**Predicting Air Pollution Using Machine Learning Algorithms**

**INDEX**

|  |  |
| --- | --- |
| **CHAPTERS** | **PAGE NO.** |
| **1 INTRODUCTION**   * 1. Problems with Air Quality Index   2. Introduction to Machine Learning   3. Machine Learning Methods      1. Supervised Algorithm      2. Unsupervised Algorithm      3. Reinforcement Algorithm   4. Importance of Machine Learning   5. Machine Learning Algorithms      1. Classification Algorithms      2. Regression Algorithms  1. **LITERATURE SURVEY**   **3 SYSTEM ANALYSIS**  3.1 Existing System  3.2 Proposed System  3.3 Advantages of Proposed System  3.4 Proposed Architecture  3.5 Detecting the Pollutants  3.6 Major Pollutants  3.7 Algorithms detecting pollutants  **4 REQUIREMENTS AND SPECIFICATIONS**  4.1 Scope of the System  4.2 Objective of the System  4.3 Functional Requirements  4.4 Non Functional Requirements  4.5 System Requirements  4.6 Software Development Environment  4.6.1 Introduction to Python  4.6.2 Characteristics of Python  4.6.3 Python Applications  4.6.4 Python with Machine Learning  4.6.5 Machine Learning with Python  4.6.6 Based on Learning Ability  4.6.6.1 Batch Learning  4.6.6.2 Online Learning  4.6.7 Based on Generation Approach  4.6.7.1 Instance Based Learning  4.6.7.2 Model Based Learning  4.7 Feasibility Study  4.7.1 Economical Feasibility  4.7.2 Technical Feasibility  4.7.3 Social Feasibility  **5 SYSTEM DESIGN**  5.1 Introduction  5.1.1 Input Design  5.1.2 Objectives  5.1.3 Output Design  5.1.4 Objectives  5.2 Data Flow Diagrams  5.3 UML Diagrams  5.3.1 Structural Diagrams  5.3.2 Behavioral Diagrams  **6** **IMPLEMENTATION OF CODE**  **7 TESTING**  7.1 Introduction  7.1.1 Testing Objectives  7.1.2 Testing Approaches  7.2 Unit Testing  7.3 Integration Testing  7. Functional Test  7.5 System Test  7.6 White Box Testing  7.7 Black Box Testing  7.8 Acceptance Testing  **8 SCREENSHOTS**  **9 CONCLUSION AND FUTURE ENHANSEMENT**  **10 REFERENCES** | 1  2  3  4  4  4  5  5  6  6  7  9  13  13  13  14  15  16  16  18  21  21  21  21  21  21  22  22  22  22  24  24  25  25  26  26  26  27  27  27  27  28  29  29  29  29  29  29  30  30  30  31  35  48  48  48  48  49  49  49  50  50  50  50  51  54  55 |

**ABSTRACT**

The survival of mankind cannot be imagined without air. Consistent developments in almost all realms of modern human society affected the health of the air adversely. Daily industrial, transport, and domestic activities are stirring hazardous pollutants in our environment. Monitoring and 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 or noise pollution can lead to irritation and loss of quality of life that may harm health. Air quality Index measure the quality of air. Air pollution may be caused by both human and natural activity. So2, Carbon dioxide, Nitrogen dioxide, carbon monoxide, chlorofluorocarbons (CFCs), coal, lead, mercury, wood and other pollutants are released into the atmosphere as a result of human activity. 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. Dataset are available from Kaggle or Central pollution control board (CPCB), India dataset air quality monitoring sites and divided into two Training and Testing. The dataset is split into train-test subsets by the ratio of 75–25% respectively. ML-based AQI prediction is carried out with and without SMOTE resampling technique and a comparative analysis is presented. The results of ML models for both the train-test subsets are presented in terms of standard metrics like accuracy, precision, recall, and F1-Score. For both the train-test sets, the XGBoost model attained the highest accuracy and the SVM model exhibited the lowest accuracy. The classical statistical error metrics, namely MAE(Mean Absolute Error), RMSE(Root Mean Square Error), RMSLE(Root Mean Squared Error of the log-transformed predicted and log-transformed actual values), and R2 are then evaluated to assess and compare the performances of ML models.

6

**CHAPTER-1**

**INTRODUCTION**

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 the survival of mankind cannot be imagined without air. Consistent developments in almost all realms of modern human society affected the health of the air adversely. Daily industrial, transport, and domestic activities are stirring hazardous pollutants in our environment. Monitoring and 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 or noise pollution can lead to irritation and loss of quality of life that may harm health. Air quality Index measure the quality of air. Air pollution may be caused by both human and natural activity. SO2, Carbon dioxide, Nitrogen dioxide, carbon monoxide, chlorofluorocarbons (CFCs), coal, lead, mercury, wood and other pollutants are released into the atmosphere as a result of human activity. Air Pollution can cause severe disease like lungs cancer, brain disease and even lead to death.

Two types of Pollutants that is causing air pollution are:

1) Primary Pollutants

2) Secondary Pollutants

* Primary Pollutants released into air directly from the source.
* Secondary Pollutants formed by reacting with either primary pollutants or with another atmospheric component.

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.

The algorithms used are:

**Regression Algorithms**: Linear Regression, Decision Tree Regressor, Random Forest Regressor

**Classification Algorithms**: Logistic Regression, Decision Tree Classifier, Random Forest Classifier, KNN - K Nearest Neighbour

**1.1 PROBLEMS WITH AIR QUALITY INDEX**

Due to lack of monitoring and maintenance, most of the authorized centres have failed to record real-time data on air pollution. On April 6, 2015, India had launched its first ever National Air Quality Index (NAQI) and acknowledged the problem of its incessantly rising air pollution. This index was launched by Prime Minister Narendra Modi in New Delhi, starting with 10 cities in the first phase.

While this was a good move, we still need answers on how to tackle the bigger problem of our deteriorating air quality and protecting the health of citizens from its ill effects.

NAQI is tool that uses numbers to simplify air quality data by classifying pollution levels into 6 categories good, satisfactory, moderate, poor, very poor and severe and denotes a colour code based on how harmful the pollution in a specific area is. This makes it easier for people to know how bad the pollution in their area is and which places they should avoid on that day. It is certain that awareness will go up with availability of a monitoring tool like this. But what is missing is that no one knows the next step. The government has not at all talked about an aggressive action plan that can tell people what precautionary actions to take based on pollution levels and how to combat the bigger problem of air pollution.

For this AQI to function efficiently, the monitoring stations in the designated areas need to function properly. While there were many expectations from this AQI, it has already started to show glitches that exist due to improper maintenance.

There is a need to expand AQI and include most of the cities in the list of the air quality index programme so that the citizens all over the country know about the pollution levels in their areas. They should also be aware of the steps they can take to protect themselves.

Globally, the air quality index feature fares very well, along with a health advisory that is issued to public, based on the severity of pollution. But for India which is still struggling to get its data right, this would be a far cry. So, it is to be seen what could be the solution to a problem that was implemented as a solution.

**1.2 AN INTRODUCTION TO MACHINE LEARNING**

Machine Learning is said as a subset of Artificial Intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959. We can define it in a summarized way as:

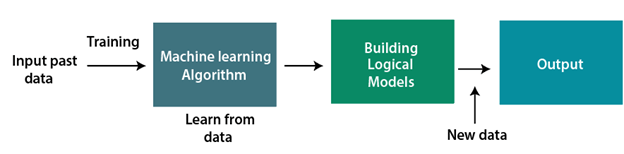
“Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed”

With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building mathematical models and making predictions using historical data or information**. Currently, it is being used for various tasks such as **image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system**, and many more.

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:



**Fig1: Working of Machine Learning Algorithm**

**1.3 MACHINE LEARNING METHODS**

At a broad level, machine learning can be classified into three types:

* Supervised learning
* Unsupervised learning
* Reinforcement learning

**1.3.1 Supervised Learning:**

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

Supervised learning can be grouped further in two categories of algorithms:

* Classification
* Regression

**1.3.2 Unsupervised Learning:**

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classifieds into two categories of algorithms:

* **Clustering**
* **Association**

**1.3.3 Reinforcement Learning:**

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

**1.4 IMPORTANCE OF MACHINE LEARNING**

Machine Learning is one of the most popular sub-fields of Artificial Intelligence. Machine learning concepts are used almost everywhere, such as Healthcare, Finance, Infrastructure, Marketing, Self-driving cars, recommendation systems, chatbots, social sites, gaming, cyber security, and many more.

Currently, Machine Learning is under the development phase, and many new technologies are continuously being added to Machine Learning. It helps us in many ways, such as analysing. large chunks of data, data extractions, interpretations, etc. Hence, there are unlimited numbers of uses of Machine Learning.

Machine learning is important because it gives enterprises a view of trends in customer behaviour and operational business patterns, as well as supports the development of new products*.* Many of today's leading companies, such as **Facebook, Google,** and **Uber** make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

The machine learning field is continuously evolving and along with evolution comes a rise in demand and importance. There is one crucial reason why data scientists need machine learning, and that is: ‘High-value predictions that can guide better decisions and smart actions in real-time without human intervention.’

Machine learning as technology helps analyse large chunks of data, easing the tasks of data scientists in an automated process and is gaining a lot of prominence and recognition. Machine learning has changed the way data extraction and interpretation works by involving automatic sets of generic methods that have replaced traditional [statistical techniques](https://www.simplilearn.com/tutorials/machine-learning-tutorial/statistics-for-machine-learning).

**1.5 MACHINE LEARNING ALGORITHMS**

Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms.

**1.5.1 Classification Algorithms**

The Classification algorithm is a Supervised Learning technique used to identify the category of new observations based on training data. The main goal of is to identify the category of a given dataset, and to predict the output for the categorical data. They include:

**1.5.1.1 Logistic Regression**

Logistic regression comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function.

