Air Quality Index Prediction

(Machine Learning Project Work)
(March-May 2021)

Swati Basu(06004092020), Hritika Verma(01204092020)
Sonam Prasad(05804092020), Sangeetha Panicker(04804092020)

ABSTRACT

Air, an essential natural resource, has been compromised in terms of quality by economic activities. Considerable research has been devoted to predicting instances of poor air quality, but most studies are limited by insufficient longitudinal data, making it difficult to account for seasonal and other factors. Several prediction models have been developed using 5-year-Indian-Dataset(2015-2020). Machine learning Classification methods, including Naive bayes' , Logistic Regression, K-Nearest-Neighbours, Decision Trees, Random Forest produce promising results for air quality index (AQI) level predictions. HyperParameter Tuning has been done using two methods including GridSearchCV and RandomizedSearchCV on every algorithm we used to make our model to predict Air Quality Index.

1 INTRODUCTION

Worldwide, air pollution is responsible for around 1.3 million deaths annually according to the World Health Organization (WHO). The depletion of air quality is just one of harmful effects due to pollutants released into the air. Other detrimental consequences, such as acid rain, global warming, aerosol formation, and photochemical smog, have also increased over the last several decades. The recent rapid spread of COVID-19 has prompted many researchers to investigate underlying pollution-related conditions contributing to COVID-19 pandemics in countries. Several shreds of evidence have shown that air pollution is linked to significantly higher COVID-19 death rates,and patterns in COVID-19 death rates mimic patterns in both high population density and high PM2.5 exposure areas. All the above mentioned raises an urgent need to anticipate and plan for pollution fluctuations to help communities and individuals better mitigate the negative impact of air pollution. To do so, air quality evaluation plays a significant role in monitoring and controlling air pollution. The Environmental Protection Agency (EPA) tracks the commonly known criteria pollutants, i.e., ground-level ozone (O_3), Sulphur dioxide (SO_2), particulates matter (PM10 andPM2.5), carbon monoxide (CO), carbon dioxide (CO_2), and nitrogen dioxide (NO_2). These substances are in compositions of a common index, called the Air Quality Index (AQI), indicating how clean or polluted the air is currently or forecasted to become in areas. As the AQI increases, a higher percentage of the population is exposed. Prediction models have been developed using Machine learning Classification methods, including Naive bayes' , Logistic Regression, K-Nearest-Neighbours, Decision Trees, Random Forest produce promising results for air quality index (AQI) level predictions. HyperParameter Tuning has been done using two methods including GridSearchCV and RandomizedSearchCV on every algorithm we used. This Project gives you an idea of fluctuations in pollution pre Covid and post Covid till 2020.

2 RELATED WORK

Researched similar papers on Air Quality Index Prediction and presented the findings in Appendix-A.

3 METHODOLOGY

3.1 Dataset Description

Air is what keeps humans alive. Monitoring it and understanding its quality is of immense importance to our well-being. The dataset contains air quality data and AQI (Air Quality Index) of 5 years i.e. 2015-2020 at hourly and daily level of various stations across multiple cities in India. A tutorial of how AQI is calculated is available here: https://www.kaggle.com/rohanrao/calculating-aqi-air-quality-index Cities included in this Dataset are : Ahmedabad, Aizawl, Amaravati, Amritsar, Bengaluru, Bhopal, Brajrajnagar, Chandigarh, Chennai, Coimbatore, Delhi, Ernakulam, Gurugram, Guwahati, Hyderabad, Jaipur, Jorapokhar, Kochi, Kolkata, Lucknow, Mumbai, Patna, Shillong, Talcher, Thiruvananthapuram, Visakhapatnam The data has been made publicly available by the Central Pollution Control Board: https://cpcb.nic.in/ which is the official portal of Government of India. Similar to air monitoring data, a dataset on noise decibel levels in India is available here: https://www.kaggle.com/rohanrao/noise-monitoring-data-in-india

Below is the screenshot of dataset obtained by using python code.

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29526	Visakhapatnam	2020-06-27	15.02	50.94	7.68	25.06	19.54	12.47	0.47	8.55	23.30	2.24	12.07	0.73	41.0	Good
29527	Visakhapatnam	2020-06-28	24.38	74.09	3.42	26.06	16.53	11.99	0.52	12.72	30.14	0.74	2.21	0.38	70.0	Satisfactory
29528	Visakhapatnam	2020-06-29	22.91	65.73	3.45	29.53	18.33	10.71	0.48	8.42	30.96	0.01	0.01	0.00	68.0	Satisfactory
29529	Visakhapatnam	2020-06-30	16.64	49.97	4.05	29.26	18.80	10.03	0.52	9.84	28.30	0.00	0.00	0.00	54.0	Satisfactory
29530	Visakhapatnam	2020-07-01	15.00	66.00	0.40	26.85	14.05	5.20	0.59	2.10	17.05	NaN	NaN	NaN	50.0	Good

29531 rows × 16 columns

Figure 1: Dataset

```
[ ] # no. of rows and columns
    df.shape

    (29531, 16)

[ ] df.describe()
```

	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
count	24933.000000	18391.000000	25949.000000	25946.000000	25346.000000	19203.000000	27472.000000	25677.000000	25509.000000	23908.000000	21490.000000	11422.000000	24850.000000
mean	67.450578	118.127103	17.574730	28.560659	32.309123	23.483476	2.248598	14.531977	34.491430	3.280840	8.700972	3.070128	166.463581
std	64.661449	90.605110	22.785846	24.474746	31.646011	25.684275	6.962884	18.133775	21.694928	15.811136	19.969164	6.323247	140.696585
min	0.040000	0.010000	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.010000	0.000000	0.000000	0.000000	13.000000
25%	28.820000	56.255000	5.630000	11.750000	12.820000	8.580000	0.510000	5.670000	18.860000	0.120000	0.600000	0.140000	81.000000
50%	48.570000	95.680000	9.890000	21.690000	23.520000	15.850000	0.890000	9.160000	30.840000	1.070000	2.970000	0.980000	118.000000
75%	80.590000	149.745000	19.950000	37.620000	40.127500	30.020000	1.450000	15.220000	45.570000	3.080000	9.150000	3.350000	208.000000
max	949.990000	1000.000000	390.680000	362.210000	467.630000	352.890000	175.810000	193.860000	257.730000	455.030000	454.850000	170.370000	2049.000000

Figure 2: Dataset Description

```
[ ] df.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 29531 entries, 0 to 29530
    Data columns (total 16 columns):
     #   Column      Non-Null Count  Dtype
    ---  ------      --------------  -----
     0   City        29531 non-null  object
     1   Date        29531 non-null  object
     2   PM2.5       24933 non-null  float64
     3   PM10        18391 non-null  float64
     4   NO          25949 non-null  float64
     5   NO2         25946 non-null  float64
     6   NOx         25346 non-null  float64
     7   NH3         19203 non-null  float64
     8   CO          27472 non-null  float64
     9   SO2         25677 non-null  float64
     10  O3          25509 non-null  float64
     11  Benzene     23908 non-null  float64
     12  Toluene     21490 non-null  float64
     13  Xylene      11422 non-null  float64
     14  AQI         24850 non-null  float64
     15  AQI_Bucket  24850 non-null  object
    dtypes: float64(13), object(3)
    memory usage: 3.6+ MB
```

Figure 3: Dataset Info

3.2 Data Preprocessing

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

3.2.1 *Handling Missing Values.*

Table 4 shows the amount of data missing in each attributes. The yellow part shows all the missing values.

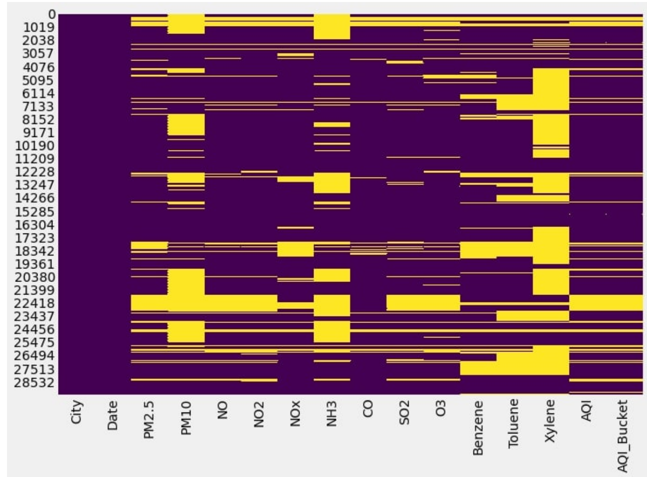


Figure 4: Missing Values in the dataset

The total amount of data missed in each attribute are :

Table 1: Total Count of Missing Values

Attribute	Count
City	0
Date	0
PM2.5	4598
PM10	11140
NO	3582
NO2	3585
NOx	4185
NH3	10328
CO	2059
SO2	3854
O3	4022
Benzene	5623
Toluene	8041
Xylene	18109
AQI	4681
AQI_Bucket	4681

- **Feature variable :** The feature variables are all numeric and since the quatity or amount of each constituents depends on an daily basis, so missing values in these attributes are handle by ***Linear Interpolation*** method.
  (Linear interpolation is an imputation technique that assumes a linear relationship between data points and utilises non-missing values from adjacent data points to compute a value for a missing data point.)
- **Target Variable :** The target variable is a categorical variable compromises of 6 different classes,i.e.,*Moderate,Satisfactory,Poor,Very*

Poor,Severe,Good. These missing values are handled by ***frequency count method*** with respect to each city. A city having maximum count of AQI_Bucket value will be inserted for the NaN values for that city in AQI_Bucket. For eg, suppose Ahmedabad has maximum count of say 'Severe', for the city Ahmedabad, NaN values for AQI_Bucket will be replaced with 'Severe'.

Table 2 shows the total count values of target variable after handling all missing values.

Table 2: Caption

Target Variable	Count
Moderate	10868
Satisfactory	10074
Poor	2791
Very Poor	2337
Severe	2013
Good	1448

3.2.2 Checking for Outliers. The outliers are there in the feature variable, so we'll just ignore it for now because temperature variations could also be possible

3.3 Data Visualization

Data visualization is the discipline of trying to understand data by placing it in a visual context so that patterns, trends and correlations that might not otherwise be detected can be exposed.

3.3.1 Yearly Average Pollution Data for Metropolitan Cities and Big Cities. The constituents of air are analysed in each year from 2015 to 2020 of major citites like *Ahmedabad, Delhi, Chennai and Kolkata* where air pollution is the topic of concern.

Fig 8 describes the yearly average pollution data in Ahmedabad.

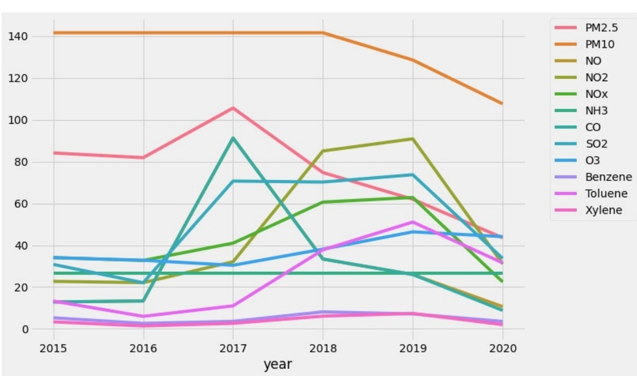


Figure 5: Ahmedabad Yearly Average Pollution data

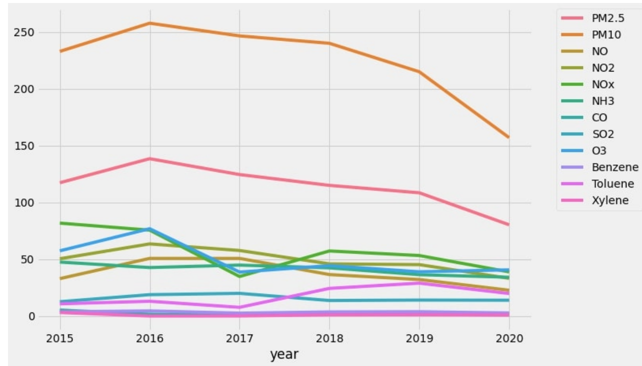


Figure 6: Delhi Yearly Average Pollution data

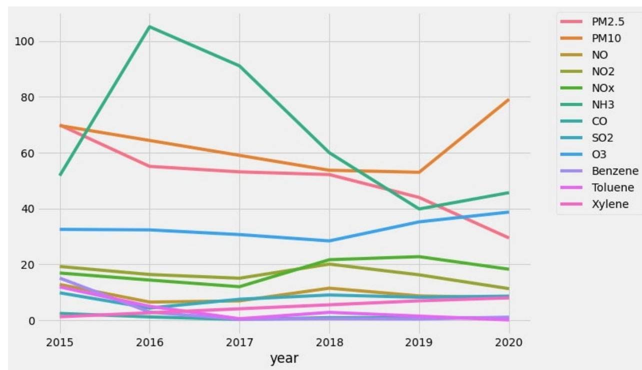


Figure 7: Chennai Yearly Average Pollution data

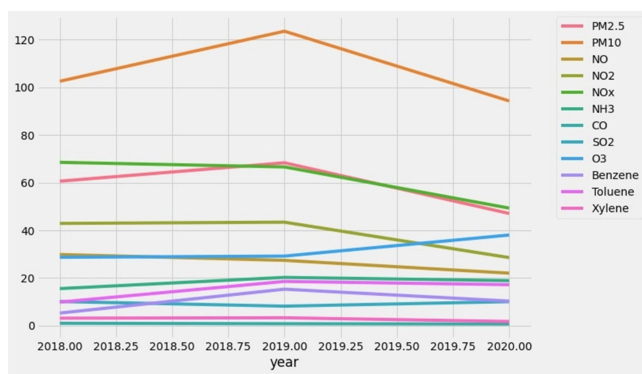


Figure 8: Kolkata Yearly Average Pollution data

OBSERVATION :

- Each City Data shows different plot for each pollutants PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2,O3, Benzene, Toluene and Xylene.
- In an Average Secenario, Yearly Mean of each pollutants shows that From Year 2017 Pollution has increased and reduced from 2019 to 2020.(due to COVID 19 Pendemic!!)

