
------------------------------------------------
KappaScore is:  0.9996077849266293
Confusion Matrix :-
[[77047     0    15     0     0     0]
 [    0  1894     0     0     0     0]
 [   21     0 19638     0     0     0]
 [    0     0     0 32639     0     0]
 [    0     0     0     0 11165     2]
 [    0     0     0     0     0  5500]]

------------------------------------------------
Classification Report :-
                precision    recall  f1-score   support

          Good       1.00      1.00      1.00     77062
     Hazardous       1.00      1.00      1.00      1894
      Moderate       1.00      1.00      1.00     19659
          Poor       1.00      1.00      1.00     32639
     Unhealthy       1.00      1.00      1.00     11167
Very unhealthy       1.00      1.00      1.00      5500

      accuracy                           1.00    147921
     macro avg       1.00      1.00      1.00    147921
  weighted avg       1.00      1.00      1.00    147921
```

Random Forest Classifier:



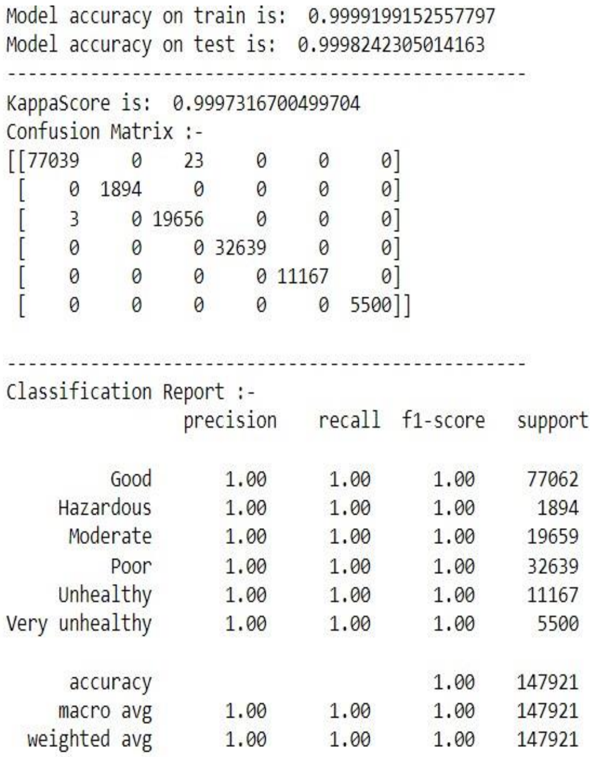
```
Model accuracy on train is:  1.0
Model accuracy on test is:  0.999810709770756
-----------------------------------------------
KappaScore is:  0.9997110177310178
Confusion Matrix :-
[[77046     0    16     0     0     0]
 [    0  1893     0     0     0     1]
 [    4     0 19654     1     0     0]
 [    0     0     0 32639     0     0]
 [    0     0     0     4 11163     0]
 [    0     0     0     0     2  5498]]

-----------------------------------------------
Classification Report :-
                precision    recall  f1-score   support

           Good       1.00      1.00      1.00     77062
      Hazardous       1.00      1.00      1.00      1894
       Moderate       1.00      1.00      1.00     19659
           Poor       1.00      1.00      1.00     32639
      Unhealthy       1.00      1.00      1.00     11167
 Very unhealthy       1.00      1.00      1.00      5500

       accuracy                           1.00    147921
      macro avg       1.00      1.00      1.00    147921
   weighted avg       1.00      1.00      1.00    147921
```

KNN-Classifier:



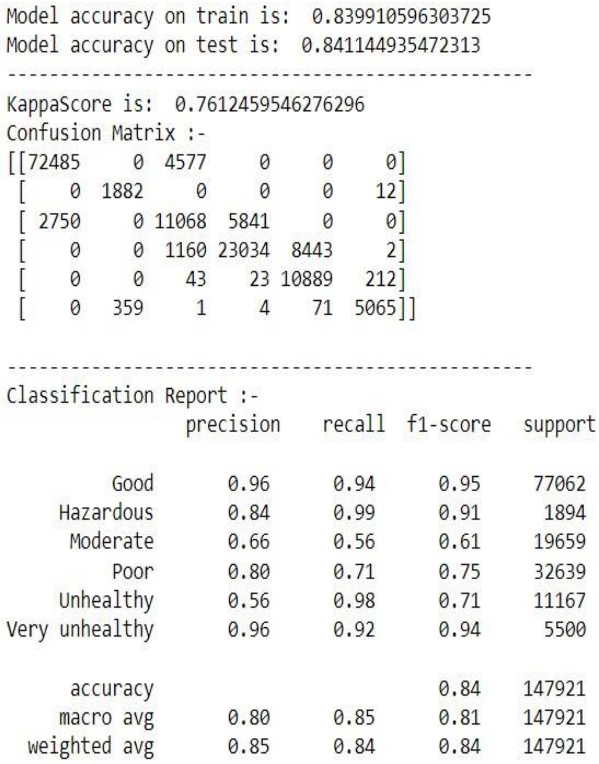
```
Model accuracy on train is:  0.9960758475332079
Model accuracy on test is:  0.9924892341182118
-----------------------------------------------
KappaScore is:  0.9885306382843999
Confusion Matrix :-
[[76845     0   217     0     0     0]
 [    0  1857     0     0     0    37]
 [  238     0 19298   123     0     0]
 [    2     0   114 32468    55     0]
 [    0     0     0   130 11002    35]
 [    0    80     0     0    80  5340]]