**1.5.1.2 K-Nearest Neighbour (KNN) Algorithm**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. KNN 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. KNN 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. KNN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. 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 like the new data.

**1.5.1.3 Random Forest Algorithm**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique**. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.** 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. It takes less training time as compared to other algorithms. It predicts output with high accuracy, even for the large dataset it runs efficiently. It can also maintain accuracy when a large proportion of data is missing.

**1.5.1.4 Decision Tree Classification Algorithm**

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 rulesandeach 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. 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, like a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

**1.5.2 Regression Algorithms**

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. They include:

**1.5.2.1 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 such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (x) 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.

**1.5.2.2 Decision Tree Regression**

Non-linear regression in Machine Learning can be done with the help of decision tree regression. The main function of the decision tree regression algorithm is to split the dataset into smaller sets. The subsets of the dataset are created to plot the value of any data point that connects to the problem statement. The splitting of the data set by this algorithm results in a decision tree that has decision and leaf nodes. ML experts prefer this model in cases where there is not enough change in the data set. One should know that even a slight change in the data can cause a major change in the structure of the subsequent decision tree. One should also not prune the decision tree regressors too much as there will not be enough end nodes left to make the prediction. To have multiple end nodes (regression output values), one should not prune the decision tree regressors excessively.

**1.5.2.3 Random Forest Regression Algorithm**

Random forest is also a widely-used algorithm for non-linear regression in Machine Learning. Unlike decision tree regression (single tree), a random forest uses multiple decision trees for predicting the output. Random data points are selected from the given dataset (say k data points are selected), and a decision tree is built with them via this algorithm. Several decision trees are then modelled that predict the value of any new data point. Since there are multiple decision trees, multiple output values will be predicted via a random forest algorithm. You must find the average of all the predicted values for a new data point to compute the final output.

**CHAPTER-2**

**LITERATURE SURVEY**

# [1] Yanzhao Li and et al. [2022] this study conducts a content analysis to investigate current topical interests to identify research gaps and propose future research directions. The analytical framework and the findings provide helpful insights into the prospects in air quality forecasting with AI techniques. AI techniques, represented by ANNs and other machine learning (ML) models, have completely changed the form of research in this field due to their more accurate results and lower requirement for initial data.

**[2] Khalid Mehmoodand et al. [2022]** presents a bibliometric analysis of all published articles on the use of machine learning networks to predict air quality found in the Web of Science (WoS) search engine from 1992 to 2021 discussed how conventional methods were transformed into machine learning approaches. It employs an innovative bibliometric technique to determine the major progress and new insights in this field, and by identifying the current hot zones and trends. Results revealed evidence of a surge of interest in air quality prediction with machine learning models. In total 658 articles were published during the emerging phase (1992–2016), representing 31.48% of the total.

**[3] R.Janarthanan and et al. [2021**] used the combination of Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) based deep learning model to classify the AQI values. The proposed deep learning model gives an accurate and specific value for AQI on the city’s specified location compared to the existing techniques. The prediction accuracy is improved in the proposed deep learning method, which will caution the public to reduce to an acceptable level. The deep learning mechanism predicts the AQI values accurately and helps to plan the metropolitan city for sustainable development. The expected AQI value can control the pollution level by incorporating road traffic signal coordination, encouraging the people to use public transportation, and planting more trees on some locations.

**[4] Chi-YehLin and et al. [2021]** proposed an ensemble learning forecasting model, named MLEGRU (Multiple Linear Regression based Gated Recurrent Unit) model based on multiple linear regression technique to integrate these deep learning predictive models provides a better forecasting accuracy than other ensemble learning methods.

**[5] Houxin Cui and et al. [2021]** developed different calibration methods to enhance the data quality of low-cost sensors. In this study, based on the laboratory and field data collected in mainland China, the performances of PM2.5, PM10, [CO](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/carbon-monoxide), NO2, SO2, and O3 sensors were estimated. Then, the sensors were calibrated by different methods and their output data were compared to the reference data to determine the validity and performance of these calibration methods. The results indicated that 1) field calibration with supervised learning technique is an effective technique to decrease the error, and 2) mobile air quality monitoring stations can calibrate the sensors at where standard air quality monitoring stations are not available. In conclusion, a complete laboratory and field calibration system (four-stage calibration) covering the whole life cycle of the sensor is proposed in this paper.

**[6] A. Gnana Soundari** **and et al. [2021]** developed a model to prognosticate the air quality indicator grounded on literal data of former times and prognosticating over a particular forthcoming time as a Gradient decent boosted multivariable retrogression problem and achieved better performance than the standard retrogression models our model is able of prognosticating the current data with 95 delicacy it'll successfully prognosticate the forthcoming air quality indicator of any particular data within a given region and it's able of tracing back to the particular position demanded attention handed the time series data of every possible region demanded attention.

**[7] Masoomeh Zeinalnezhad and et al. [2020]** aimed to respond to the limitation by improving the accuracy of the daily prediction of pollutants via time-series data analysis by using Adaptive Neuro-Fuzzy Inference System modelling. A nonlinear multivariate regression model was developed and experimentally refined to obtain the least error possible. This study presents two models of semi-experimental nonlinear regression and ANFIS to predict the concentration of the four important pollutants. Considering the health effects and extent of their sources of propagation in continuous measurements, CO, O3, SO2, and NO2 were selected. The significance of this research comes from the necessity of a more accurate prediction to do timely.

**[8] RicardoNavares and et al. [2019]** compared several comprehensive deep network configurations to identify those which are able to better extract relevant information out of the set of time series in order to predict one day-ahead air quality. The results, supported by statistical evidence, indicate that a single comprehensive model might be a better option than multiple individual models. This paper presents a comparison study of different LSTM configurations in order to obtain the most suitable to forecast air quality in the region of Madrid.

**[9] W. C Leong and et al. [2019**] proposed a support vector machine to model the air pollution index and kernel functions model parameters are investigated. The data screening technique was applied to deal with the missing data as well as the outlier to improve the performance. The best setting for API SVM model is using radial basis function (RBF) kernel function with the coefficient of determination (R2) value of 0.9843 and sum square error (SSE) and mean sum square error (MSSE) values of 2008 and 1.444 respectively.

**[10] Gaganjot Kaur Kang and et al. [2018]** investigated on various big-data and machine learning based techniques for air quality forecasting and published research results relating to air quality evaluation using methods of artificial intelligence, decision trees, deep learning etc. Furthermore, it throws light on some of the challenges and future research needs. 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. It highlights some observations on future research issues, challenges, and needs.

**[11] Aditya C R and et al. [2018]** Logistic regression is employed to detect whether a data sample is either polluted or not polluted. Auto regression is employed to predict future values of PM2.5 based on the previous PM2.5 readings. This system attempts to predict PM2.5 level and detect air quality based on a data set consisting of daily atmospheric conditions in a specific city. The proposed system will help to detect and predict pollution levels and take the necessary action in accordance with that. Also, this will help people establish a data source for small localities which are usually left out in comparison to the large cities

**[12] Giorgio Corani and et al. [2016]** designed a multi-label classifier based on Bayesian networks and learned its structure through structural learning. Presented experiments in three different case studies regarding the prediction of PM2.5 and ozone. The multi-label classifier computes more accurate posterior probabilities which better support the decision maker.

**[13] Ana Russo and et al. [2013]** The reduced set of variables including these derived variables is therefore proposed as an optimal variable set for training neural network models in forecasting geophysical and weather properties. The introduction of the stochastic variables as input data for training the ANN model allows to preserve the predictive power with considerable less input.

**[14] José Juan Carbajal-Hernández and et al. [2012]** a new model based on fuzzy inference systems has been introduced to assess air quality status. The proposed AQI works in two steps: first, the toxicity of a set of measured concentrations is classified by levels (Sigma operator); second, the effects in the ecosystem are evaluated in order to determine air quality status (fuzzy inference system). Our results show that our models are an appropriate tool for assessing site pollution and for providing guidance to improve contingency actions in urban areas.

**[15] Elia Georgiana Dragomir and et al. [2010]** the KNN technique is used in the air quality forecast domain in order to predict the value of the air quality index. This index is used to categorize the pollution level and to inform the population about some possible episodes of pollution. In this paper, we have presented an experiment done in order to determine the particularities of applying this technique for air quality analysis. Aiming at generating a prediction for the air quality index, training data that were collected in June 2009 were used as input data for the algorithm. The experimental results show that 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. 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.

**CHAPTER-3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

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. Consistent developments in almost all realms of modern human society affected the health of the air adversely. Daily industrial, transport, and domestic activities are stirring hazardous pollutants in our environment. Monitoring and 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 or noise pollution can lead to irritation and loss of quality of life that may harm health. Air quality Index measure the quality of air. Air pollution may be caused by both human and natural activity. SO2, Carbon dioxide, Nitrogen dioxide, carbon monoxide, chlorofluorocarbons (CFCs), coal, lead, mercury, wood and other pollutants are released into the atmosphere as a result of human activity. 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. Dataset are available from Kaggle or Central pollution control board (CPCB), India dataset air quality monitoring sites and divided into two Training and Testing. The dataset is split into train-test subsets by the ratio of 75–25% respectively. ML-based AQI prediction is carried out with and without SMOTE resampling technique and a comparative analysis is presented. The results of ML models for both the train-test subsets are presented in terms of standard metrics like accuracy, precision, recall, and F1-Score. For both the train-test sets, the XGBoost model attained the highest accuracy and the SVM model exhibited the lowest accuracy. The classical statistical error metrics, namely MAE (Mean Absolute Error), RMSE (Root Mean Square Error), RMSLE (Root Mean Squared Error of the log-transformed predicted and log-transformed actual values), and R2 are then evaluated to assess and compare the performances of ML models.