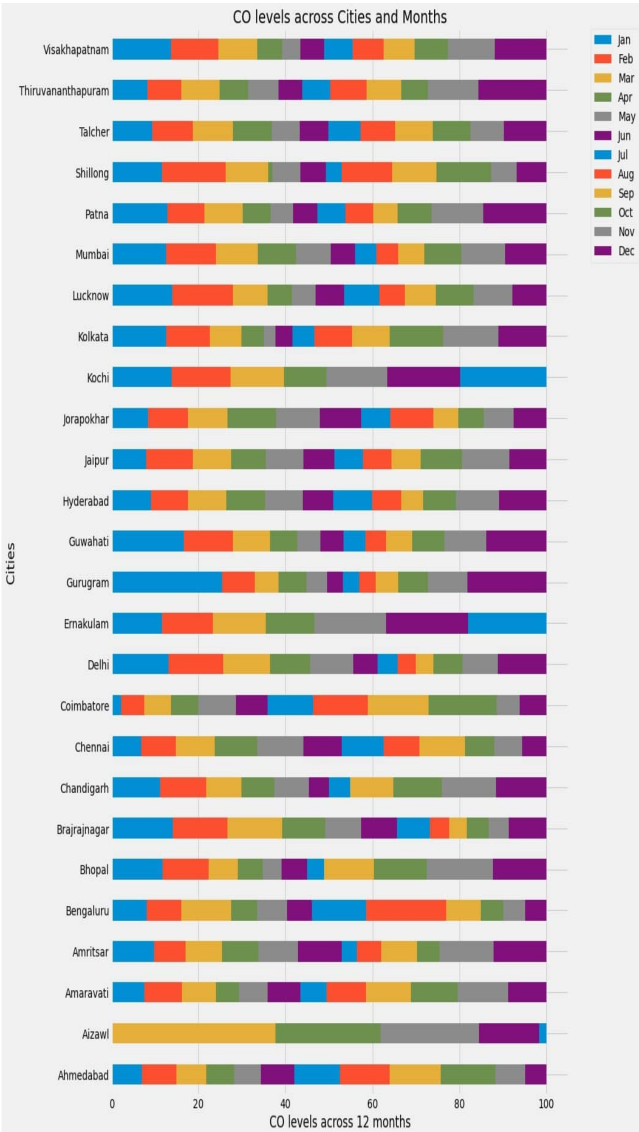


Figure 9: CO levels across Cities and Months

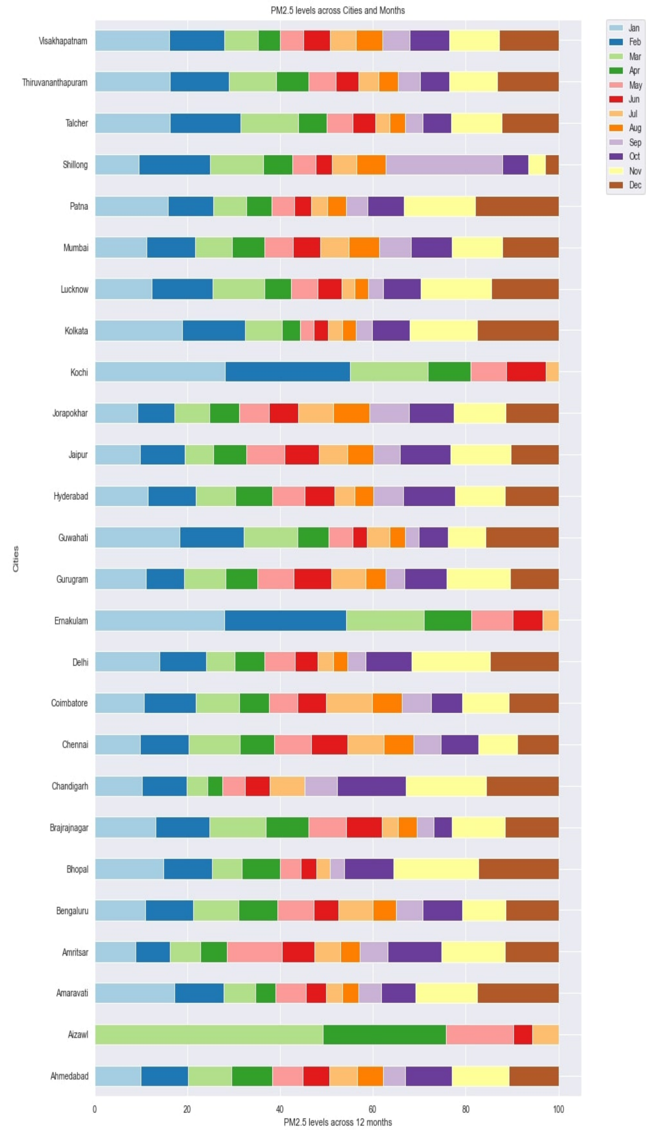


Figure 10: PM2.5 levels across Cities and Months

3.3.2 CO Levels across Cities and Months. : Fig 9 shows the CO levels across cities and months.
OBSERVATION : For all given cities, CO level in winter season (December,January and February) is more compare to other months data.

3.3.3 PM2.5 levels across Cities and Months. : Fig 10 shows the PM2.5 levels across cities and months.
OBSERVATION : For all given cities, PM2.5 level in winter season (December,January and February) is also more compare to other months data.

3.3.4 AQI Bucket and Cities. Fig 11 shows the relationship between AQI_Bucket and Cities.

3.3.5 AQ Acceptability and Cities: listed by Acceptable AQ. : Fig 12 shows the relationship between AQ Acceptability and Cities: listed by Acceptable AQ.
OBSERVATION : AQ Acceptability and cities shows that Metro(Major) cities like Delhi,Mumbai,Ahmedabad,Chennai are under Unacceptable category of Air Quality !!!

3.3.6 Weekday vs Weekend Polution. Fig 13 - 18 shows the Pollution day-wise data for big cities.
OBSERVATION : Above Weekdays vs Weekend Pollution data graphs shows that generally pollutant readings decreases in weekend (specially PM10).

3.3.7 Pre Corona [2016 to 2020]. : Here I divide the data set into two part namely Vehicular Pollution content (PM2.5, PM10, NO2,

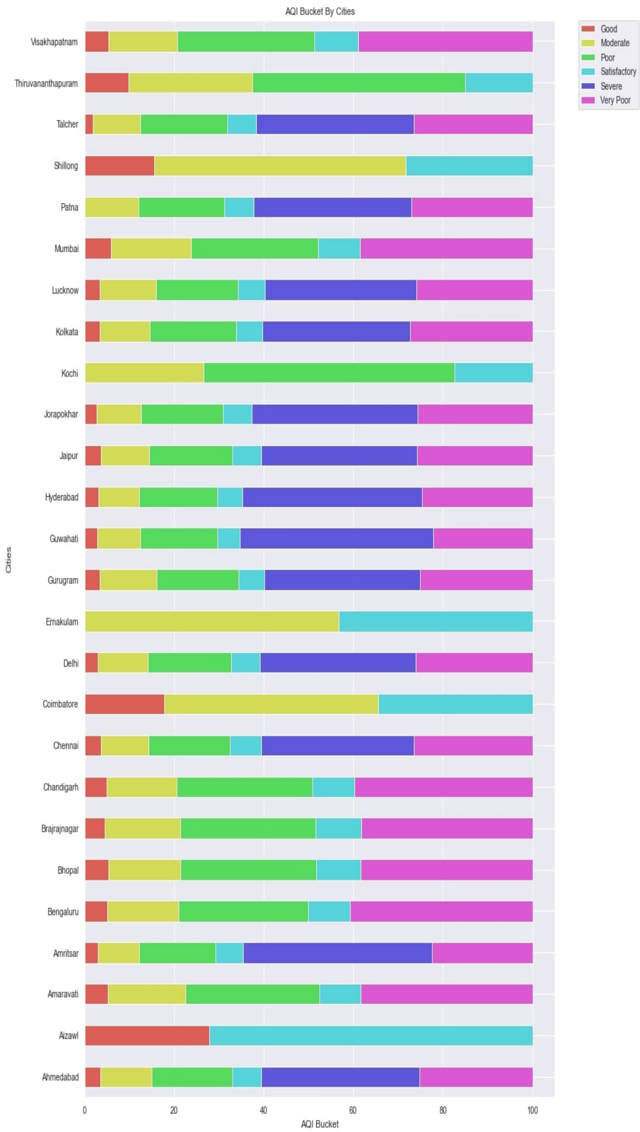


Figure 11: AQI Bucket and Cities

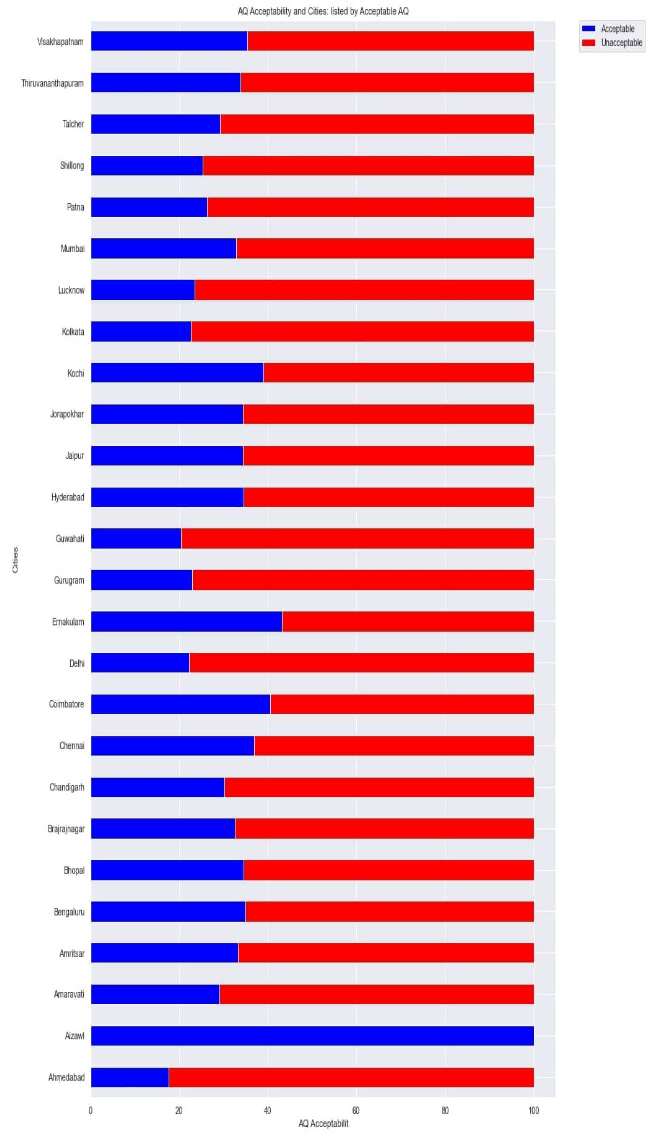


Figure 12: AQ Acceptability and Cities: listed by Acceptable AQ

NH3, CO) and Industrial Pollution content (CO, SO2, O3, Benzene, Toluene, Xylene) and find how these contents correlated with AQI (air quality index)

- **Vehicular Pollution Content** : PM2.5, PM10, NO2, NH3, CO
- **Industrial Pollution Content** : CO, SO2, O3, Benzene, Toluene, Xylene

Fig 19 shows the Vehicular Pollution Content. Fig 20 shows the Industrial Pollution Content. Fig 21 shows the Satisfactory Levels for Big Cities.