-----------------------------------------------
Classification Report :-
                precision    recall  f1-score   support

           Good       1.00      1.00      1.00     77062
      Hazardous       0.96      0.98      0.97      1894
       Moderate       0.98      0.98      0.98     19659
           Poor       0.99      0.99      0.99     32639
      Unhealthy       0.99      0.99      0.99     11167
 Very unhealthy       0.99      0.97      0.98      5500

       accuracy                           0.99    147921
      macro avg       0.98      0.99      0.98    147921
   weighted avg       0.99      0.99      0.99    147921
```

Gradient Boosting Classifier:



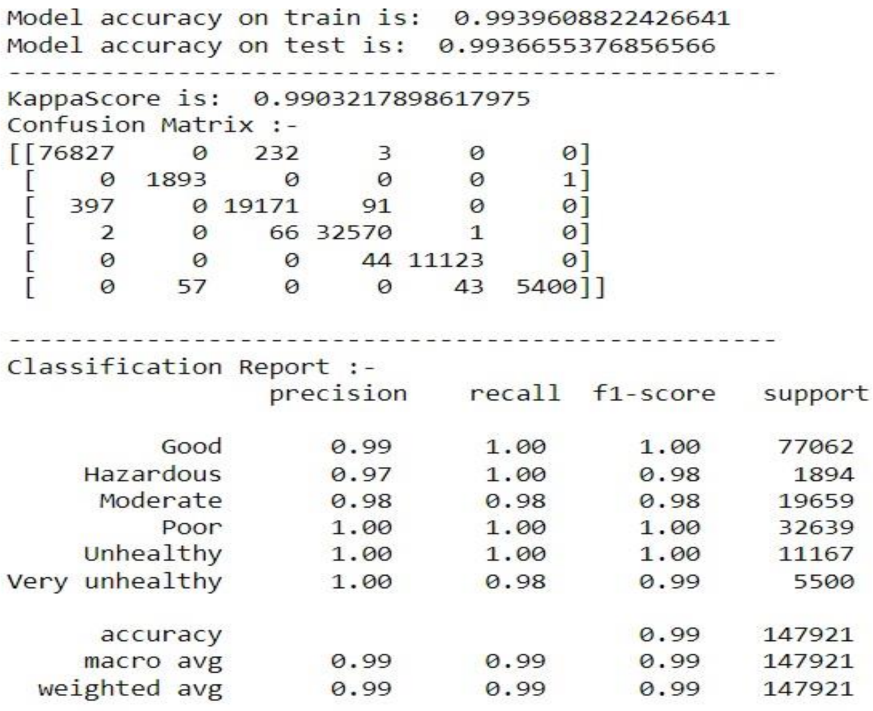
```
Model accuracy on train is:  0.9999199152557797
Model accuracy on test is:  0.9998242305014163
-----------------------------------------------
KappaScore is:  0.9997316700499704
Confusion Matrix :-
[[77039     0    23     0     0     0]
 [    0  1894     0     0     0     0]
 [    3     0 19656     0     0     0]
 [    0     0     0 32639     0     0]
 [    0     0     0     0 11167     0]
 [    0     0     0     0     0  5500]]

-----------------------------------------------
Classification Report :-
                precision    recall  f1-score   support