**3.2 PROPOSED SYSTEM**

Air pollution is a potent greenhouse gas. The Air pollution level has been rising every day due to the rising air pollution, numerous automobiles, mechanization and urbanization. The main pollutant emissions in India are due to the energy productions industry vehicle traffic on the roads, soil and road dust, waste incineration, power plants, open waste burning etc. The present research investigate air pollution data started from the Central Pollution Control Board(CPCB). Analysis of some major pollutants such as PM2.5, PM10, NO2, SO2, O3, etc and predicate of AQI the essence of the current work.

Quality of data is the first and most important prerequisite for effective visualization and creation of efficient ML models. The pre-processing steps help in reducing the noise present in the data which eventually increases the processing, speed and generalization capability of ML algorithms. Outliers and missing data are the two most common errors in data extraction and monitoring applications. The data pre-processing step performs various operations on data such as filling out not-a-number (NAN) data, removing or changing outlier data, etc. A large number of missing values may be existing due to a variety of factors, such as a station that can sense data but does not possess a device to record it. All the missing values are filled with the median values against each feature to solve the missing data problem. Next, a normalization process has been applied to standardize the data, ensuring that the significance of variables is unaffected by their ranges or units. The data normalization process helps to bring different data attributes into a similar scale of measurement. This process plays a vital role in the stable training of ML models and boosts performance. The data types of all the variables are also examined during normalization.

The data-set under study involves a specific parameter viz, AQI and government agencies use this parameter to alert people about the quality of the air and also practice forecasting it. Scholars in the realm suggest that reducing input variables lowers the computational cost of modeling and enhances prediction performance. Applying Machine learning for predicting Air Pollution such as Decision tree classifier, Random forest classifier, KNN(K-nearest neighbour), logistics regression could achieve impressive results in providing best accurate, efficient prediction.

**3.3 ADVANTAGE OF PROPOSED SYSTEM**

It helps in checking the visibility of the air and the save used from the root accidents. It helps to improve the air pollution by knowing the pollution of the city. Air quality production helps in preventing from the disease like lungs cancer. It helps to check the quality for casting in hourly basis.

It helps monitor the presence of pollutants resulting in better environmental conditions for humans to reside. this also impacts their health and reduce the chances of occurring any health issues by maintaining a moderate ambiance or as required.

**3.4 PROPOSED ARCHITECTURE**

Air Pollution Dataset

Data Processing

Normalization & Applying Outlier

Testing set 25%

Training set 75%

Regression Algorithm

Classification Algorithm

Machine Learning Algorithm

Best model prediction by Accuracy

Finding air quality by AQI

**Fig2 : Architecture for predicting Air Pollution Index**

**3.5 DETECTING THE POLLUTANTS CAUSING AIR POLLUTION**

Air pollution can be prevented by advocating the use of public transport and carpooling. It can also be controlled by avoiding wastage of electricity and practicing reuse and recycling of compatible products.

**Common Air Pollutants that cause Air Pollution**

* Sulphur dioxide.
* Nitrogen oxides.
* Particular matter (PM10, PM2.5 and PM1)
* Toxic Organic Micro-Pollutants (TOMPS)
* Carbon monoxide.

Air Pollution analysis tells you how clean or polluted your air is, and what associated health effects might be a concern for you.

The first step in air pollutant concentration is to define clearly the objectives of measuring concentrations of air pollutants, and to determine the target substances for measurement. The objectives that are usually intended areas follows.

1. To thoroughly understand the pollution levels and concentration fluctuations when the sources and pollutants are clearly known.

2. When neither the source nor the pollutants are clearly known, to examine the causes of the pollution.

3. When the source is identified, to thoroughly understand the levels of pollution and the types of pollutants emitted.

4. When the pollutant has been identified, to specify the source and to determine the level of its contribution.

5. To easily understand border area concentrations and exposure sites for voluntary periods of time.

6. To thoroughly understand the broad range pollution level and its fluctuations.

**3.6 MAJOR POLLUTANTS USED IN OUR PROJECT**

Analysis of some major air pollutants such as

* SO2
* NO2
* RSPM (Respirable Suspended Particulate Matter)
* SPM (Suspended Particulate Matter) etc.

**Sulphur dioxide air pollutant**

SO2 gas is emitted by the burning of fossil fuels or other materials that contain sulfur. Sulfur dioxide can damage trees and plants, inhibit plant growth, and damage sensitive ecosystems and waterways. It also can contribute to respiratory illness and aggravate existing heart and lung conditions.

Health effects caused by exposure to high levels of SO2 include breathing problems, respiratory illness, changes in the lung's defenses, and worsening respiratory and cardiovascular disease. People with asthma or chronic lung or heart disease are the most sensitive to SO2. It also damages trees and crops.

**Nitrogen dioxide air pollutant**

Nitrogen dioxide or NO2, is a gaseous air pollutant composed of nitrogen and oxygen and is one of a group of related gases called nitrogen oxides, or NOx. NO2 forms when fossil fuels such as coal, oil, gas or diesel are burned at high temperatures. NO2 and other nitrogen oxides in the outdoor air contribute to particle pollution and to the chemical reactions that make ozone. It is one of six widespread air pollutants that have national air quality standards to limit them in the outdoor air. NO2 can also form indoors when fossil fuels like wood or natural gas are burned.

Nitrogen dioxide causes a range of harmful effects on the lungs, including:

* Increased inflammation of the airways.
* Worsened cough and wheezing.
* Reduced lung function.
* Increased asthma attacks.
* Greater likelihood of emergency department and hospital admissions.

**RSPM air pollutant**

RSPM called Respirable Suspended Particulate Matter, these particles cause the worst damage as they can penetrate deep into the lungs. The CPCB only monitors total SPM. In other words, the quantity of all particles in the air.

**SPM air pollutant**

Suspended particulate matter (SPM) are finely divided solids or liquids that may be dispersed through the air from combustion processes, industrial activities or natural sources.

* To do data analysis on India Air Quality data and predict the value of Air Quality Index based on given features of concentration of Sulphur dioxide, nitrogen dioxide, respirable suspended particulate matter, suspended particulate matter and classify the Air Quality as good, moderate, poor, unhealthy, healthy.

**3.7 ALGORITHMS DETECTING THE POLLUTANTS**

Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms.

**Regression Algorithms:** Regression finds correlations between dependent and independent variables. Therefore, regression algorithms help predict continuous variables such as house prices, market trends, weather patterns, oil and gas prices (a critical task these days!), etc.

* Linear Regression
* Decision Tree Regressor
* Random Forest Regressor

**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 such as sales, salary, age, product price, etc. 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.

**Decision Tree Regression:**

Non-linear regression in Machine Learning can be done with the help of decision tree regression. The main function of the decision tree regression algorithm is to split the dataset into smaller sets. The subsets of the dataset are created to plot the value of any data point that connects to the problem statement. The splitting of the data set by this algorithm results in a decision tree that has decision and leaf nodes. ML experts prefer this model in cases where there is not enough change in the data set. One should know that even a slight change in the data can cause a major change in the structure of the subsequent decision tree. One should also not prune the decision tree regressors too much as there will not be enough end nodes left to make the prediction. To have multiple end nodes (regression output values), one should not prune the decision tree regressors excessively. This trains a model in structure of a tree to predict data in the future to produce meaningful continuous output.

**Random Forest Regression Algorithm:**

Random forest is also a widely-used algorithm for non-linear regression in Machine Learning. Unlike decision tree regression (single tree), a random forest uses multiple decision trees for predicting the output. Random data points are selected from the given dataset (say k data points are selected), and a decision tree is built with them via this algorithm. Several decision trees are then modelled that predict the value of any new data point. Since there are multiple decision trees, multiple output values will be predicted via a random forest algorithm. You must find the average of all the predicted values for a new data point to compute the final output. This happens due to the large number of decision trees mapped under this algorithm, as it requires more computational power. It's a bagging technique not a boosting technique trees run parallel. i.e no interaction between these tree while building trees.

**Classification Algorithms:**

Classification is an algorithm that finds functions that help divide the dataset into classes based on various parameters. a computer program gets taught on the training dataset and categorizes the data into various categories depending on what it learned.

* Logistic Regression
* Decision Tree Classifier
* Random Forest Classifier
* KNN - K Nearest Neighbour

**Logistic Regression:**

Logistic regression comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

**Decision Tree Classification Algorithm:**

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. 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, like a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. It can be used for both Regression and Classification.

**Random Forest Algorithm:**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. 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. It predicts output with high accuracy, even for the large dataset it runs efficiently. It can also maintain accuracy when a large proportion of data is missing.

**K-Nearest Neighbour (KNN) Algorithm:**

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. 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 like the new data.

**CHAPTER-4**

**REQUIREMENTS AND SPECIFICATIONS**

**4.1 SCOPE OF THE SYSTEM**

The primary aim of this study is to determine the most accurate supervised machine learning algorithm to build a model that can predict the AQI. We also aim to outline how machine learning helps in controlling air pollution.

**4.2 OBJECTIVE OF THE SYSTEM**

The main objective of this project is to build and train a model using machine learning algorithms and to evaluate the performance and accuracy of created models and determine the most accurate supervised machine learning algorithm in the prediction of AQI.

**4.3 FUNCTIONAL REQUIREMENTS**

In Software engineering, a functional requirement defines a function of a software system or its components. A function is described as set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements which impose constants on the design or implementation.

**4.4 NON-FUNCTONAL REQUIREMENTS**

In System engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system rather than specific behaviour. In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be.

**4.5 SYSTEM REQUIREMENTS**

**4.5.1 SOFTWARE REQUIREMENTS**

Operating system: - Windows XP.

Coding Language: PYTHON

**4.6 SOFTWARE DEVELOPMENT ENVIRONMENT**

**4.6.1 AN INTRODUCTION TO PYTHON**

Python is a popular object-oriented programming language having the capabilities of high-level programming language. It is easy to learn syntax and portability capability makes it popular these days. The chapter describes about the software tool that is used in our project.