OBSERVATION :

- Hence we can observe that in pre-corona stage. the highest Vehicular Pollution is in the City Delhi and then Ahemdabad and lowest or minimum in Shillong.

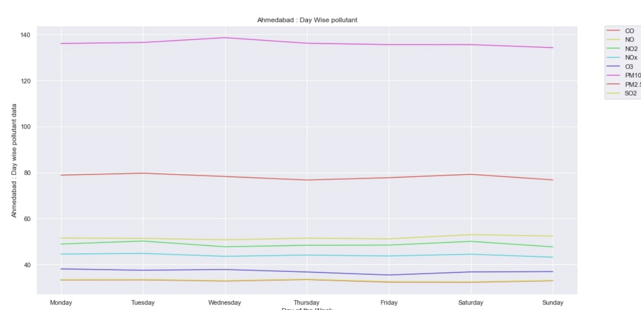


Figure 13: Ahemdabad Pollution day-wise

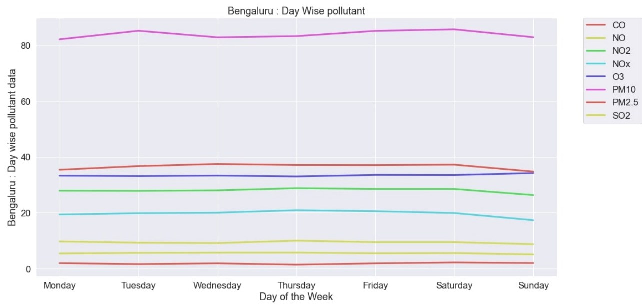


Figure 14: Banglore Pollution day-wise

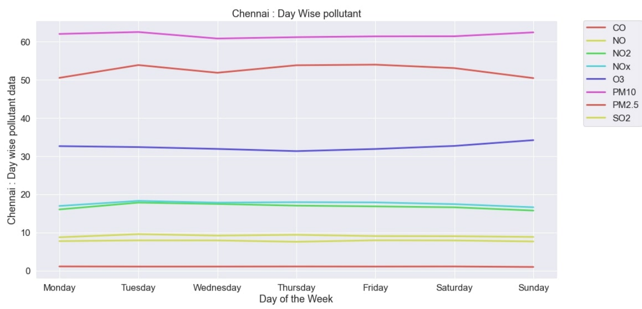


Figure 15: Chennai Pollution day-wise

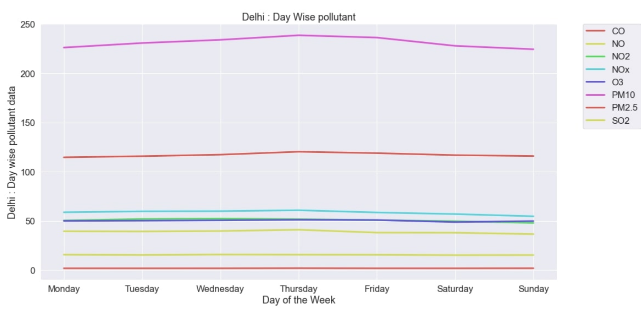


Figure 16: Delhi Pollution day-wise

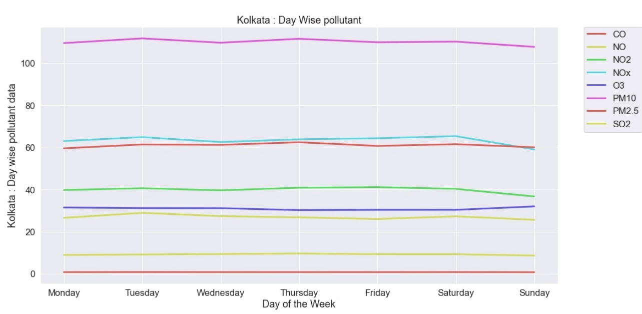


Figure 17: Kolkata Pollution day-wise

- Similarly we can observe that highest Industrial Pollution content is in Delhi, Ahemdabad and Mumbai and lowest in Brajrajnagar.

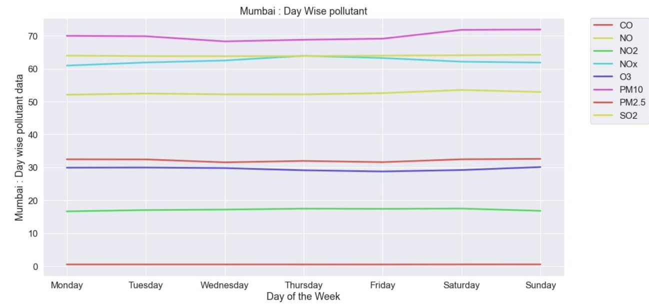


Figure 18: Mumbai Pollution day-wise

- So the overall satisfactory level quite high for cities like Mumbai, Chennai, Kolkata and Bengaluru but most severe for Delhi and Ahemdabad.

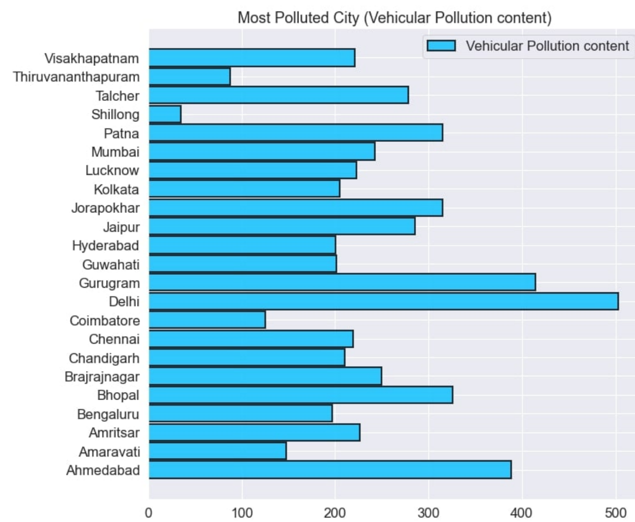


Figure 19: Vehicular Pollution Content

3.3.8 Post Corona [2020 >]. Fig 22 shows the Vehicular Pollution Content. Fig 23 shows the Industrial Pollution Content. Fig 24 shows the Satisfactory Levels for Big Cities.

OBSERVATION :

- Here we can observe that in post-corona stage. the highest Vehicular Pollution is in the City Brajrajnagar and then Patna but less in Delhi and ahemdabad as compared to pre-corona stage. The lowest or minimum is still Shillong.
- Similarly we can observe that highest Industrial Pollution content is in Ahemdabad but less in Delhi and Mumbai.
- So the overall satisfactory level quite high for cities like Mumbai, Chennai and Bengaluru but moderate for Delhi and Ahemdabad.Here the poor and severe levels are not much in any cities.

3.3.9 Cities and the Proportion of Pollution in each of them. : Fig 25 shows the complete overall distribution of Pollution in each city.

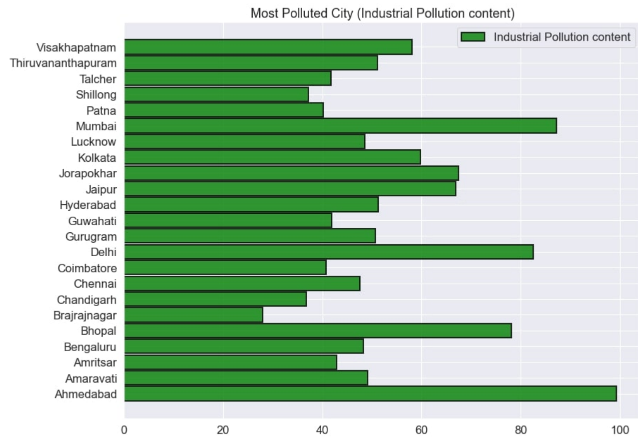


Figure 20: Industrial Pollution Content

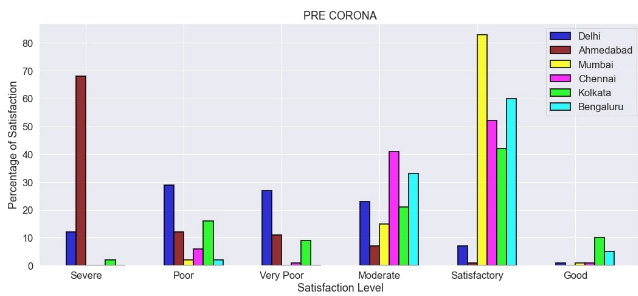


Figure 21: Satisfactory Levels for Big Cities

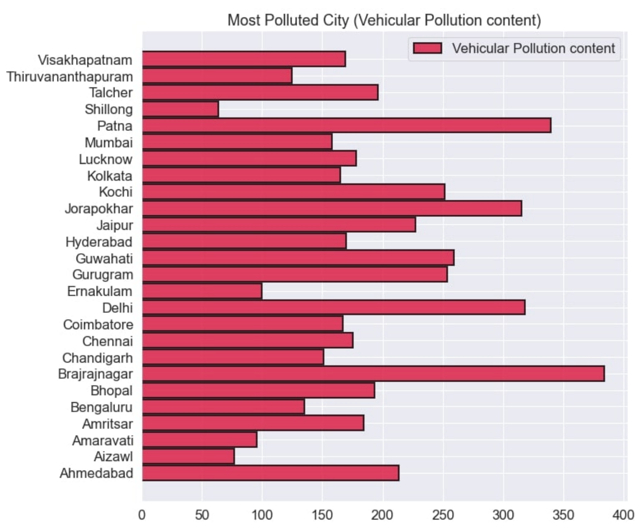


Figure 22: Vehicular Pollution Content

3.3.10 Histogram for AQI. Fig 26 shows the histogram for AQI.
OBSERVATION : Maximum frequency occurs in the interval between 0-250

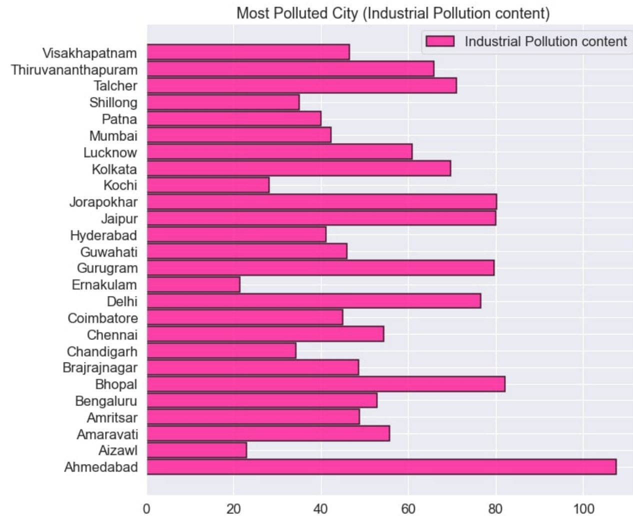


Figure 23: Industrial Pollution Content

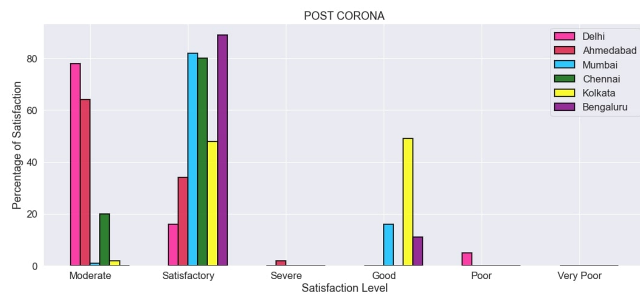


Figure 24: Satisfactory Levels for Big Cities

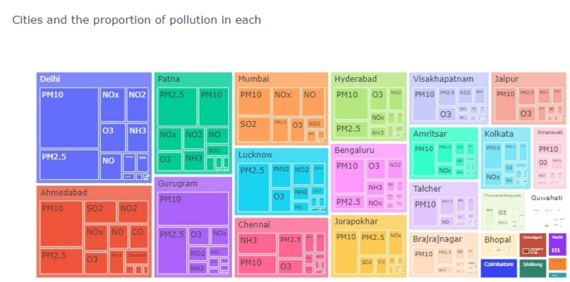


Figure 25: Cities and the Proportion of Pollution in each of them

3.3.11 HeatMap showing correlations within all the feature variables. Fig 27 shows the correlations within all the feature variables in the form of heatmap.
OBSERVATION : Here We can observe that the correlation coefficient between the 'PM2.5' and 'AQI' is positively correlated. It is a positive correlation. This depicts that for a positive increase in