           Good       1.00      1.00      1.00     77062
      Hazardous       1.00      1.00      1.00      1894
       Moderate       1.00      1.00      1.00     19659
           Poor       1.00      1.00      1.00     32639
      Unhealthy       1.00      1.00      1.00     11167
 Very unhealthy       1.00      1.00      1.00      5500

       accuracy                           1.00    147921
      macro avg       1.00      1.00      1.00    147921
   weighted avg       1.00      1.00      1.00    147921
```

Naïve Bayes Classifier:



```
Model accuracy on train is:  0.839910596303725
Model accuracy on test is:  0.841144935472313
-----------------------------------------------
KappaScore is:  0.7612459546276296
Confusion Matrix :-
[[72485     0  4577     0     0     0]
 [    0  1882     0     0     0    12]
 [ 2750     0 11068  5841     0     0]
 [    0     0  1160 23034  8443     2]
 [    0     0    43    23 10889   212]
 [    0   359     1     4    71  5065]]

-----------------------------------------------
Classification Report :-
                precision    recall  f1-score   support

           Good       0.96      0.94      0.95     77062
      Hazardous       0.84      0.99      0.91      1894
       Moderate       0.66      0.56      0.61     19659
           Poor       0.80      0.71      0.75     32639
      Unhealthy       0.56      0.98      0.71     11167
 Very unhealthy       0.96      0.92      0.94      5500

       accuracy                           0.84    147921
      macro avg       0.80      0.85      0.81    147921
   weighted avg       0.85      0.84      0.84    147921
```

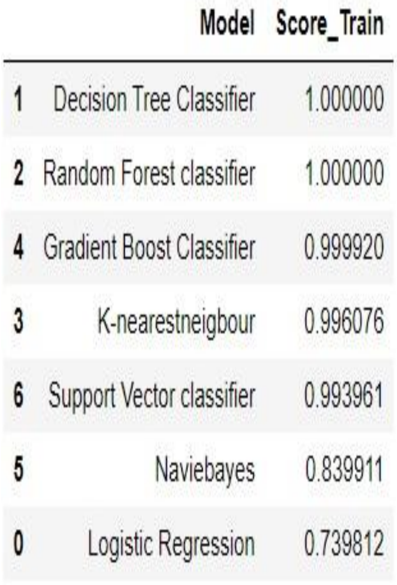
Support Vector Classifier:



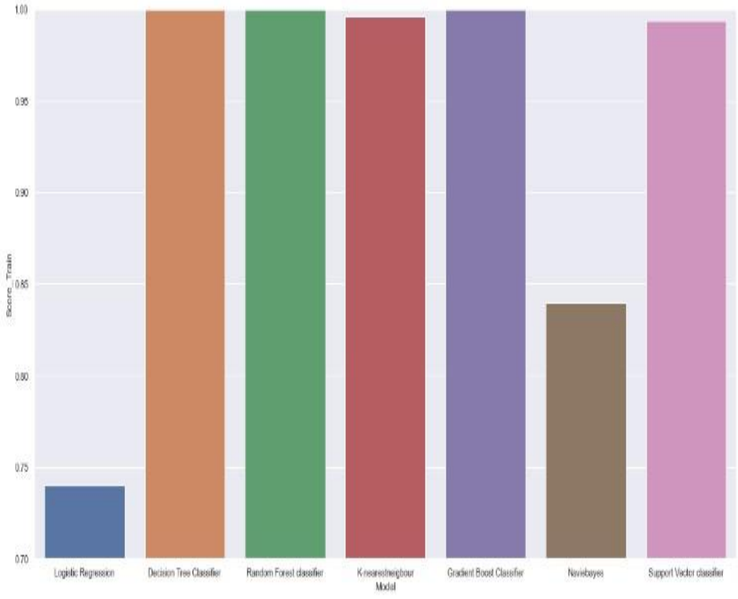
```
Model accuracy on train is:  0.9939608822426641
Model accuracy on test is:  0.9936655376856566
--------------------------------------------------
KappaScore is:  0.9903217898617975
Confusion Matrix :-
[[76827     0   232     3     0     0]
 [    0  1893     0     0     0     1]
 [  397     0 19171    91     0     0]
 [    2     0    66 32570     1     0]
 [    0     0     0    44 11123     0]
 [    0    57     0     0    43  5400]]

--------------------------------------------------
Classification Report :-
                 precision    recall  f1-score   support