Python was developed by Guido van Rossum at Stitching Mathematics Centrum in the Netherlands. It was written as the successor of programming language named ‘ABC’. It is first version was released in 1991. The name Python was picked by Guido van Rossum from a TV show named Monty Python’s Flying Circus. It is an open-source programming language which means that we can freely download it and use it to develop programs.

**4.6.2 CHARACTERISTICS OF PYTHON**

Following are important characteristics of **Python Programming** –

• It supports functional and structured programming methods as well as OOP.

• It can be used as a scripting language or can be compiled to byte-code for

building large applications.

• It provides very high-level dynamic data types and supports dynamic type

checking.

• It supports automatic garbage collection.

• It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**4.6.3 PYTHON APPLICATIONS**

Python is known for its general-purpose nature that makes it applicable in almost each domain of software development. Python can be used in any sphere of development. Here, we are specifying applications areas where python can be applied.

**1) Web Applications**

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, 27beautiful Soup, Feed parser etc. It also provides Frameworks such as Django, Pyramid, Flask etc to design and develop web-based applications. Some important developments are: PythonWikiEngines, Pocoo, PythonBlogSoftware etc.

**2) Desktop GUI Applications**

Python provides Tk GUI library to develop user interface in python-based application. Some other useful toolkits widgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

**3) Software Development**

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

**4) Scientific and Numeric**

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science, and mathematics.

**5) Business Applications**

Python is used to build Business applications like ERP and e-commerce systems. Tryton is a high-level application platform.

**6) Console Based Application**

We can use Python to develop console-based applications. For example: **IPython**.

**7) Audio or Video based Applications**

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: Tim Player, cplay etc.

**8) 3D CAD Applications**

To create CAD application Fandango is a real application which provides full features of CAD.

**9) Enterprise Applications**

Python can be used to create applications which can be used within an Enterprise or an organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

**10) Applications for Images**

Using Python several applications can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.

**4.6.4 PYTHON WITH MACHINE LEARNING**

Machine Learning (ML) is basically that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do. In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. The key focus of ML is to allow computer systems to learn from experience without being explicitly programmed or human intervention.

**4.6.5 MACHINE LEARNING WITH PYTHON**

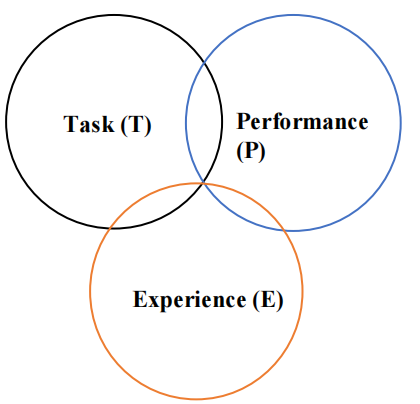
Machine Learning (ML) is that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do. In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. The focus of ML is to allow computer systems learn from experience without being explicitly programmed or human intervention.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we cannot do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises. ML is a field of AI consisting of learning algorithms that improve their performance (P) at executing some Task (T) over time with experience (E) Based on the above, the following diagram represents a Machine Learning Model.

Task(T) From the perspective of problem, we may define the task T as the real-world problem to be solved. The problem can be anything like finding best house price in a specific location or to find best marketing strategy etc. On the other hand, if we talk about machine learning, the definition of task is different because it is difficult to solve ML based tasks by conventional programming approach.

A task T is said to be a ML based task when it is based on the process and the system must follow for operating on data points. The examples of ML based tasks are Classification, Regression, Structured annotation, Clustering, Transcription etc.

Experience (E) As name suggests, it is the knowledge gained from data points provided to the algorithm or model. Once provided with the dataset, the model will run iteratively and will learn some inherent pattern. The learning thus acquired is called experience(E). Making an analogy with human learning, we can think of this situation as in which a human being is learning or gaining some experience from various attributes like situation, relationships etc. Supervised, unsupervised and reinforcement learning are some ways to learn or gain experience. The experience gained by out ML model or algorithm will be used to solve the task T.

Performance (P) An ML algorithm is supposed to perform task and gain experience with the passage of time. The measure which tells whether ML algorithm is performing as per expectation or not is its performance (P). P is basically a quantitative metric that tells how a model is performing the task, T, using its experience, E. There are many metrics that help to understand the ML performance, such as accuracy score, F1 score, confusion matrix, precision, recall, sensitivity etc.

**Fig 3 : Conventional programming Approach**

**4.6.6 BASED ON LEARNING ABILITY**

**4.6.6.1 BATCH LEARNING**

In many cases, we have end-to-end Machine Learning systems in which we need to train the model in one go by using whole available training data. Such kind of learning method or algorithm is called Batch or Offline learning. It is called Batch or Offline learning because it is a one-time procedure and the model will be trained with data in one single batch. The following are the main steps of Batch learning methods –

**Step 1** − First, we need to collect all the training data for start training the model.

**Step 2** −Now, start the training of model by providing whole training data in one go.

**Step 3** − Next, stop learning/training process once you got satisfactory results/performance.

**Step 4** − Finally, deploy this trained model into production. Here, it will predict the output for new data sample.

**4.6.6.2 ONLINE LEARNING**

It is completely opposite to the batch or offline learning methods. In these learning methods, the training data is supplied in multiple incremental batches, called mini batches, to the algorithm. Followings are the main steps of Online learning methods

**Step 1** − First, we need to collect all the training data for starting training of the model.

**Step 2** − Now, start the training of model by providing a mini-batch of training data to the algorithm.

**Step 3** − Next, we need to provide the mini-batches of training data in multiple increments to the algorithm.

**Step 4** − As it will not stop like batch learning hence after providing whole training data in mini-batches, provide new data samples also to it.

**Step 5** − Finally, it will keep learning over a period based on the new data samples.

**4.6.7 BASED ON GENERALIZATION APPROACH**

In the learning process, followings are some methods that are based on generalization approaches

**4.6.7.1 INSTANCE BASED LEARNING**

Instance based learning method is one of the useful methods that build the ML models by doing generalization based on the input data. It is opposite to the previously studied learning methods in the way that this kind of learning involves ML systems as well as methods that uses the raw data points themselves to draw the outcomes for newer data samples without building an explicit model on training data. In simple words, instance-based learning basically starts working by looking at the input data points and then using a similarity metric, it will generalize and predict the new data points.

**4.6.7.2 MODEL BASED LEARNING**

In Model based learning methods, an iterative process takes place on the ML models that are built based on various model parameters, called hyperparameters and in which input data is used to extract the features. In this learning, hyperparameters are optimized based on various model validation techniques. That is why we can say that Model based learning methods uses more traditional ML approach towards generalization.

**4.7 FEASIBILITY STUDY**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

ECONOMICAL FEASIBILITY

TECHNICAL FEASIBILITY

SOCIAL FEASIBILITY

**4.7.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**4.7.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing the system.

**4.7.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**CHAPTER-5**

**SYSTEM DESIGN**

**5.1 INTRODUCTION**

The design of a system is essentially a blueprint or a plan for a solution for the system. Here we consider a system to be set of components with clearly defined behaviour that interacts with each other in a fixed defined manner to produce some behaviour or services for its environment.

**5.1.1 Input Design**

The input is the raw data that is processed to produce output. The input determines the quality of system output. Input design is a system to detect and predict pollution and should be able to work in both classification and regression algorithms by collecting dataset for validation and evaluation, and to recognize or predict AQI value with provided algorithms showing better accuracy.

**5.1.2 Objectives**

* To reduce input volume
* To remove outliers and anomalies
* To make the input free from errors
* To use validation checks and develop effective input controls

**5.1.3 Output Design**

The design of the output is the most important task of any system. A quality output is one, which meets the requirements of the end user and presents the information clearly. Here, the type of output needed is identified, and consider the necessary output controls and prototype report layouts. Efficient and intelligent output design improves the systems relationship to help user decision making.

**5.1.4 Objectives**

* Convey information about pollutants, rate of pollution, AQI value, prediction of increase in pollutant percentage in future.
* Evaluating best algorithm for air pollution prediction by validating ML models.
* Predicting the accuracy in the provided algorithms in calculating AQI index value.
* Classifying Accuracy and comparatively analysing ML models.

**5.2 DATA FLOW DIAGRAMS**

1. A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. The DFD is also called as a data flow graph or bubble chart.
2. A neat and clear DFD can depict the right amount of the system requirement graphically.
3. Data Flow Diagram represent detailed and well explained diagram of system components. It shows how data enters and leaves the system, what changes the information, and where data is stored.
4. The objective of a DFD is to show the scope and boundaries of a system as a whole.
5. It is used as the part of system documentation file. Data Flow Diagrams can be understood by both technical or nontechnical person because they are very easy to understand.