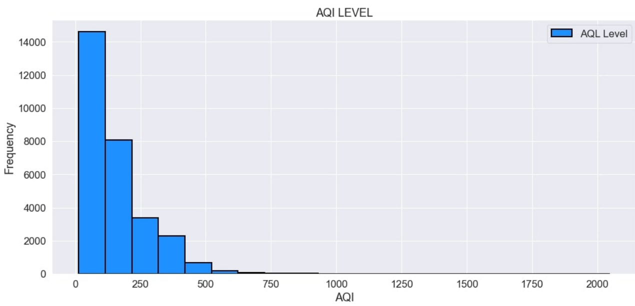


Figure 26: Histogram for AQI

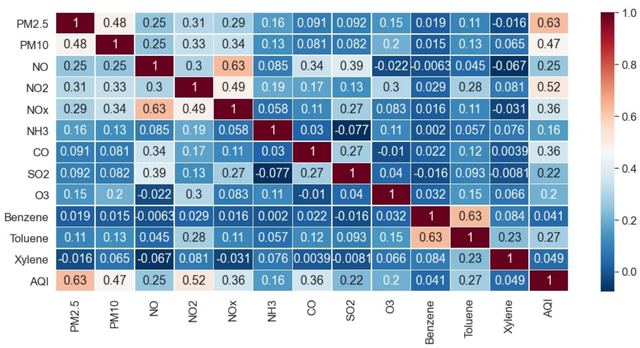


Figure 27: HeatMap showing correlations within all the feature variables

the variable 'PM2.5', there is also a positive increase in the second variable 'AQI'.

(The correlation coefficient is a statistical measure of the strength of the relationship between the relative movements of two variables. The values range between -1.0 and 1.0. A calculated number greater than 1.0 or less than -1.0 means that there was an error in the correlation measurement. A correlation of -1.0 shows a perfect negative correlation, while a correlation of 1.0 shows a perfect positive correlation. A correlation of 0.0 shows no linear relationship between the movement of the two variables.)

3.3.12 Scatter Plot. Fig 28 shows the scatter plot between AQI and PM2.5

OBSERVATION : Here we can observe that the y variable('AQI') tends to increase as the x variable('PM2.5') increases, and also so there is a positive correlation between these two variables and we have calculated the strength of correlation above ,i.e., 0.63.

(A scatterplot shows the relationship between two quantitative variables measured for the same individuals.)

3.3.13 Pairplot. Fig 29 shows the complete pairplot between all the attributes.

3.4 Proposed Approach

After Data visualization we split the data into normal test train and 10 fold cross validation. We then had performed undersampling and oversampling on both the split-ed dataset.In both the cases

Figure 28: Scatter Plot between AQI and PM2.5

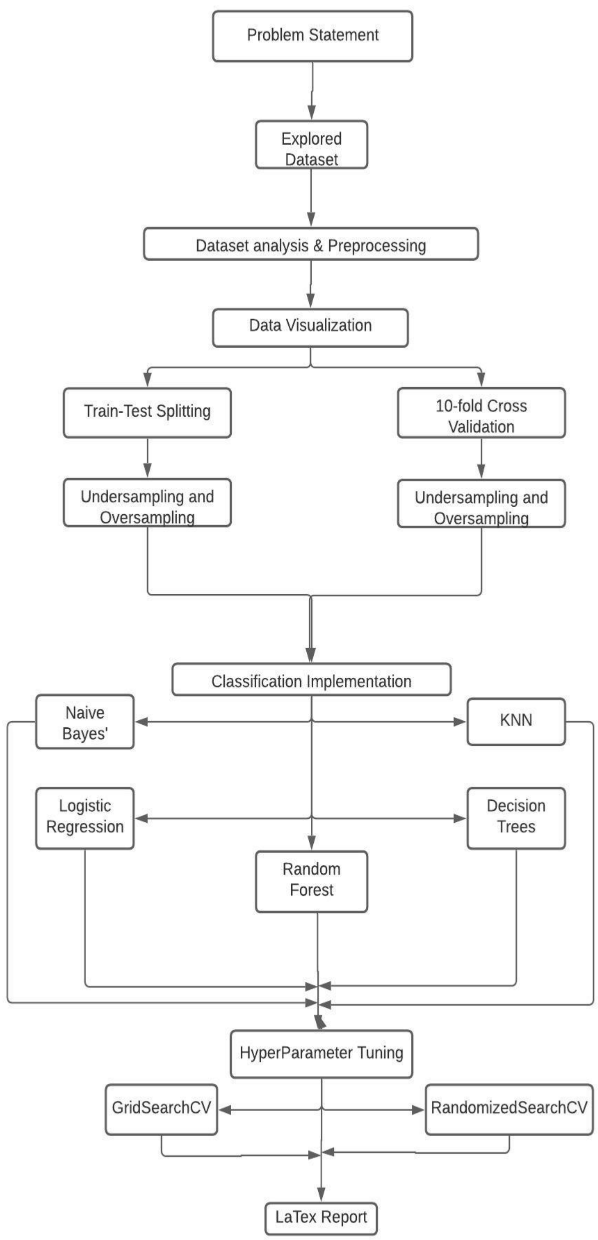
Figure 29: Pairplot

oversampling has maximum accuracy. The algorithm like KNN, Logistic Regression, Decision Tree, Naive Bayes, Random Forest were implemented on the both splinted data-set. At last we performed hyper tuning to find the best parameters using Grid search and Random search.

4 EXPERIMENT DESIGN

Workflow Design is presented in figure 30

Figure 30: Workflow Diagram



4.1 Under and Oversampling

Undersampling: which consists in down-sizing the majority class by removing observations until the dataset is balanced.

Oversampling: which consists in over-sizing the minority class by adding observations.

Oversampling and undersampling in data analysis are techniques used to adjust the class distribution of a data set (i.e. the ratio between the different classes/categories represented). These terms are used both in statistical sampling, survey design methodology and in machine learning.

Oversampling and undersampling are opposite and roughly equivalent techniques. There are also more complex oversampling techniques, including the creation of artificial data points with algorithms like Synthetic minority oversampling technique. In signal processing, oversampling is the process of sampling a signal at a sampling frequency significantly higher than the Nyquist rate. Theoretically, a bandwidth-limited signal can be perfectly reconstructed if sampled at the Nyquist rate or above it. The Nyquist rate is defined as twice the bandwidth of the signal. Oversampling is capable of improving resolution and signal-to-noise ratio, and can be helpful in avoiding aliasing and phase distortion by relaxing anti-aliasing filter performance requirements. In signal processing, undersampling or bandpass sampling is a technique where one samples a bandpass-filtered signal at a sample rate below its Nyquist rate (twice the upper cutoff frequency), but is still able to reconstruct the signal.

When one undersamples a bandpass signal, the samples are indistinguishable from the samples of a low-frequency alias of the high-frequency signal. Such sampling is also known as bandpass sampling, harmonic sampling, IF sampling, and direct IF-to-digital conversion.

Random Oversampling:

Random Oversampling includes selecting random examples from the minority class with replacement and supplementing the training data with multiple copies of this instance, hence it is possible that a single instance may be selected multiple times. Random Undersampling:

Random Undersampling is the opposite to Random Oversampling. This method seeks to randomly select and remove samples from the majority class, consequently reducing the number of examples in the majority class in the transformed data.

4.2 ALGORITHMS :

4.3 Logistic Classification :-

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1, with a sum of one. Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

Types of Logistic Classification: Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types

(a). Binary or Binomial: In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc. (b). Multinomial: In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

(c). Ordinal: In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

(d). Logistic Regression Assumptions: Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same.

(e). Classification Models : Binary Logistic Regression Model: The simplest form of logistic regression is binary or binomial logistic regression in which the target or dependent variable can have only 2 possible types either 1 or 0.

(f). Multinomial Logistic Regression Model: Another useful form of logistic regression is multinomial logistic regression in which the target or dependent variable can have 3 or more possible unordered types i.e. the types having no quantitative significance.

4.4 Naive Bayes :-

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

4.5 Decision Tree :-

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning. A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into sub trees.

4.6 Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result. The predictions from each tree must have very low correlations.

Random Forest is capable of performing both Classification and Regression tasks. It is capable of handling large datasets with high dimensionality. It enhances the accuracy of the model and prevents the overfitting issue.

4.7 k-Nearest-Neighbour(KNN)

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data. K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well:

Lazy learning algorithm: KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.

Non-parametric learning algorithm: KNN is also a non-parametric learning algorithm because it does not assume anything about the underlying data.

4.8 Hypertuning

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters. However, there is another kind of parameters, known as Hyperparameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Some examples of model hyperparameters include:

The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization The learning rate for training a neural network. The C and sigma hyperparameters for support vector machines. The k in k-nearest neighbors. they all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as Hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. Of course, you must select from a specific list of hyperparameters for a given model as it varies from model to model.

Often, we are not aware of optimal values for hyperparameters which would generate the best model output. So, what we tell the model is to explore and select the optimal model architecture automatically. This selection procedure for hyperparameter is known as Hyperparameter Tuning.

5 RESULTS AND OBSERVATIONS :

Baseline Comparison Chart has been done and added in the end of Appendix A.

5.1 NORMAL TEST/TRAIN SPLIT (75 TRAIN and 25 TEST) RATIO

- C1 - Good
- C2 - Moderate
- C3 - Poor
- C4 - Satisfactory
- C5 - Severe
- C6 - Very Poor

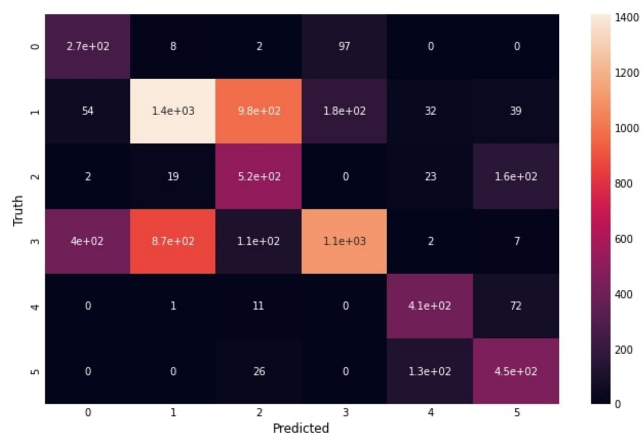


Figure 31: Confusion Matrix of Logistic Regression (Undersampling) - Normal Test/Train

5.2 K - FOLD CROSS VALIDATION (10 FOLD)

5.3 Hypertuning Results

5.3.1 Logistic Classification. : Performed hypertuning and found hypertuned parameters.

- max_iter: 500
- solver: newton-cg
- penalty: l2
- C: 10
- Best Score : 90.16%

Table 3: Normal Test/Train Split - UNDERSAMPLING

ML Model	PRECISION %						RECALL %						AVG. Precision %	AVG. Recall %	Accuracy
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6			
Logistic Regression	65.25	53.09	30.40	86.45	72.11	59.21	73.31	57.79	68.59	39.28	82.15	75.60	61.09	66.12	56.27
Random Forest Classifier	94.36	46.42	34.08	99.02	87.10	73.94	98.87	60.22	78.58	23.94	98.03	90.83	72.49	75.08	56.35
KNN CLassifier	67.47	55.52	37.40	95.06	79.68	70.84	86.96	65.58	75.94	35.80	96.56	85.80	67.66	74.44	61.43
Decision Tree	88.23	42.60	28.85	98.55	64.02	91.91	96.51	47.71	78.80	26.96	98.68	93.03	69.03	73.61	53.14
Naive Bayes Classifier	50.89	52.69	38.87	74.76	33.26	70.37	33.72	58.65	71.5	24.81	86.86	78.03	53.47	58.9	