           Good       0.99      1.00      1.00     77062
      Hazardous       0.97      1.00      0.98      1894
       Moderate       0.98      0.98      0.98     19659
           Poor       1.00      1.00      1.00     32639
      Unhealthy       1.00      1.00      1.00     11167
 Very unhealthy       1.00      0.98      0.99      5500

       accuracy                           0.99    147921
      macro avg       0.99      0.99      0.99    147921
   weighted avg       0.99      0.99      0.99    147921
```

15. Overall model result tabulation for Classification both train & test:

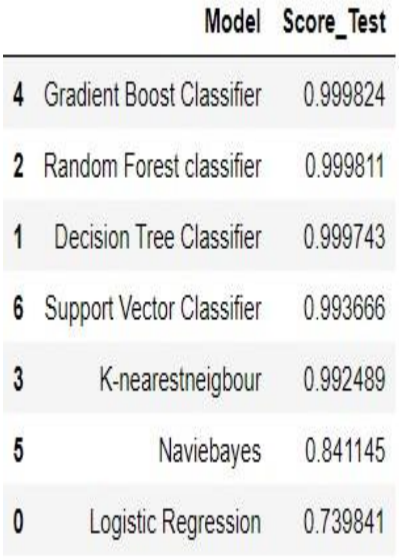
Train:



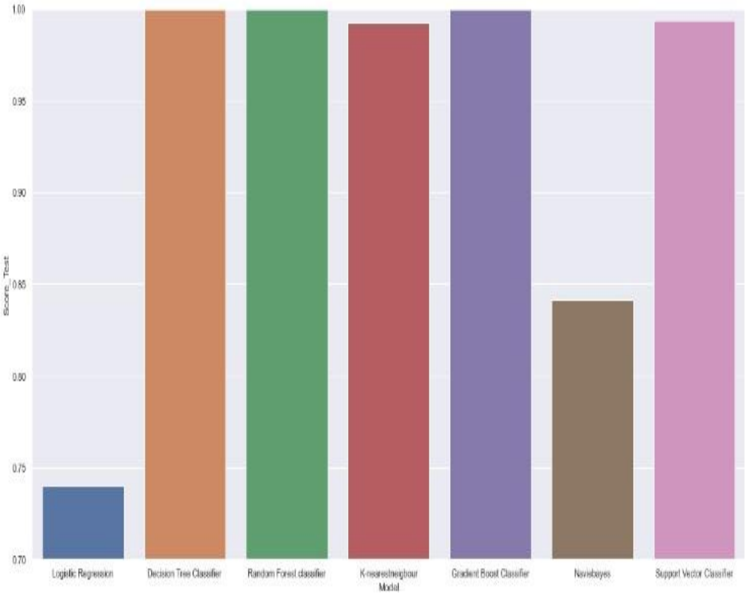
	Model	Score_Train
1	Decision Tree Classifier	1.000000
2	Random Forest classifier	1.000000
4	Gradient Boost Classifier	0.999920
3	K-nearestneigbour	0.996076
6	Support Vector classifier	0.993961
5	Naviebayes	0.839911
0	Logistic Regression	0.739812



Test:



	Model	Score_Test
4	Gradient Boost Classifier	0.999824
2	Random Forest classifier	0.999811
1	Decision Tree Classifier	0.999743
6	Support Vector Classifier	0.993666
3	K-nearestneigbour	0.992489
5	Naviebayes	0.841145
0	Logistic Regression	0.739841



16. CONCLUSION FOR REGRESSION MODEL:

- The supervised Regression learning algorithms named in the above result have been implemented on the given dataset. The performance of the models was evaluated using the RMSE, R_Squared, MSE, MAE, and R2_Score with Range of AQI Features.
- The above table shows that the Decision Tree has the highest value but Random Forest better performance measures like RMSE, RSquared, MSE, MAE, and R2_Score. Therefore, it can be concluded that the Random Forest gives a better performance comparing with other all model and score can be used to predict the existence of AQI.

17. CONCLUSION FOR CLASSIFICATION MODEL:

- The supervised classification learning algorithms named in the above result have been implemented on the given dataset. The performance of the models was evaluated using the accuracy, precision, recall, f1-score and support with multi classification as Good, Moderate, Poor, Unhealty, Very unhealty and Hazardous.
- The above table shows that the Decision Tree and Random Forest as show has the highest values but Random Forest better performance measures like Precision, Recall, f1-score, Support and accuracy. Therefore, it can be concluded that the Random Forest gives a better performance when comparing with other model and accuracy can be used to predict the existence of AQI_Range in AQI.

18. REFERENCE:

31