**5.3 UML DIAGRAMS**

Unified Modelling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed and is used to specify, construct, and document the artifacts (major elements) of the software system, as well as for business modelling and other non-software systems.  UML is a simple modelling approach that is used to model all the practical systems. It makes it easy to understand the objects and how they interact with each other.  A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system. There are two broad categories of diagrams and they are again divided into subcategories −

* Structural Diagrams
* Behavioural Diagrams

**5.3.1 Structural Diagrams**

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable. These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are –

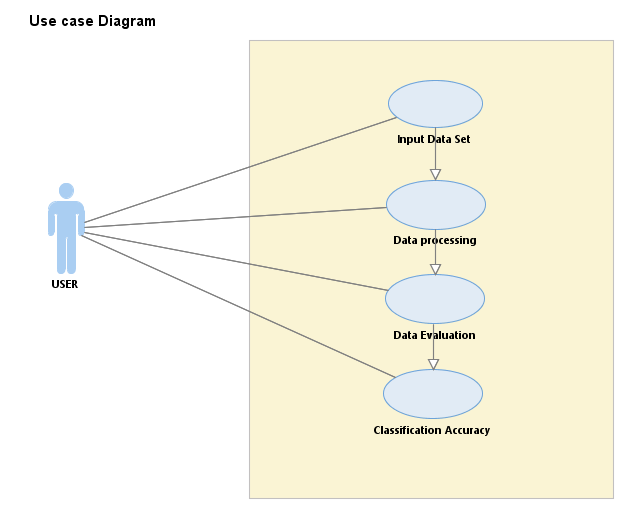
* Class diagram
* Object diagram
* Component diagram
* Deployment diagram

**5.3.2 Behavioural Diagrams**

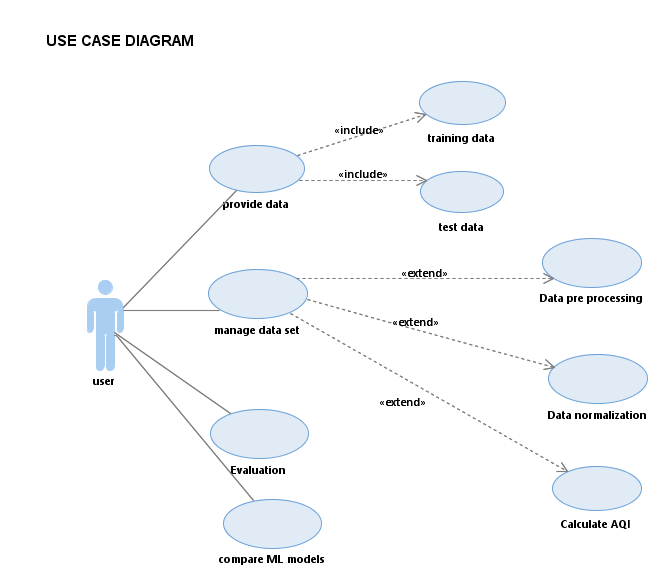
Any system can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered. Behavioural diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system. Types of behavioural diagrams −

* Use case diagram
* Sequence diagram
* Collaboration diagram
* State chart diagram
* Activity diagram

**USE CASE DIAGRAM**

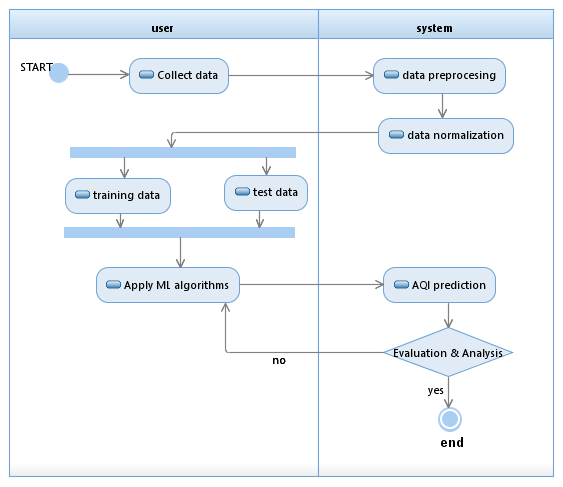


**Fig.4: use case diagram showing flow of data process**

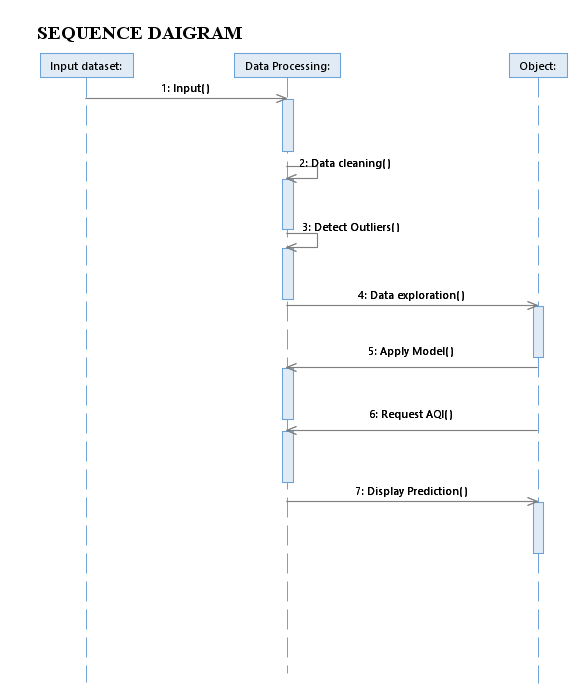


**Fig.5: Domain Use Case Diagram**

**ACTIVITY DIAGRAM**



**Fig.6: Activity Diagram for Predicting AQI**



**Fig.7: Sequence Diagram for Predicting AQI**

**CHAPTER-6**

**IMPLEMENTATION OF CODE**

**#importing necessary libraries**

**#ln[1]:**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings("ignore")

**#ln[2]:**

!pip install sklearn

**#ln[3]:**

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn import metrics

from sklearn.metrics import mean\_absolute\_error,mean\_squared\_error,r2\_score

from sklearn.metrics import accuracy\_score,confusion\_matrix

**#ln[4]:**

from google.colab import drive

**#ln[5]:**

drive.mount('/content/drive')

**#ln[6]:**

print(pd.\_\_file\_\_)

**#ln[7]:**

!pip install pandas

**#ln[8]:**

import pandas

**#ln[9]:**

df=pandas.read\_csv(r'/content/drive/MyDrive/data.csv',encoding='unicode\_escape')

**#Reading data**

**#ln[10]:**

df.head()

**#ln[11]:**

df.shape

**#ln[12]:**

df.info()

**#Checking the overall information on the dataset**

**#ln[13]:**

df.isnull().sum()

**#ln[14]:**

df.describe()

**#ln[15]:**

df.nunique()

**#ln[16]:**

df.columns

**# Data Visualization**

**#ln[17]:**

!pip install seaborn

**#ln[18]:**

import seaborn as sns

**#ln[19]:**

sns.pairplot(data=df)

**#ln[20]:**

df['state'].value\_counts()

**#ln[21]:**

plt.figure(figsize=(15,6))

plt.xticks(rotation=90)

df.state.hist()

plt.xlabel('staate')

plt.ylabel('Frequencies')

plt.plot()

**#ln[22]:**

df['type'].value\_counts()

**#ln[23]:**

plt.figure(figsize=(15, 6))

plt.xticks(rotation=90)

df.type.hist()

plt.xlabel('Type')

plt.ylabel('Frequencies')

plt.plot()

**#ln[24]:**

df['agency'].value\_counts()

**#ln[25]:**

plt.figure(figsize=(15, 6))

plt.xticks(rotation=90)

df.agency.hist()

plt.xlabel('Agency')

plt.ylabel('Frequencies')

plt.plot()

**#ln[26]:**

plt.figure(figsize=(30, 10))

plt.xticks(rotation=90)

sns.barplot(x='state',y='so2',data=df);

**#ln[27]:**

plt.rcParams['figure.figsize']=(30,10)

**#ln[28]:**

df[['so2','state']].groupby(["state"]).mean().sort\_values(by='so2').plot.bar(color='purple')

plt.show()

**#ln[29]:**

plt.figure(figsize=(30, 10))

plt.xticks(rotation=90)

sns.barplot(x='state',y='no2',data=df);

**#ln[30]:**

df[['no2','state']].groupby(["state"]).mean().sort\_values(by='no2').plot.bar(color='purple')

plt.show()

**#ln[31]:**

plt.figure(figsize=(30, 10))

plt.xticks(rotation=90)

sns.barplot(x='state',y='rspm',data=df);

**#ln[32]:**

plt.figure(figsize=(30, 10))

plt.xticks(rotation=90)

sns.barplot(x='state',y='spm',data=df);

**#ln[33]:**

plt.figure(figsize=(30, 10))

plt.xticks(rotation=90)

sns.barplot(x='state',y='pm2\_5',data=df);

**Checking all null values and treating those null values**

**#ln[34]:**

nullvalues = df.isnull().sum().sort\_values(ascending=False)

nullvalues

**#ln[35]:**

null\_values\_percentage = (df.isnull().sum()/df.isnull().count()\*100).sort\_values(ascending=False)

missing\_data\_with\_percentage = pd.concat([nullvalues, null\_values\_percentage], axis=1, keys=['Total', 'Percent'])

missing\_data\_with\_percentage

**#ln[36]:**

df.drop(['agency'],axis=1,inplace=True)

df.drop(['stn\_code'],axis=1,inplace=True)

df.drop(['date'],axis=1,inplace=True)

df.drop(['sampling\_date'],axis=1,inplace=True)

df.drop(['location\_monitoring\_station'],axis=1,inplace=True)

**#ln[37]:**

df.isnull().sum()

**#ln[38]:**

df

**#ln[39]:**

df['location']=df['location'].fillna(df['location'].mode()[0])

df['type']=df['type'].fillna(df['type'].mode()[0])

**#ln[40]:**

df.fillna(0, inplace=True)

**#ln[41]:**

df.isnull().sum()

**#ln[42]:**

Df

**# CALCULATE AIR QUALITY INDEX BASED ON FORMULA**

**Function to calculate so2 individual pollutant index**

**#ln[43]:**

def cal\_SOi(so2):

si=0

if (so2<=40):

si= so2\*(50/40)

elif (so2>40 and so2<=80):

si= 50+(so2-40)\*(50/40)

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

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

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

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

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

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

elif (so2>1600):

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

return si

df['SOi']=df['so2'].apply(cal\_SOi)

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

data.head()

**Function to calculate no2 individual pollutant index**

**#ln[44]:**

def cal\_Noi(no2):

ni=0

if(no2<=40):

ni= no2\*50/40

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

ni= 50+(no2-40)\*(50/40)

elif(no2>80 and no2<=180):

ni= 100+(no2-80)\*(100/100)

elif(no2>180 and no2<=280):

ni= 200+(no2-180)\*(100/100)

elif(no2>280 and no2<=400):

ni= 300+(no2-280)\*(100/120)

else:

ni= 400+(no2-400)\*(100/120)

return ni

df['Noi']=df['no2'].apply(cal\_Noi)

data= df[['no2','Noi']]

data.head()