Table 4: Normal Test/Train Split - OVERSAMPLING

ML Model	Precision %						Recall %						Avg. Precision %	Avg. Recall %	Accuracy
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6			
Logistic Regression	58.12	87.93	62.28	88.52	80.99	60.16	99.43	78.99	78.14	80.35	66.86	77.85	73	80.27	79.43
Random Forest Classifier	95.40	98.01	94.81	98.13	98.42	96.62	99.15	96.68	97.97	97.71	97.84	98.96	96.90	98.05	98
KNN CLassifier	76.24	94.63	80.00	93.65	92.40	86.94	95.46	89.51	91.54	91.19	92.93	90.09	87.31	91.79	90.84
Decision Tree	95.95	95.32	96.86	97.24	96.79	97.16	95.44	97.02	94.73	96.44	96.61	95.64	96.56	95.98	96.39
Naives Bayes Classifier	36.26	85.44	56.35	76.65	74.46	80.37	92.60	71.18	81.99	67.99	78.79	84.10	68.22	80	

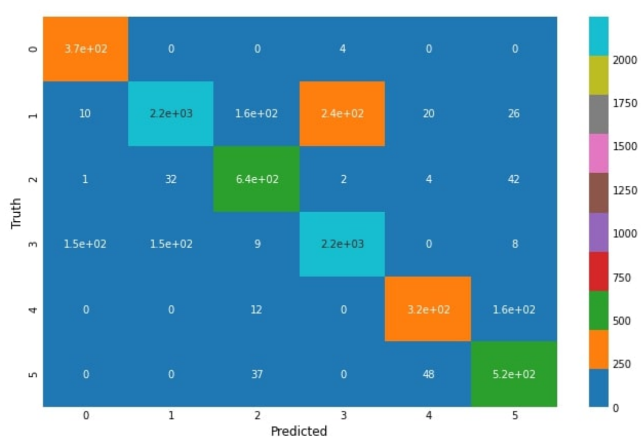


Figure 32: Confusion Matrix of Random Forest Classification (Undersampling) - Normal Test/Train

Figure 33: Confusion Matrix of KNN Classification (Undersampling) - Normal Test/Train

ML Model	Precision						Recall						Average Precision	Average Recall	Accuracy
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6			
Logistic Regression	23.55	35.09	16.20	39.05	90.90	03.10	93.22	14.04	55.08	48.89	25.89	03.57	37.19	40.11	33.61
	28.09	50.67	10.98	58.43	01.92	35.33	91.53	70.74	59.22	29.32	14.28	98.07	42.30	60.53	51.33
	26.12	62.75	45.60	70.44	04.76	17.76	62.06	72.53	69.68	45.36	16.66	98.11	47.84	60.73	61.90
	10.97	36.86	36.92	72.15	17.39	36.01	100	50.37	75.81	55.50	52.11	86.17	53.78	69.99	63.35
	34.47	58.84	32.67	45.54	26.00	52.86	89.05	51.26	79.48	47.96	87.40	85.47	65.03	73.43	63.83
	21.25	45.66	16.49	49.86	42.85	43.75	63.44	37.95	88.41	52.27	50.0	82.35	52.56	62.40	47.20
	12.85	60.11	44.11	33.51	19.63	45.56	59.28	68.07	63.14	20.12	77.0	37.24	46.42	54.14	53.77
	00.08	36.16	44.19	70.65	37.87	66.03	20.0	51.24	66.66	06.73	46.34	96.12	50.49	47.85	28.91
	07.37	73.54	32.79	28.57	40.0	49.53	18.69	25.94	74.03	25.97	14.28	92.16	41.05	41.85	39.38
	55.39	36.52	14.62	57.59	0	50.0	85..71	43.50	64.86	52.39	0	90.38	41.51	56.14	51.91
Random Forest Classifier	42.40	16.09	21.93	43.64	100	59.06	99.43	20.58	35.78	55.16	50.87	50.89	47.19	52.12	48.88
	95.71	40.18	21.84	100	50.00	68.49	99.19	70.74	87.37	24.46	100	96.15	62.70	79.65	50.66
	96.62	54.19	31.36	98.87	85.71	43.80	98.85	67.74	85.82	23.21	100	100	68.43	79.27	53.77
	95.89	32.44	61.45	100	86.23	85.00	98.59	62.18	82.31	70.70	100	93.08	76.84	75.64	58.24
	89.54	56.93	41.96	88.88	80.39	76.07	100	48.73	79.27	22.65	96.85	97..86	72.30	74.22	60.81
	99.31	59.40	21.70	98.36	88.88	64.00	100	64.84	94.51	29.64	100	94.11	71.94	80.51	56.41
	94.55	62.16	35.58	95.51	61.72	76.50	99.28	57.87	82.85	22.88	100	92.06	71.00	75.82	59.87
	100	14.58	49.02	100	82.00	91.08	100	55.08	90.07	08.36	100	97.18	72.78	75.11	33.86
	88.50	68.74	28.47	79.52	78.40	79.06	93.90	46.37	81.41	26.23	100	89.63	70.45	72.92	61.32
	88.34	56.68	26.42	97.83	66.66	60.00	100	80.31	71.17	45.31	100	98.07	65.99	82.47	64.95
KNN Classifier	36.13	22.42	17.93	30.99	99.81	13.97	99.43	29.05	34.73	43.17	40.44	05.80	67.78	79.19	65.97
	66.33	48.79	25.85	91.77	36.84	62.82	81.04	79.69	80.58	32.64	100	94.23	78.36	90.68	88.38
	34.82	64.12	33.97	96.37	35.29	36.69	89.65	68.23	83.07	37.51	100	96.22	75.17	83.45	83.30
	24.25	42.01	57.93	95.88	69.44	74.03	80.28	61.73	71.95	28.99	93.89	83.15	83.00	86.30	87.19
	52.38	66.22	41.23	78.74	53.60	68.49	88.32	50	69.87	28.84	93.70	88.24	80.56	86.08	83.64
	89.72	66.16	26.17	95.68	34.78	41.17	90.34	73.24	88.41	39.42	100	82.35	88.83	91.91	92.48
	52.86	65.85	33.98	88.23	48.52	69.12	85.71	54.92	74	29.95	99	84.13	75.14	85.09	80.79
	73.68	17.44	49.07	99.74	62.12	86.09	93.33	58.15	84.12	21.52	100	91.54	79.03	89.92	86.08
	74.90	75.00	32.49	56.50	55.23	76.73	81.30	51.30	78.20	29.35	96.93	79.03	75.84	81.59	77.78
	58.84	63.70	28.91	94.82	33.33	57.14	89.56	83.04	64.86	51.18	100	100	83.36	90.46	86.31
Decision Tree	43.89	13.66	28.97	42..25	100.0	57.14	9943	2687	2175	5940	4386	4821	47.65	49.57	45.46
	85.66	34.86	32.43	97.42	01.71	83.60	93.95	51.20	81.55	27.65	100.0	98.07	55.95	72.26	44.26
	72.80	57.75	41.25	97.75	02.37	69.33	95.40	68.73	82.67	26.91	100.0	98.11	56.87	75.83	47.37
	75.38	33.87	86.32	89.95	48.96	96.30	69.01	62.78	93.69	18.85	100.0	95.68	71.80	74.20	60.95
	66.84	59.07	58.54	79.50	21.93	88.21	91.24	45.20	73.93	22.65	100.0	99.14	62.35	71.63	57.22
	98.56	53.63	44.04	98.39	01.22	100.0	94.48	50.03	90.24	30.23	100.0	100.0	65.97	81.93	57.67
	86.07	62.58	57.11	93.33	18.34	90.39	97.14	52.81	95.14	23.65	100.0	94.13	67.97	74.29	57.77
	93.33	23.56	56.49	99.34	05.60	97.93	93.33	59.88	98.41	08.26	100.0	100.0	62.71	76.86	34.67
	68.33	57.29	31.06	31.25	30.24	91.07	16.66	27.89	83.33	23.37	100.0	94.00	51.54	57.09	46.39
	84.90	53.23	46.00	98.69	00.67	92.85	98.90	68.73	88.28	43.37	100.0	100.0	62.72	81.69	60.78
Naive Bayes Classifier	42.10	14.48	63.63	50.84	81.77	68.96	68.36	27.11	7.36	66.23	70.06	0.89	43.27	40.00	51.9
	54.18	40.31	22.22	73.81	32.11	62.59	49.59	66.50	67.96	19.41	100.0	96.15	42.60	66.66	41.75
	21.33	52.62	26.50	81.74	15.38	39.25	18.39	55.84	79.9	24.53	33.33	100.0	37.16	52.00	45.54
	58.46	30.59	55.85	89.10	52.18	75.31	53.52	55.15	71.74	13.33	84.03	89.63	60.25	61.23	50.55
	46.45	61.51	45.50	53.33	33.69	76.50	33.57	49.91	72.43	19.64	98.42	92.52	52.84	61.03	55.77
	59.37	60.04	22.16	72.66	42.10	61.90	13.10	59.58	85.97	33.83	100.0	76.47	46.72	61.55	50.32
	47.51	56.03	38.44	56.41	11.28	82.98	47.85	41.13	52.28	13.51	100.0	70.66	48.61	54.24	41.58
	83.33	28.07	55.11	91.36	37.07	88.47	13.33	54.31	83.33	6.90	100.0	91.90	45.84	59.29	31.29
	76.73	74.79	33.33	43.71	21.54	78.67	49.18	33.11	70.33	20.77	95.40	85.02	54.02	59.05	48.59
	70.73	54.58	25.78	80.98	14.48	63.29	15.93	73.86	66.66	44.38	100.00	96.15	49.31	66.16	56.48