**Function to calculate rspm individual pollutant index**

**#ln[45]:**

def cal\_RSPMI(rspm):

rpi=0

if(rpi<=30):

rpi=rpi\*50/30

elif(rpi>30 and rpi<=60):

rpi=50+(rpi-30)\*50/30

elif(rpi>60 and rpi<=90):

rpi=100+(rpi-60)\*100/30

elif(rpi>90 and rpi<=120):

rpi=200+(rpi-90)\*100/30

elif(rpi>120 and rpi<=250):

rpi=300+(rpi-120)\*(100/130)

else:

rpi=400+(rpi-250)\*(100/130)

return rpi

df['Rpi']=df['rspm'].apply(cal\_RSPMI)

data= df[['rspm','Rpi']]

data.head()

**Function to calculate spm individual pollutant index**

**#ln[46]:**

def cal\_SPMi(spm):

spi=0

if(spm<=50):

spi=spm\*50/50

elif(spm>50 and spm<=100):

spi=50+(spm-50)\*(50/50)

elif(spm>100 and spm<=250):

spi= 100+(spm-100)\*(100/150)

elif(spm>250 and spm<=350):

spi=200+(spm-250)\*(100/100)

elif(spm>350 and spm<=430):

spi=300+(spm-350)\*(100/80)

else:

spi=400+(spm-430)\*(100/430)

return spi

df['SPMi']=df['spm'].apply(cal\_SPMi)

data= df[['spm','SPMi']]

data.head()

**Function to calculate the air quality(AQI) of every data value**

**#ln[47]:**

def cal\_aqi(si,ni,rspmi,spmi):

aqi=0

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

aqi=si

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

aqi=ni

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

aqi=rspmi

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

aqi=spmi

return aqi

df['AQI']=df.apply(lambda x:cal\_aqi(x['SOi'],x['Noi'],x['Rpi'],x['SPMi']),axis=1)

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

data.head()

**#ln[48]:**

def AQI\_Range(x):

if x<=50:

return "Good"

elif x>50 and x<=100:

return "Moderate"

elif x>100 and x<=200:

return "Poor"

elif x>200 and x<=300:

return "Unhealthy"

elif x>300 and x<=400:

return "Very unhealthy"

elif x>400:

return "Hazardous"

df['AQI\_Range'] = df['AQI'] .apply(AQI\_Range)

df.head()

**#ln[49]:**

df['AQI\_Range'].value\_counts()

**Splitting the dataset into Dependent and Independent columns**

**#ln[50]:**

X=df[['SOi','Noi','Rpi','SPMi']]

Y=df['AQI']

X.head()

**#ln[51]:**

Y.head()

**#ln[52]:**

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)

**# LINEAR REGRESSION**

**#ln[53]:**

model=LinearRegression()

model.fit(X\_train,Y\_train)

**#ln[54]:**

#predicting train

train\_pred=model.predict(X\_train)

#predicting on test

test\_pred=model.predict(X\_test)

**#ln[55]:**

RMSE\_train=(np.sqrt(metrics.mean\_squared\_error(Y\_train,train\_pred)))

RMSE\_test=(np.sqrt(metrics.mean\_squared\_error(Y\_test,test\_pred)))

print("RMSE TrainingData = ",str(RMSE\_train))

print("RMSE TestData = ",str(RMSE\_test))

print('-'\*50)

print('RSquared value on train:',model.score(X\_train, Y\_train))

print('RSquared value on test:',model.score(X\_test, Y\_test))

**# DECISION TREE REGRESSOR**

**#ln[56]:**

DT=DecisionTreeRegressor()

DT.fit(X\_train,Y\_train)

**#ln[57]:**

#predicting train

train\_preds=DT.predict(X\_train)

#predicting on test

test\_preds=DT.predict(X\_test)

**#ln[58]:**

RMSE\_train=(np.sqrt(metrics.mean\_squared\_error(Y\_train,train\_preds)))

RMSE\_test=(np.sqrt(metrics.mean\_squared\_error(Y\_test,test\_preds)))

print("RMSE TrainingData = ",str(RMSE\_train))

print("RMSE TestData = ",str(RMSE\_test))

print('-'\*50)

print('RSquared value on train:',DT.score(X\_train, Y\_train))

print('RSquared value on test:',DT.score(X\_test, Y\_test))

**# RANDOM FOREST REGRESSOR**

**#ln[59]:**

RF=RandomForestRegressor().fit(X\_train,Y\_train)

**#ln[60]:**

#predicting train

train\_preds1=RF.predict(X\_train)

#predicting on test

test\_preds1=RF.predict(X\_test)

**#ln[61]:**

RMSE\_train=(np.sqrt(metrics.mean\_squared\_error(Y\_train,train\_preds1)))

RMSE\_test=(np.sqrt(metrics.mean\_squared\_error(Y\_test,test\_preds1)))

print("RMSE TrainingData = ",str(RMSE\_train))

print("RMSE TestData = ",str(RMSE\_test))

print('-'\*50)

print('RSquared value on train:',RF.score(X\_train, Y\_train))

print('RSquared value on test:',RF.score(X\_test, Y\_test))

**# CLASSIFICATION ALGORITHMS**

**#ln[62]:**

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

**#ln[63]:**

X2 = df[['SOi','Noi','Rpi','SPMi']]

Y2 = df['AQI\_Range']

**#ln[64]:**X\_train2, X\_test2, Y\_train2, Y\_test2 = train\_test\_split(X2, Y2, test\_size=0.33, random\_state=70)

**# LOGISTIC REGRESSION**

**#ln[65]:**

**#fit the model on train data**

log\_reg = LogisticRegression().fit(X\_train2, Y\_train2)

**#predict on train**

train\_preds2 = log\_reg.predict(X\_train2)

**#accuracy on train**

print("Model accuracy on train is: ", accuracy\_score(Y\_train2, train\_preds2))

**#predict on test**

test\_preds2 = log\_reg.predict(X\_test2)

**#accuracy on test**

print("Model accuracy on test is: ", accuracy\_score(Y\_test2, test\_preds2))

print('-'\*50)

**# Kappa Score.**

print('KappaScore is: ', metrics.cohen\_kappa\_score(Y\_test2,test\_preds2))

**#ln[66]:**

log\_reg.predict([[727,327.55,78.2,100]])

**#ln[67]:**

log\_reg.predict([[2.7,45,35.16,23]])

**#ln[68]:**

log\_reg.predict([[10,2.8,82,20]])

**#ln[69]:**

log\_reg.predict([[2,45.8,37,32]])

**# DECISION TREE CLASSIFIER**

**#ln[70]:**

**#fit the model on train data**

DT2 = DecisionTreeClassifier().fit(X\_train2,Y\_train2)

**#predict on train**

train\_preds3 = DT2.predict(X\_train2)

**#accuracy on train**

print("Model accuracy on train is: ", accuracy\_score(Y\_train2, train\_preds3))

**#predict on test**

test\_preds3 = DT2.predict(X\_test2)

**#accuracy on test**

print("Model accuracy on test is: ", accuracy\_score(Y\_test2, test\_preds3))

print('-'\*50)

**# Kappa Score**

print('KappaScore is: ', metrics.cohen\_kappa\_score(Y\_test2,test\_preds3))

**# RANDOM FOREST CLASSIFIER**

**#ln[71]:**

**#fit the model on train data**

RF=RandomForestClassifier().fit(X\_train2,Y\_train2)

**#predict on train**

train\_preds4 = RF.predict(X\_train2)

**#accuracy on train**

print("Model accuracy on train is: ", accuracy\_score(Y\_train2, train\_preds4))

**#predict on test**

test\_preds4 = RF.predict(X\_test2)

**#accuracy on test**

print("Model accuracy on test is: ", accuracy\_score(Y\_test2, test\_preds4))

print('-'\*50)

**# Kappa Score**

print('KappaScore is: ', metrics.cohen\_kappa\_score(Y\_test2,test\_preds4))

**# K-NEAREST NEIGHBOURS**

**#ln[72]:**

**#fit the model on train data**

KNN = KNeighborsClassifier().fit(X\_train2,Y\_train2)

**#predict on train**

train\_preds5 = KNN.predict(X\_train2)

**#accuracy on train**

print("Model accuracy on train is: ", accuracy\_score(Y\_train2, train\_preds5))

**#predict on test**

test\_preds5 = KNN.predict(X\_test2)

**#accuracy on test**

print("Model accuracy on test is: ", accuracy\_score(Y\_test2, test\_preds5))

print('-'\*50)

**# Kappa Score**

print('KappaScore is: ', metrics.cohen\_kappa\_score(Y\_test2,test\_preds5))

**# Predictions on random values**

**#ln[73]:**

KNN.predict([[7.4,47.7,78.182,100]])

**#ln[74]:**

KNN.predict([[1,1.2,3.12,0]])

**#ln[75]:**

KNN.predict([[325.7,345,798.182,203]])

**CHAPTER-7**

**TESTING**

**7.1 INTRODUCTION**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**7.1.1 TESTING OBJECTIVES**

All field entries must work properly. Pages must be activated from the identified link. The entry screen, messages and responses must not be delayed. Testing cannot show the absence of defects, it can only show that software errors are present. Once code has been generated the testing begins.

The purpose of testing is more than just debugging and detecting of bugs. Testing is usually performing for the following:

For improving and assuring software quality.

For estimation reliability.

For verification and validation.

The objection that should kept in mind while testing is being executed as follows:

It should be easily predictable.

It should be fixed.

It should follow certain constraints a rule.

**7.1.2 TESTING APPROACHES**

Field testing will be performed manually and functional tests will be written in detail.

The different types of Tests are as follows:

1) Unit Testing

2) Integration Testing

3) Functional Test

4) System Test

5) White Box Testing

6) Black Box Testing

7) Acceptance Testing

**7.2 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**7.3 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from thecombination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level interact without error.