ML Model	Precision						Recall						Average Precision	Average Recall	Accuracy
	C1	C2	C3	C4	C5	C6	C1	C2	C3	C4	C5	C6			
Logistic Regression	52.55	27.34	15.33	10.20	10.0	20.83	98.87	69.97	23.50	20.66	00.38	02.23	37.70	35.93	22.10
	53.37	58.64	23.52	94.38	12.5	70.68	95.56	78.05	69.90	45.81	57.14	78.84	52.18	70.88	62.78
	41.14	72.82	49.47	63.13	11.11	23.91	98.85	59.43	74.40	58.34	33.33	62.26	43.60	64.43	61.46
	53.48	90.80	47.69	90.48	04.34	42.39	96.18	79.67	88.41	91.96	00.93	19.87	54.87	63.00	70.84
	53.51	74.35	60.84	72.75	49.35	69.16	100.0	72.97	80.34	35.92	59.84	70.94	63.33	70.00	67.42
	51.97	82.69	57.87	60.89	10.0	56.25	100.0	67.20	89.63	65.71	75.0	52.94	68.28	75.08	69.48
	58.29	79.83	52.73	32.29	28.93	68.96	97.85	69.62	77.14	53.30	68.00	41.37	58.51	67.88	65.42
	46.87	84.06	80.62	97.57	40.0	73.78	100.0	88.09	61.11	96.30	39.02	85.21	70.48	78.29	90.01
	69.29	83.75	33.19	55.10	50.0	56.71	96.34	59.42	50.96	63.11	10.71	87.55	58.01	61.35	62.98
	44.03	75.20	67.66	82.36	14.28	75.51	99.45	81.80	81.08	61.20	50.0	71.15	59.84	74.11	72.90
Random Forest Classifier	78.57	72.32	51.35	99.57	100	48.27	99.43	100	100	87.08	50.72	100	75.01	89.54	75.69
	95.73	95.58	87.82	100	58.33	85.24	99.59	100	98.05	94.41	100	100	87.12	98.67	97.05
	97.75	96.37	73.62	95.15	85.14	76.47	100	91.69	100	91.79	100	98.11	87.51	96.93	92.82
	95.83	99.70	99.79	99.23	100	100	97.18	100	97.96	99.80	100	100	99.09	99.15	99.52
	89.54	98.43	91.55	88.92	87.76	91.55	100	84.51	99.57	98.05	96.06	99.57	91.29	96.29	93.09
	99.31	100	100	97.40	100	100	100	98.25	100	100	100	100	99.45	99.70	99.05
	95.89	97.84	91.84	81.11	72.79	93.54	100	83.12	99.71	96.31	99	100	88.83	96.35	90.99
	100	94.52	100	100	97.61	99.30	100	99.42	99.60	98.42	100	100	98.57	99.57	98.88
	91.66	98.03	71.23	73.74	97.02	92.34	76.01	79.63	100	99.22	100	100	87.34	92.47	88.38
	87.92	100	92.5	95.48	100	94.54	100	93.46	100	98.42	50.0	100	95.07	90.31	96.54
KNN Classifier	68.37	73.75	39.16	94.02	100	32.54	97.74	89.83	95.08	78.41	41.66	73.66	67.97	79.39	66.07
	78.27	88.16	63.44	94.82	70	77.19	91.53	93.16	89.32	85.30	100	84.61	78.65	90.65	88.75
	69.64	91.28	61.66	84.56	66.66	57.35	89.65	80.35	90.55	85.61	33.33	73.58	71.86	75.51	83.30
	59.09	86.39	86.74	92.86	91.12	83.87	91.54	87.29	90.44	88.42	72.30	88.76	83.34	86.46	87.47
	72.31	91.74	79.08	79.86	78.4	81.76	93.43	78.61	88.88	82.83	77.16	91.02	80.53	85.32	83.64
	72.91	95.29	87.64	92.73	100	82.35	96.55	94.52	95.12	88.33	100	82.35	88.49	92.81	92.48
	65.48	92.30	76.08	74.41	67.54	78.22	92.14	77.56	90	78.18	77	94.13	75.67	84.83	81.47
	50	60.89	81.31	98.53	90.24	91.44	93.33	86.37	88.09	83.96	90.24	97.88	78.73	89.98	86.21
	89.13	91.04	56.17	58.86	76.07	82.18	83.33	70	90.38	72.46	81.12	90.32	75.57	81.27	77.31
	51.32	94.50	81.51	88.86	100	85.45	95.60	86.76	87.38	84.53	50	90.38	83.61	82.44	86.31
Decision Tree	77.51	31.57	65.86	93.78	100.0	41.95	91.52	96.85	85.96	86.34	15.00	76.78	68.44	75.41	55.65
	65.06	89.23	89.81	97.70	53.84	96.22	85.48	99.71	94.17	90.62	100.0	98.07	86.98	94.67	93.66
	97.67	92.71	75.46	93.24	31.57	72.41	96.55	90.56	95.66	88.96	100.0	79.24	77.18	91.83	90.38
	94.23	62.71	99.67	91.94	90.95	100.0	69.01	98.80	61.99	93.87	94.36	54.64	89.91	78.78	82.96
	92.85	89.72	91.64	77.20	86.15	92.88	94.89	84.51	89.10	88.14	88.18	92.09	88.41	89.48	87.77
	95.17	98.28	100.0	94.81	88.88	100.0	95.17	96.54	90.24	99.40	100.0	76.47	96.19	92.95	96.95
	95.62	91.90	91.66	69.54	58.08	93.93	93.57	82.98	75.42	90.16	79.0	85.51	83.46	84.44	84.28
	100.0	62.40	100.0	98.39	96.55	99.23	100.0	94.26	94.04	86.73	68.29	90.84	92.76	89.09	88.96
	88.00	91.83	77.83	50.83	93.71	92.47	26.82	78.26	90.06	95.32	98.97	87.78	82.45	79.54	80.22
	87.63	96.64	92.24	93.35	66.66	96.15	89.56	92.88	96.39	97.78	100.0	48.07	88.81	87.45	94.34
Naive Bayes Classifier	51.05	82.65	54.86	83.33	82.90	21.71	96.04	87.01	73.84	61.80	78.67	14.73	62.75	68.68	71.90
	034.29	76.48	58.71	89.99	12.27	76.78	91.53	89.22	62.13	51.39	100.0	82.69	58.16	79.49	69.11
	31.73	81.94	54.85	69.88	1.56	47.36	7.86	67.74	57.87	72.90	16.66	50.94	47.88	56.99	68.70
	31.92	64.11	83.70	85.01	89.56	77.71	74.64	76.38	91.86	65.16	48.35	89.63	72.00	74.34	75.00
	49.80	84.80	71.52	61.15	60.22	83.05	97.08	65.06	90.17	59.64	79.52	85.47	68.94	79.49	73.34
	35.91	93.63	66.50	76.63	70.05	88.23	99.31	82.38	84.14	67.19	87.50	88.23	71.52	84.79	78.15
	52.17	89.04	63.68	68.31	17.87	81.41	95.00	58.79	66.2	71.58	79.00	75.57	62.31	74.36	66.54
	18.75	37.79	75.63	80.92	44.47	93.43	100.0	63.14	82.93	35.16	73.17	86.61	52.03	73.97	50.25
	68.12	88.76	51.58	33.91	42.08	74.94	88.61	45.65	89.42	35.16	78.17	84.10	59.87	71.05	61.63
	27.71	91.06	67.08	67.40	20.00	87.17	99.45	62.36	95.49	61.27	50.00	65.38	60.07	72.32	65.42

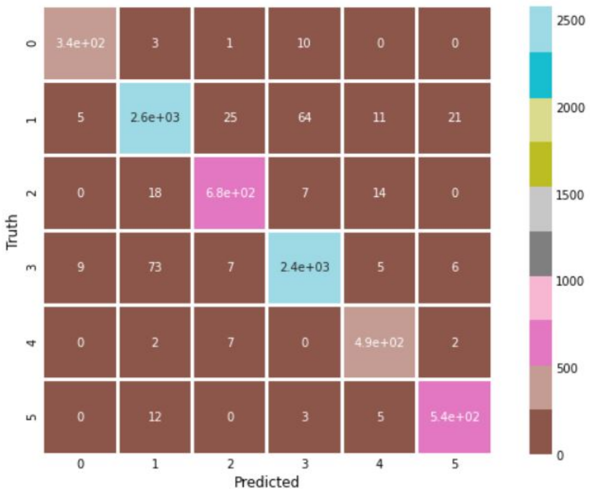


Figure 34: Confusion Matrix of Decision Tree (Undersampling) - Normal Test/Train

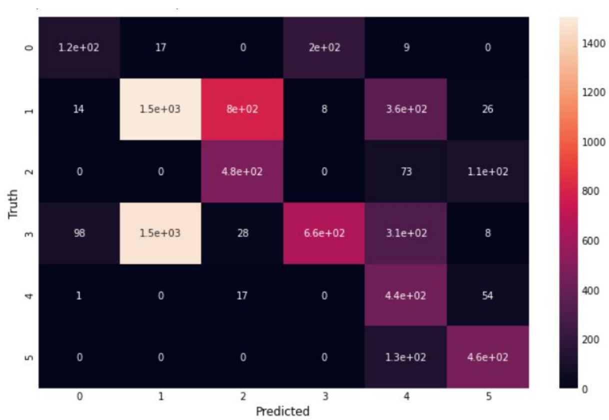


Figure 35: Confusion Matrix of Naive Bayes (Oversampling) - Normal Test/Train

5.3.2 *Random Forest Classification.* : Performed hypertuning and found hypertuned parameters.

- Best n estimators: 336
- Best max_features: sqrt
- Best max_depth: 50
- Best criterion: entropy
- Best Score: 99.20%

5.3.3 *Decesion Tree.* : Performed Randomise-Search and found hypertuned parameters.

- criterion: gini
- max_depth: None
- max_leaf_nodes: None
- min_samples leaf : 5
- min_samples split: 2
- Best Score: 98.53%

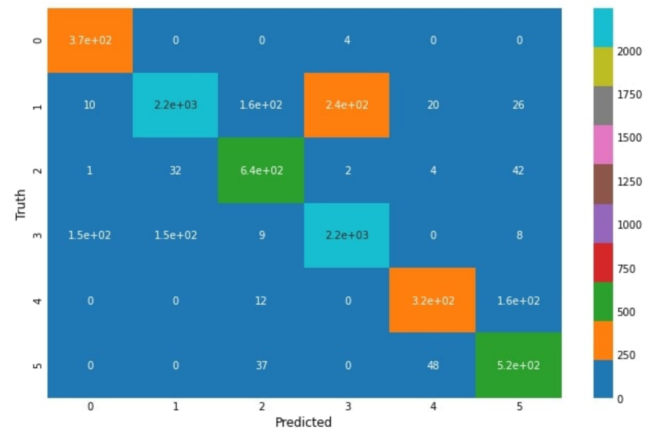


Figure 36: Confusion Matrix of Logistic Regression (Oversampling) - Normal Test/Train

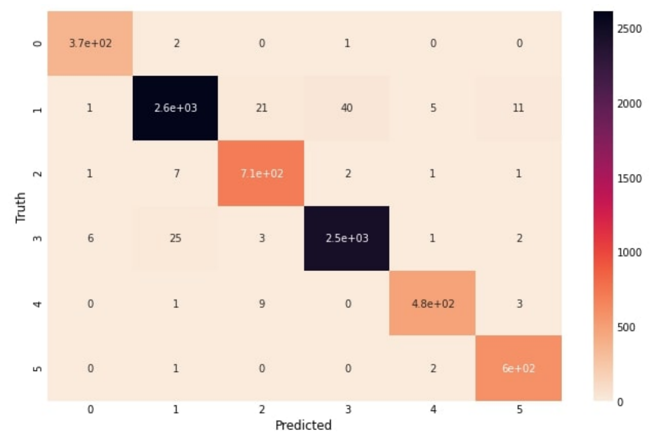


Figure 37: Confusion Matrix of Random Forest Classification (Oversampling) - Normal Test/Train

5.3.4 *Decesion Tree.* : Performed Grid-Search and found hypertuned parameters.

- criterion: entropy
- max_depth: None
- max_leaf_nodes: None
- min_samples leaf : 1
- min_samples split: 2
- Best Score: 99.01%

5.3.5 *KNN Classification.* : Performed Grid-Search(GS) and Randomized-Search(RS) and found hypertuned parameters.

- Metric : minkowski
- Best leaf_size(GS) : 1
- Best p(GS) : 1
- Best n_neighbours(GS) : 7
- Best Score(GS) : 91.64%
- Best leaf_size(RS) : 10
- Best p(RS) : 1

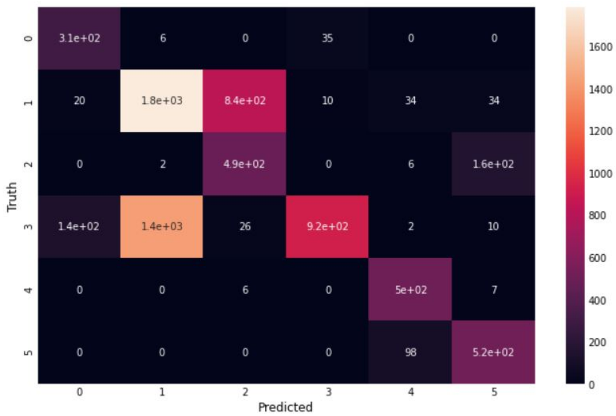


Figure 38: Confusion Matrix of KNN Classification (Oversampling) - Normal Test/Train

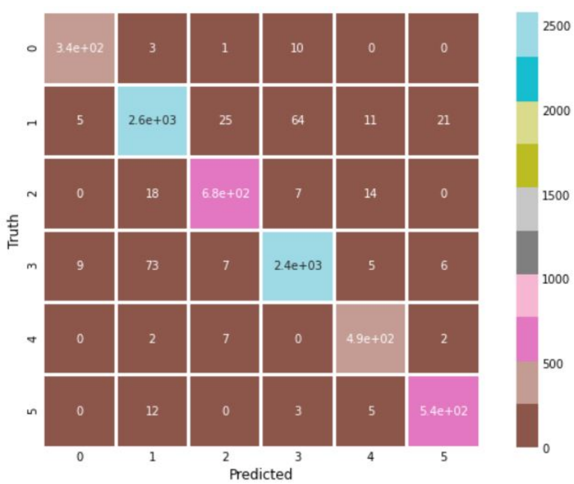


Figure 39: Confusion Matrix of Decision Tree (Oversampling) - Normal Test/Train

- Best n_neighbours(RS) : 1
- Best Score(GS) : 97.37%

6 FUTURE WORK AND CONCLUSION

Machine learning Classification methods, including Naive bayes' , Logistic Regression, K-Nearest-Neighbours, Decision Trees, Random Forest produce promising results for air quality index (AQI) level predictions. HyperParameter Tuning has been done using two methods including GridSearchCV and RandomizedSearchCV on every algorithm we used to make our model to predict Air Quality Index. We performed all this after performing Undersampling and Oversampling on dataset for get more accurate result and we found that in each algorithm we applied, **Oversampling of Data** gives more accurate results on multiple classification algorithms along with Better Performance Rate.We also applied K-fold (k=10) Cross