**7.4 FUNCTIONAL TEST**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.5 SYSTEM TEST**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**7.6 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.7 BLACK BOX TESTING**

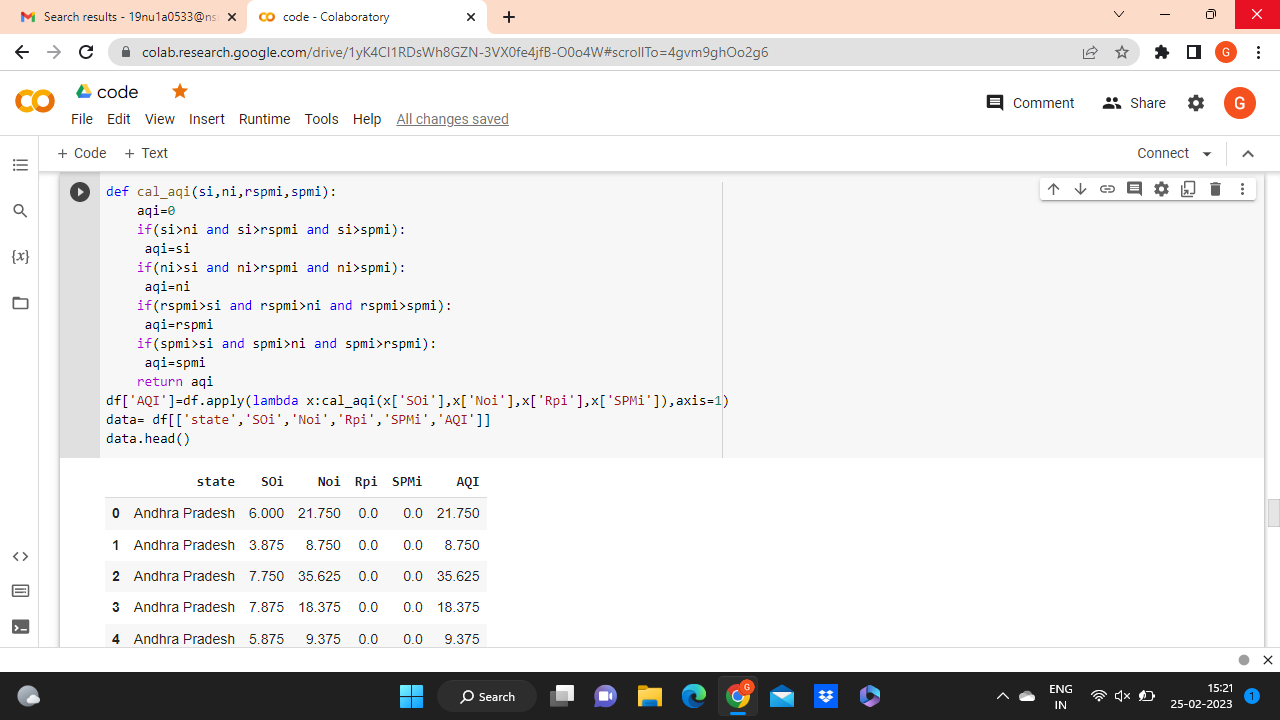
Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box, you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**7.8 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

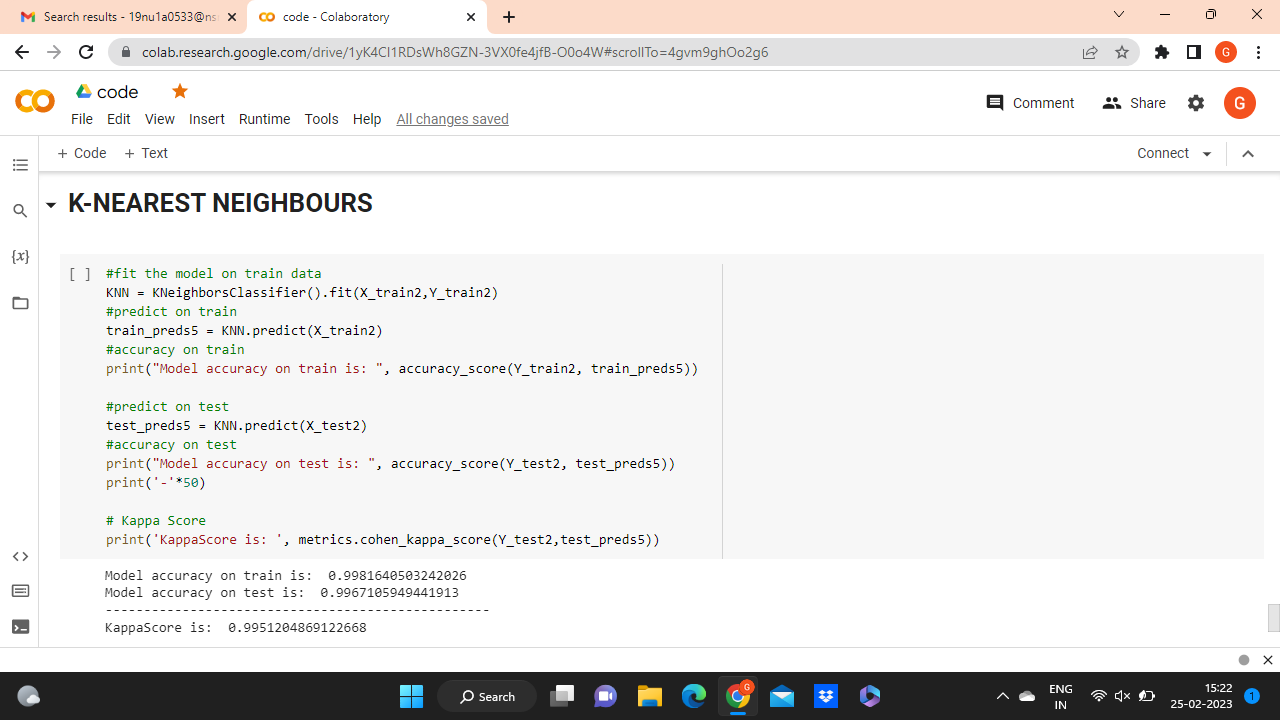
**CHAPTER-8**

**SCREENSHOTS**

****

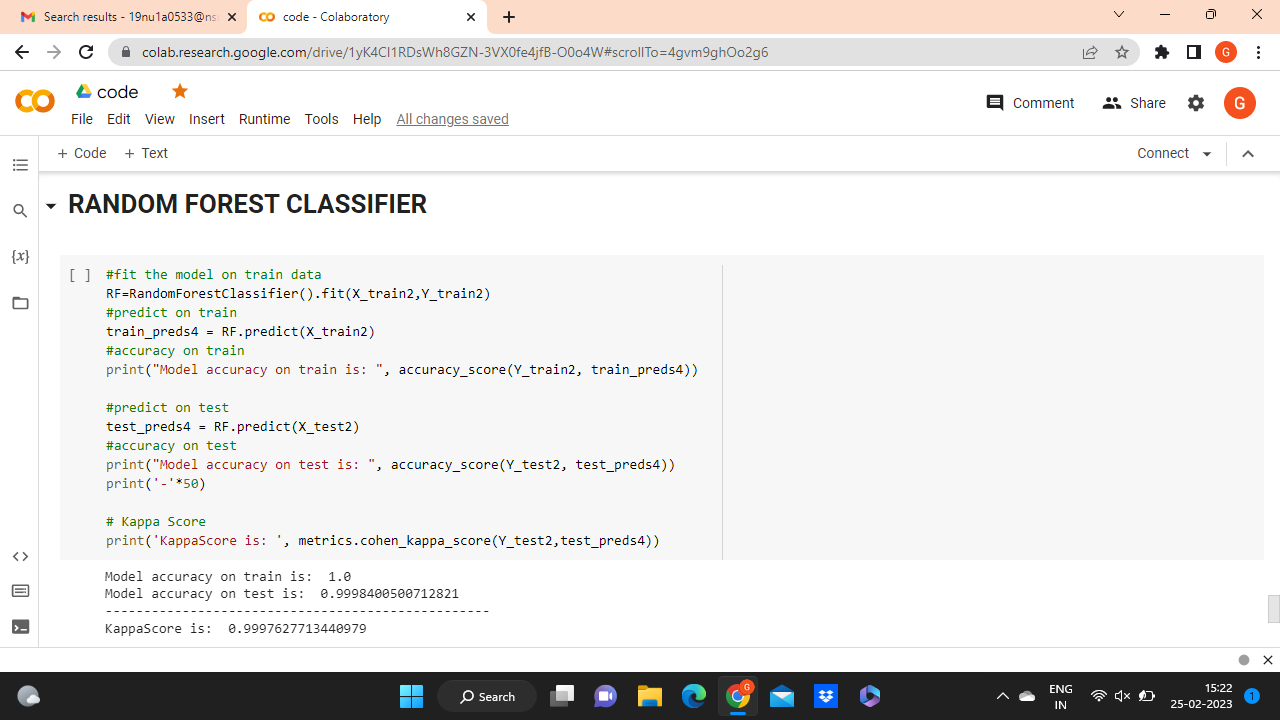
**Description**

Function to calculate the air quality (AQI) of every data value.

****

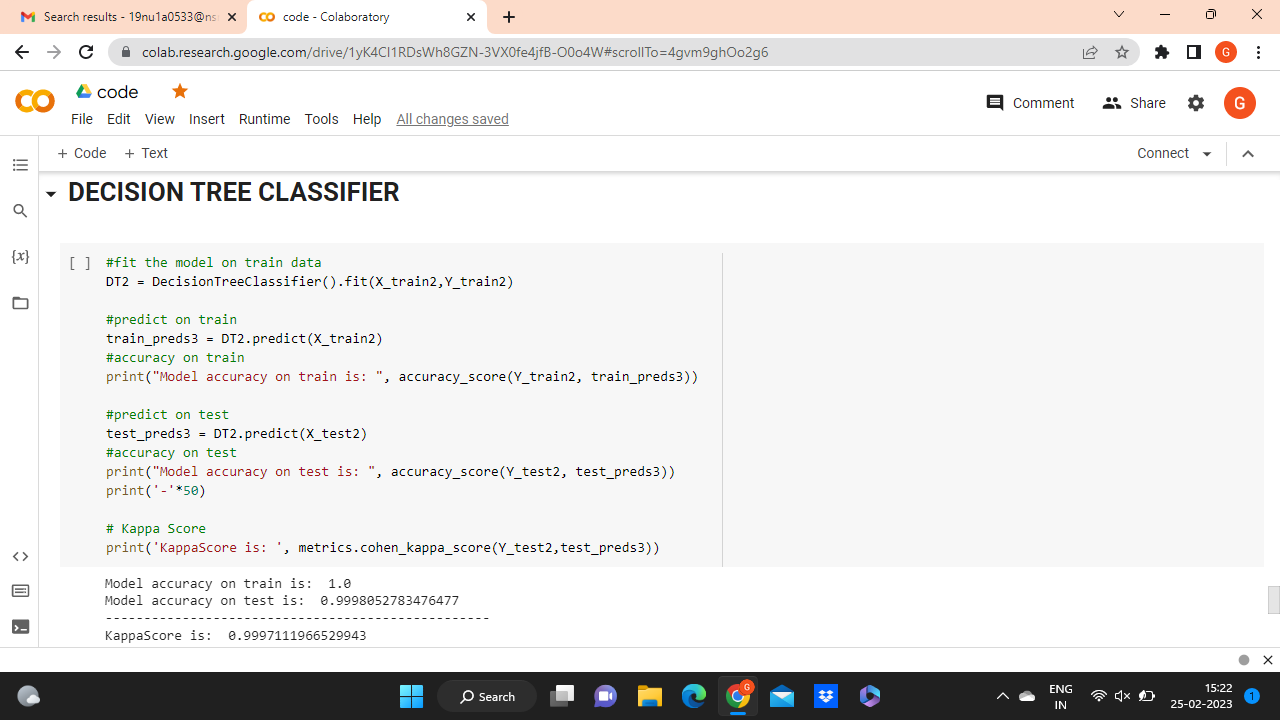
**Description**

Using k-nearest neighbour algorithm the performance measure we approached called KappaScore is: 0.9951204869122668

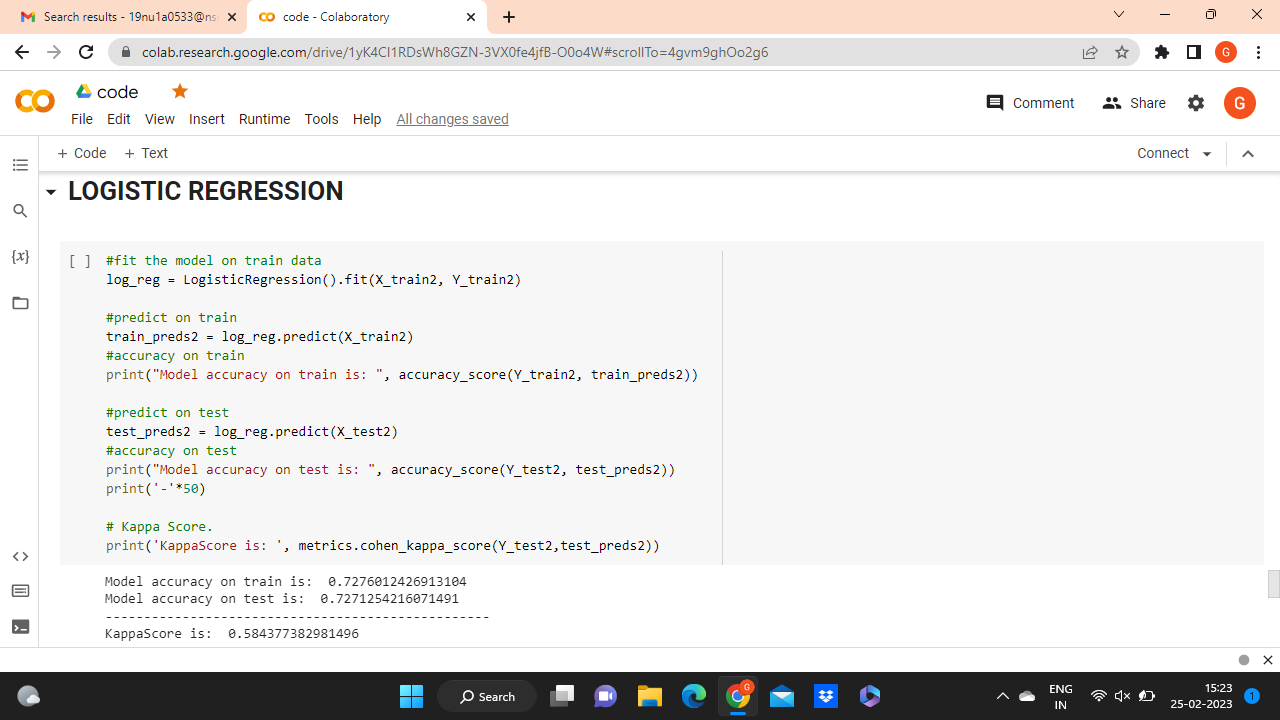
****

**Description**

Using Random Forest Classifier algorithm the performance measure we approached called KappaScore is: 0.9997627713440979

** Description**

Using Decision tree classifier algorithm the performance measure we approached called KappaScore is: 0.9997111966529943

****

**Description**

Using logistic regression algorithm, the performance measure we approached called KappaScore is :0.584377382981496**.**

**CHAPTER-9**

**CONCLUSION AND FUTURE ENHANCEMENT**

Air pollution is harmful to both the environment and human existence. When some substances in the atmosphere exceed a certain concentration, it results in air pollution.

In this project, we started with downloading the data set, pre-processing it, created the model and found out the predictions using the models. This data set includes common air pollutants that cause air pollution such as Sulphur dioxide(SO2), Nitrogen oxides(NO2), RSPM( Respirable Suspended Particulate Matter), SPM (Suspended Particulate Matter) etc. The pollutants are predicted using the proposed models i.e, Decision tree classifier, Random forest classifier, KNN (K-nearest neighbour), logistics regression which results in providing best accurate, efficient prediction.

In future, this model can be used by government agencies to alert people about the quality of the air and also practice forecasting it. Finally, it is concluded that this research is very useful for the society since forecasting air quality levels acts as an important tool to prevent air pollution by taking necessary actions and steps to control the pollutant.

**CHAPTER-10**

**REFERENCES**

[1] Bellinger C, Jabbar MSM, Zaïane O, Osornio-Vargas A (2017) A systematic review of data mining and machine learning for air pollution epidemiology. BMC Public Health.

[2] Bhalgat P, Bhoite S, Pitare S (2019) Air Quality Prediction using Machine Learning Algorithms. Int J Comput Appl Technol Res 8(9):367–370.

[3] Castelli M, Clemente FM, Popoviˇc A, Silva S, Vanneschi L (2020) A machine learning approach to predict air quality in California. Complexity 2020(8049504):1–23.

[4] Doreswamy HKS, Yogesh KM, Gad I (2020) Forecasting Air pollution particulate matter (PM2.5) using machine learning regression models. Procedia Comput Sci 171:2057–2066.

[5] Gopalakrishnan V (2021) Hyperlocal air quality prediction using machine learning. Towards data science.

[6] Gurjar BR (2021) Air pollution in india: major issues and challenges. energy future 9(2):12–27.

[7] Liang Y, Maimury Y, Chen AH, Josue RCJ (2020) Machine learning-based prediction of air quality. Appl Sci 10(9151):1–17.

[8] Madan T, Sagar S, Virmani D (2020) Air quality prediction using machine learning algorithms–a review. In: 2nd international conference on advances in computing, communication control and networking (ICACCCN) pp 140–145.

[9] Madhuri VM, Samyama GGH, Kamalapurkar S (2020) Air pollution prediction using machine learning supervised learning approach. Int J Sci Technol Res 9(4):118–123

[10] Mahalingam U, Elangovan K, Dobhal H, Valliappa C, Shrestha S, Kedam G (2019) A machine learning model for air quality prediction for smart cities. In: 2019 international conference on wireless communications signal processing and networking (WiSPNET). IEEE 452–457.

[11] Monisri PR, Vikas RK, Rohit NK, Varma MC, Chaithanya BN (2020) Prediction and analysis of air quality using machine learning. Int J Adv Sci Technol 29(5):6934–6943

[12] Nahar K, Ottom MA, Alshibli F, Shquier MA (2020) Air quality index using machine learning—a jordan case study. COMPUSOFT, Int J Adv Comput Technol 9(9):3831–3840

[13] Patil RM, Dinde HT, Powar SK (2020) A literature review on prediction of air quality index and forecasting ambient air pollutants using machine learning algorithms 5(8):1148–1152

[14] Sanjeev D (2021) Implementation of machine learning algorithms for analysis and prediction of air quality. Int. J. Eng. Res. Technol. 10(3):533–538

[15] Soundari AG, Jeslin JG, Akshaya AC (2019) Indian air quality prediction and analysis using machine learning. Int J Appl Eng Res 14(11):181–186

[16] Sweileh WM, Al-Jabi SW, Zyoud SH, Sawalha AF (2018) Outdoor air pollution and respiratory health: a bibliometric analysis of publications in peer-reviewed journals (1900–2017). Multidiscip Respiratory Med.

[17] Zhu D, Cai C, Yang T, Zhou X (2018) A machine learning approach for air quality prediction: model regularization and optimization. Big Data and Cognitive Comput.