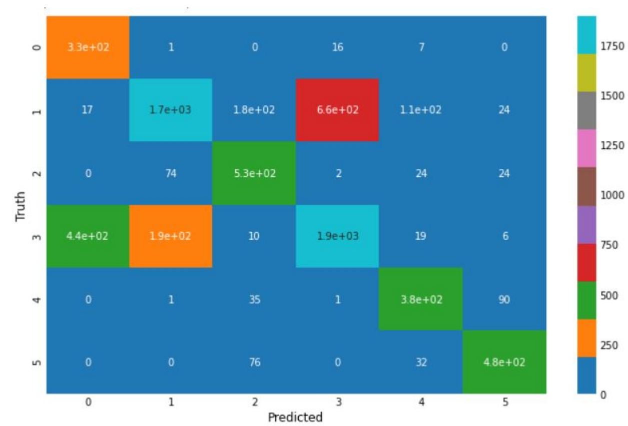


Figure 40: Confusion Matrix of Naive Bayes (Oversampling) - Normal Test/Train

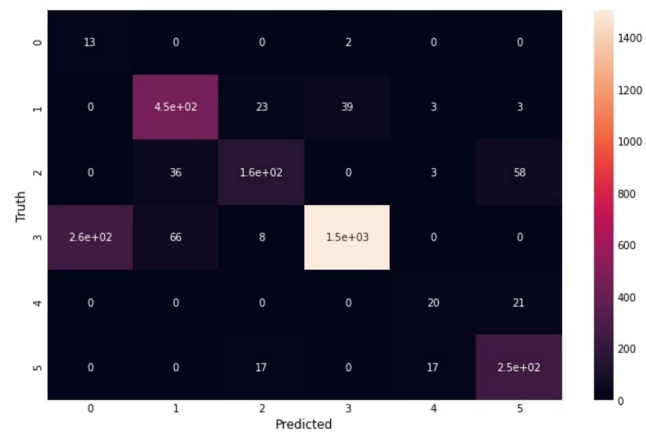


Figure 41: Confusion Matrix for highest accuracy of Logistic Classification done using 10_Kfold (Oversampling)

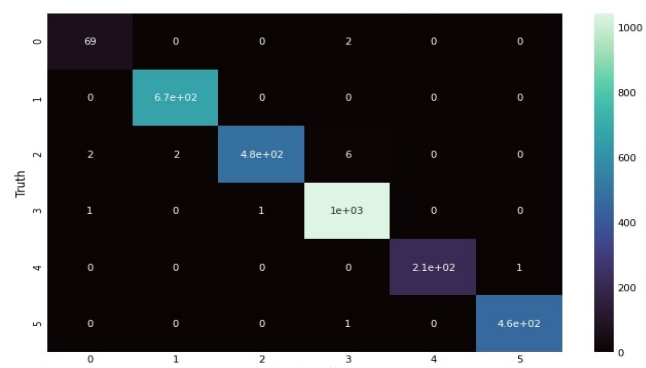


Figure 42: Confusion Matrix for highest accuracy of Rnadom Forest Classification done using 10_Kfold (Oversampling)

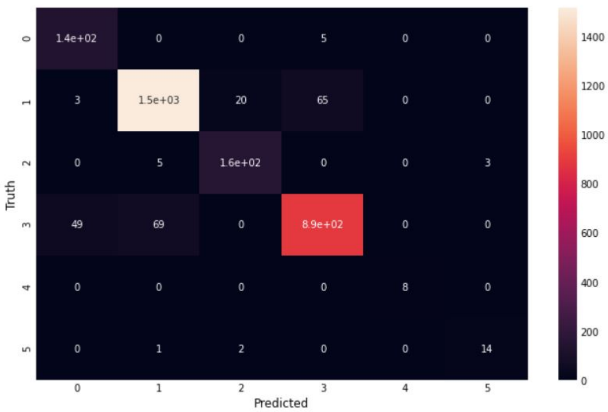


Figure 43: Confusion Matrix for highest accuracy of KNN done using 10_Kfold (Oversampling)

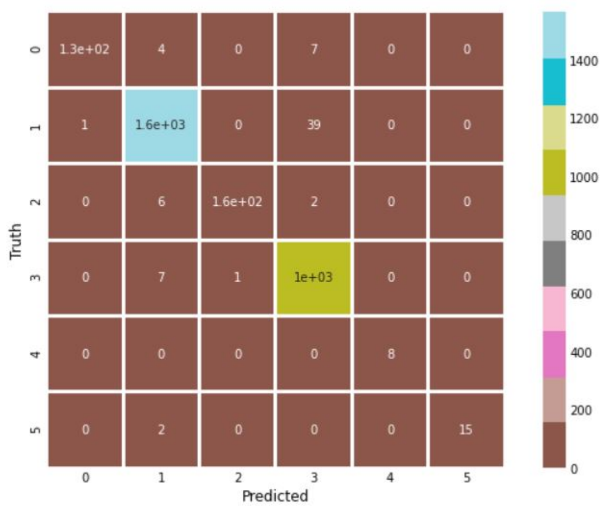


Figure 44: Confusion Matrix for highest accuracy of Decision Tree done using 10_Kfold (Oversampling)

validation to check classification results. As **future work**, we intend to improve and investigate the usage of Neural Networks in predicting Air Quality Index to get more accuracy. Also, we will try to inculcate other boosting algorithms to achieve the results. In this project we have just performed classification algorithms to predict the quality of air but we can also apply regression algorithms to find out the air quality index.

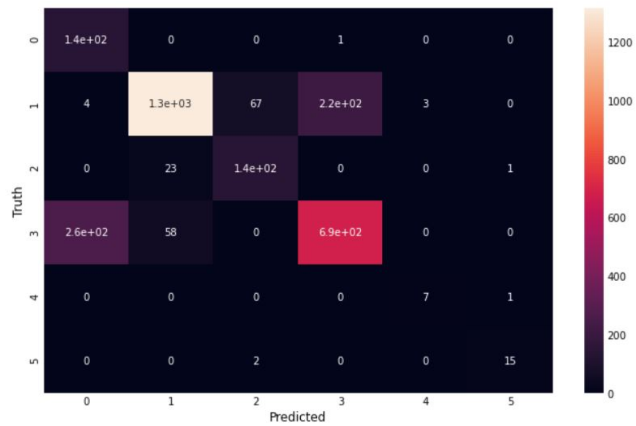


Figure 45: Confusion Matrix for highest accuracy of Naive Bayes Classification done using 10_Kfold (Oversampling)

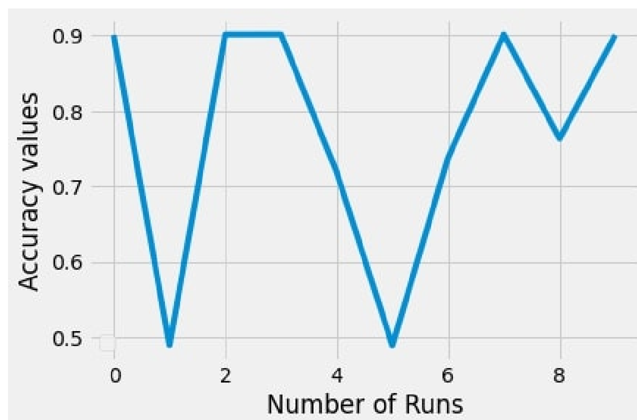


Figure 46: Logistic Classification - Accuracy values vs Number of Runs

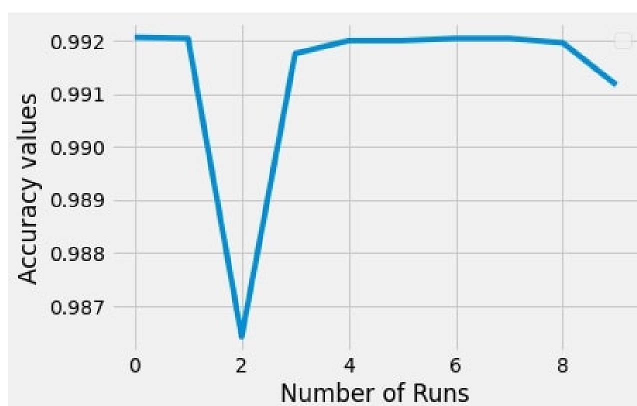


Figure 47: Random Forest - Accuracy values vs Number of Runs

REFERENCES

[1] Burhan BARAN. "AIR QUALITY INDEX PREDICTION IN BESIKTAS DISTRICT BY ARTIFICIAL NEURAL NETWORKS AND K NEAREST NEIGHBORS". In: *Mühendislik Bilimleri ve Tasarım Dergisi* 9.1 (2021), pp. 52–63.

[2] José Juan Carbajal-Hernández et al. "Assessment and prediction of air quality using fuzzy logic and autoregressive models". In: *Atmospheric Environment* 60 (2012), pp. 37–50.

[3] Fabio Cassano et al. "A Recurrent Neural Network Approach to Improve the Air Quality Index Prediction". In: *International Symposium on Ambient Intelligence*. Springer. 2019, pp. 36–44.

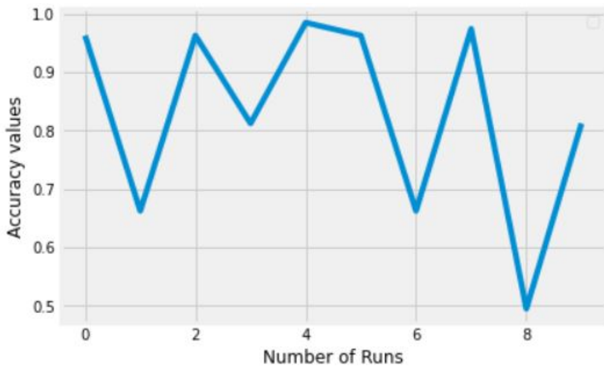


Figure 48: Decision Tree - Accuracy values vs Number of Runs

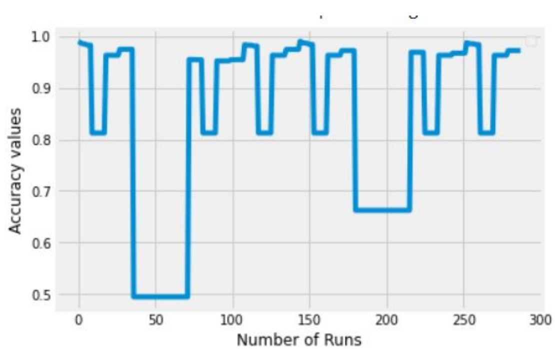


Figure 49: Decision Tree - Accuracy values vs Number of Runs

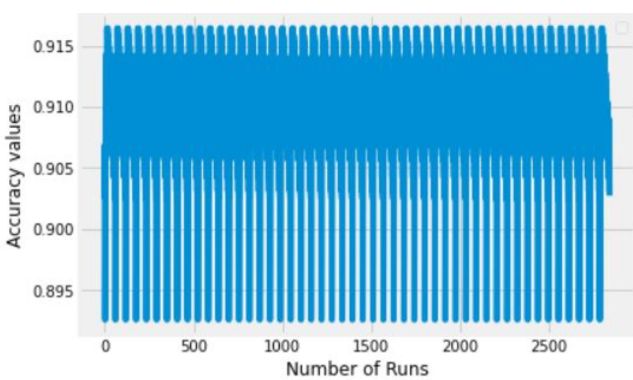


Figure 50: KNN Classification - Accuracy values vs Number of Runs for Grid Search

[4] J Dhilipan, DB Shanmgam, and Rikhitha Manoj Kumar. "AIR QUALITY ANALYSIS BEFORE AND DURING COVID-19 LOCKDOWN". In: ().

[5] Elia Georgiana Dragomir. "Air quality index prediction using K-nearest neighbor technique". In: *Bulletin of PG University of Ploiesti, Series Mathematics, Informatics, Physics, LXII* 1.2010 (2010), pp. 103–108.

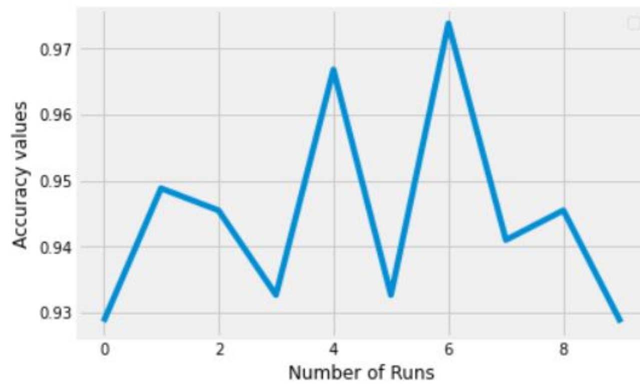


Figure 51: KNN Classification - Accuracy values vs Number of Runs for Randomized Search

[6] Gaganjot Kaur Kang et al. "Air quality prediction: Big data and machine learning approaches". In: *International Journal of Environmental Science and Development* 9.1 (2018), pp. 8–16.

[7] Ibrahim Kök, Mehmet Ulvi Şimşek, and Suat Özdemir. "A deep learning model for air quality prediction in smart cities". In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE. 2017, pp. 1983–1990.

[8] Huixiang Liu et al. "Air quality index and air pollutant concentration prediction based on machine learning algorithms". In: *Applied Sciences* 9.19 (2019), p. 4069.

[9] Jasleen Sethi and Mamta Mittal. "Ambient air quality estimation using supervised learning techniques". In: *EAI Endorsed Transactions on Scalable Information Systems* 6.22 (2019).

[10] A Gnana Soundari, J Gnana Jeslin, and AC Akshaya. "INDIAN AIR QUALITY PREDICTION AND ANALYSIS USING MACHINE LEARNING". In: *International Journal of Applied Engineering Research* 14.11 (2019).

[11] Dongwen Zhang, Qi Zhao, and Yunfeng Xu. *Prediction of Air Quality Index Based on LSTM Model: A Case Study on Delhi and Houston.* Tech. rep. EasyChair, 2019.

Table 5: Comparative analysis between prior research works

SNo.	Paper Title	Paper Year	Approach	Results
1	AIR QUALITY ANALYSIS BEFORE AND DURING COVID-19 LOCKDOWN Dhilipan et al. [4]	December 2020	The air quality examples have been moved into two phases: the pre-lockdown stage (1 March to 24 March 2020) and the post-lockdown stage (25 March to 15 April 2020). The null values are replaced using the accurate method (calculating mean using group by series, month, year and replacing null values by it) to increase accuracy. Then the BTX and Particulate matter and it is updated in the data. Leftover nan or null values with '0' to detect the fault in cities for recording pollutants level.Analysis for Particulate_Matter factor 2015 to 2020 using Bar Graphs for different cities.	Lockdown has affected the particulate matter level of cities such that it has decreased in most of the cities significantly and making the air more suitable to breathe.Outcomes show an articulated decrease in air poisons during lockdown particularly in Delhi and Kolkata; these two urban communities are known to be exceptionally contaminated urban areas in India and on the planet. The outcomes will draw in the consideration of the Indian Government to consider the most proficient method to carefully limit vehicular and modern contamination to improve air quality which will assist with supporting better general wellbeing in India.
2	INDIAN AIR QUALITY PREDICTION AND ANALYSIS USING MACHINE LEARNING [10]	2019	Data is collected from cpcb.nic.in website and developed a model to predict the air quality index based on historical data of previous years and predicting over a particular upcoming year as aGradient descent boosted multivariable regression problem. Outliers are removed from the data by boundary value analysis (BVA) and Box Plot. Using the Naïve Forecast approach, the dataset is splitted into two parts of first 75% and rest 25% data into test and train datasets to identify the huge seasonal variations and trend. Resampling the data month wise is done to remove the seasonal trends and applied linear regression to fit the data. Also used AHP MCDM technique to find of order of preference by similarity to ideal solution.	Model is capable of predicting the current data with 95% accuracy. It will successfully predict the upcoming air quality index of any particular data within a given region. With this model one can forecast the AQI and alert the respected region of the country. Also it a progressive learning model it is capable of tracing back to the particular location needed attention provided the time series data of every possible region needed attention.

Comparative analysis between prior research works Table 5

SNo.	Paper Title	Paper Year	Approach	Results
3	Air Quality Index Prediction using K-Nearest Neighbor Technique [5]	2010	The training data is gathered by one of the monitoring station from the national air quality-monitoring network located in Ploiesti. The data used in this application were recorded in June 2009. The daily mean values of these parameters are used in order to establish the quality index for each pollutant. Only the data recorded in 29 days of June 2009 are available for this experiment. This is a drawback because the training set should have had more instances in order to create a model more accurate and precise. The experiment has been made using Weka (Waikato Environment for Knowledge Analysis), a data mining specialized software. Used 10-fold cross-validation.	If the actual and the predicted values are equal, then the error is zero. Otherwise, it is displayed the error value: as a negative number if the predicted value of the air quality index is smaller than the actual one, or as a positive number if the prediction gives a value greater than the actual value. The results were relatively good, if we consider that for 19 of the 29 instances the prediction error was zero. The accuracy of the model can be improved by taking into consideration a longer period of time for the model's training set. Among the parameters that have been selected for this experiment, there is a strong correlation, and, therefore, these can be used in the forecasting process.
4	Ambient Air Quality Estimation using Supervised Learning Techniques [9]	July 2019	The air quality dataset of Faridabad from CPCB website has been used. Computation of AQI - Calculation of Sub- Index, Formation of AQI Noise/missing data has been ignored. Metrics used for AQI Prediction - Precision, Recall, Accuracy, Error rate, F1 score and ROC curve, correlation coefficient, coefficient of determination, min max accuracy and mean absolute percentage error. To predict the AQI, three classification techniques namely Decision tree, Naïve Bayes and SVM and three ensemble techniques namely Random Forest, Voting and Stacking have been used.	It has been observed that ensemble techniques outperform in the ensemble category and Stacking ensemble has highest accuracy and F1 score and lowest error rate. Decision Trees show highest accuracy and error rate compared to all other classification techniques. In case of regression techniques, SVR has the highest value of min max accuracy and least value of MAPE. Preventing burning of garbage in residential areas and using natural gas rather than coal in power plants are some of the measures that could be used to improve the air quality.
5	Air quality prediction:big data and machine learning approaches [6]	January 2018	Big data and machine learning techniques has been used in air quality forecasting. Quality of air has been evaluated using artificial intelligence techniques. Big data model is used to predict the level of ground level ozone. Forecast the reading of an air quality monitoring station for 0-48 hours using a data driven method. Enable the determination of the regional source of air pollution using sensor and satellite data.	With the advancement of IoT infrastructures, big data technologies, and machine learning techniques, real-time air quality monitor and evaluation is desirable for future smart cities. Paper reports the recent literature study, reviews and compares current research work on air quality evaluation based on big data analytics, machine learning models and techniques. It highlights some observations on future research issues, challenges, and needs.

Comparative analysis between prior research works Table 5

SNo.	Paper Title	Paper Year	Approach	Results
6	Air Quality Index and Air Pollutant Concentration Prediction Based on Machine Learning Algorithms [8]	September 2019	2 Datasets - Beijing Air Quality Dataset (December 2013 to August 2018), is from the Beijing Municipal Environmental Monitoring Center(has 1738 instances), and air quality recording that contains the responses of a gas multi-sensor device deployed on a field in an Italian city. The dataset contains 9358 instances of hourly averaged responses from an array of five metal oxide chemical sensors embedded in an air quality chemical multi-sensor device. Data were recorded from March 2004 to February 2005. Used support vector regression (SVR) and random forest regression (RFR) to build regression models for predicting the Air Quality Index (AQI) in Beijing and the nitrogen oxides (NOX) concentration in an Italian city. The root-mean-square error (RMSE), correlation coefficient (r), and coefficient of determination (R 2) were used to evaluate the performance of the regression models. HeatMap- correlation coefficients for air pollution indicators of (a) Beijing and (b) an Italy city.	Experimental results showed that the SVR-based model performed better in the prediction of the AQI (RMSE = 7.666, R 2 = 0.9776, and r = 0.9887), and the RFR-based model performed better in the prediction of the NOX concentration (RMSE = 83.6716, R 2 = 0.8401, and r = 0.9180). With the increasing number of samples, the time complexity of the SVR model increased cubically. Therefore, the SVR model is not suitable for processing a large number of samples. This study established two prediction models based on different prediction scenarios, which improved the prediction accuracy of air indicators and provides guidance for modeling and analyzing urban air quality.
7	A Recurrent Neural Network Approach to Improve the Air Quality Index Prediction [3]	June 2019	Using two different Recurrent Neural Network models, they have performed two tests to prove that it is possible to predict the level of the pollutants in a specific area by using the data coming from the surrounding area. By using this approach on both the weather and air stations on the territory it is possible to have alerts many days ahead on the pollution levels. To avoid the overfitting and the random weight initialization problem, we have repeated the training 20 times randomly choosing the training and testing data.	Two different types of experiments have been conducted: the former using as test some random data from the datasets, the latter letting the RNN to blindly predict the behaviour of a specific area knowing the behaviour of the neighbour one. Results shows that the RNN is able to predict with a very high accuracy the CAQI level of random days while the "blind prediction" results are promising.

Comparative analysis between prior research works Table 5

SNo.	Paper Title	Paper Year	Approach	Results
8	Prediction of air quality index based on Lstm [11]	2019	uses data provided by the environmental protection department to predict Air Quality Index (AQI) through temperature, PM2.5, PM10, SO 2 , wind direction, NO 2 , CO and O3 paper proposes a prediction model of environmental quality based on Long Short Term Memory (LSTM). Introduced the background, technical characteristics, development status and problems of air environment monitoring. It will introduced the environmental prediction model. AQI prediction by using LSTM and analyze the error of the prediction results.	Recent advances in the development of deep learning models have led to a rapid increase in their application in academic and industrial settings. the greatest environmental concern is air pollution in the form of fine PM, which consists of liquid and solid particle compounds that are dangerous to human health. According the experimental results, we have optimized the LSTM and with a learning rate of 0.01, epoch of 100, and batch sizes of 32, 64, 128, and 256. For the prediction result, the LSTM model had minimum RMSE values of 11.113for PM10 and 12.174 for PM2.5 at a batch size of 32. At the same time, the DAEs model had minimumRMSE values of 15.038 for PM10 and 15.431 for PM2.5 at a batch size of 64. Also compared thetotal average RMSE of prediction of PM10 and PM2.5, the LSTM prediction model were more accuratethan the DAE model.
9	Assessment and prediction of air quality using fuzzy logic and autoregressive models [2]	2012	In the proposed SD method, wavelet decomposition (WD) is chosen as the primary decomposition technique to generate a high frequency detail sequence WD(D) and a low frequency approximation sequence WD(A). Long short-term memory (LSTM) neural network with good ability of learning and time series memory is applied to make it easy to be predicted. The proposed BA-LSSVM model considering air pollutant factors is applied to forecast WD(A). The proposed optimal-hybrid model outperforms other hybrid models.	In the first step the air quality parameters are predicted, and in the second step the fuzzy inference system assesses predicted values having as a result a predicted air quality index. Three months of measurements were extracted from data base (January, February and March, 2008). In other words, for one day of information, the next 24 h can be predicted using the AR model, and in the second step the predicted values are processed by the fuzzy inference system, calculating the predicted AQI (P-AQI),inference system, calculating the predicted AQI (P-AQI). The P-AQI performances were evaluated using correlation coefficient (R), mean error (ME), root mean square error (RMSE) and normalized root mean square error (NRMSE).

Comparative analysis between prior research works Table 5

SNo.	Paper Title	Paper Year	Approach	Results
10	A deep learning model for air quality prediction in smart cities[7]	2017	In this paper, a novel deep learning model is proposed for analyzing IoT smart city data. The dataset contains 8 features including ozone, particulate matter, carbon monoxide, sulfur dioxide, nitrogen dioxide, longitude, latitude and timestamp was used for experiment. The dataset has 17568 samples that are collected at five-minute intervals. Each sample value is given in the form of EPA's AQI standard. In this study, ozone and nitrogen dioxide pollutants are selected for air quality prediction. Training set %: 69.5 and test set %: 30.5. They proposed a novel model based on Long Short Term Memory (LSTM) networks to predict future values of air quality in a smart city.	proposed model have the lowest error rates in Yellow and Green alarms. In red alarms, it has a bit higher error rates than SVR. In this paper, they proposed a DL model to overcome air pollution problems in SC. We firstly configured the network with the best hyper parameters according to the results obtained from the experiments. Then, the proposed model is trained, and evaluated with widely used RMSE and MAE metrics. Consequently, the obtained results show that the employment of the LSTM based prediction model to the IoT data is effective and promising.

Baseline Comparison Table

SNo.	Algorithm	Paper used for Comparison	Performance Rate of algo applied in Paper	Performance Rate in our Project
1	KNN Classification	AIR QUALITY INDEX PREDICTION IN BESIKTAS DISTRICT BY ARTIFICIAL NEURAL NETWORKS AND K NEAREST NEIGHBORS [1]	92.86% (n _neighbours is 5)	90.78% (n_neighbours is 5)
2	Decision Trees	Ambient air quality estimation using supervised learning techniques [9]	95.5 %	96.11 %
3	Naive Bayes	Ambient air quality estimation using supervised learning techniques [9]	73.93 %	72.28 %
4	Random Forest	Ambient air quality estimation using supervised learning techniques [9]	99.3 %	97.53%
5	Logistic Regression	None	None	79.43%