

PREDICTION OF AIR QUALITY USING SUPERVISED MACHINE LEARNING APPROACH

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

SWETHA SRINIVASAN (Reg. No. 37110762)

NIKITHA REDDY (Reg. No. 37110781)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119
APRIL – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **SWETHA SRINIVASAN (Reg. No. 37110762)** and **NIKITHA REDDY (Reg. No.37110781)** who carried out the project entitled "**PREDICTION OF AIR QUALITY USING SUPERVISED MACHINE LEARNING APPROACH**" under my supervision from **August 2019 to March 2020**.

Internal Guide

Dr. P.AJITHA, M.E., PH.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **SWETHA SRINIVASAN** and **NIKITHA REDDY** hereby declare that the Project Report entitled **“PREDICTION OF AIR QUALITY USING SUPERVISED MACHINE LEARNING APPROACH”** is done by me under the guidance of **Dr. P. AJITHA, M.E.,Ph.D., Department of Computer Science and Engineering** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAM A** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. P. AJITHA, M.E., Ph.d., Associate Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Generally, Air pollution refers to the release of pollutants into the air that are detrimental to human health and the planet as a whole. It can be described as one of the most dangerous threats that the humanity ever faced. It causes damage to animals, crops, forests etc. To prevent this problem in transport sectors have to predict air quality from pollutants using machine learning techniques. Hence, air quality evaluation and prediction has become an important research area. The aim is to investigate machine learning based techniques for air quality forecasting by prediction results in best accuracy. The analysis of dataset by supervised machine learning technique(SMLT) to capture several information's like, variable identification, uni-variate analysis, bi-variate and multi-variate analysis, missing value treatments and analyze the data validation, data cleaning/preparing and data visualization will be done on the entire given dataset. Our analysis provides a comprehensive guide to sensitivity analysis of model parameters with regard to performance in prediction of air quality pollution by accuracy calculation. To propose a machine learning-based method to accurately predict the Air Quality Index value by prediction results in the form of best accuracy from comparing supervise classification machine learning algorithms. Additionally, to compare and discuss the performance of various machine learning algorithms from the given transport traffic department dataset with evaluation of GUI based user interface air quality prediction by attributes.

Keywords: dataset, Machine learning-Classification method, python, GUI results.

TABLE OF CONTENTS

ABSTRACT	V
LIST OF FIGURES	XI

CHAPTER	TITLE	PAGE NO
1	INTRODUCTION	1
	1.1 OBJECTIVE	2
	1.2 PROBLEM DESCRIPTION/PROBLEM STATEMENT	2
	1.3 DOMAIN OVERVIEW	3
	1.4 AIR POLLUTION CAUSES	6
	1.5 SCOPE	6
2	LITERATURE SURVEY	7
	2.1 APPLICATION OF BP NEURAL NETWORK	7
	2.2 AIR QUALITY PREDICTION IN URBAN	7
	2.3 REVENUE ORIENTED AIR QUALITY PREDICTION	8
	2.4 BIG DATA PLATFORM FOR AIR quality	9
	2.5 PREDICTION OF AIR QUALITY INDEX	10
	2.6 PREDICTION OF AIR QUALITY INDEX	

2.7 DATA-DRIVEN PREDICTION MODEL

	OF INDOOR AIR QUALITY	11
	2.8 PREDICTIVE DATA FEATURE	
	EXPLORATION-BASED AIR QUALITY	11
	2.9 SPATIO-TEMPORAL-AWARE SPARSE	12
	2.10 A SEQUENCE-TO-SEQUENCE AIR QUALITY	13
3	SYSTEM DESIGN & METHODOLOGY	14
	3.1 SYSTEM	14
	3.2 DRAWBACKS	15
	3.3 PROPOSED SYSTEM	15
	3.4 LOGISTIC REGRESSION	16
	3.5 RANDOM FOREST	17
4	SOFTWARE DEVELOPMENT METHODOLOGY	19
	4.1 SYSTEM SPECIFICATION	19
	4.2 ANACONDA NAVIGATOR	19
	4.3 LANGUAGES USED IN THE SYSTEM	20
	4.3.1 ABOUT PYTHON LANGUAGE	20
	4.4 METHODOLOGY	24
	4.4.1 SYSTEM DESIGN	24
	4.5 CONSTRUCTION OF A PREDICTIVE MODEL	25
	4.6 UML DIAGRAMS	26

	4.7 USECASE DIAGRAM	27
	4.8 MODEL DESCRIPTION	30
	4.9 TESTING	32
	4.9.1 SOFTWARE TESTING	32
	4.9.2 TEST CASE	33
	4.9.3 TESTING TECHNIQUES	33
5	RESULTS AND DISCUSSION	34
6	CONCLUSION AND FUTURE WORK	36
	REFERENCES	37
	APPENDIX	39

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	PROCESS OF LEARNING MACHINE	4
4.1	ARCHITECTURE	25
4.2	USECASE	28
4.3	SEQUENCE DIAGRAM	28
4.4	ENTITY RELATIONSHIP DIAGRAM	29
5.1	FORM	36
5.2	PREDICTED RESULT	37

CHAPTER 1

INTRODUCTION

As the largest growing industrial nation, India is producing record amount of pollutants specifically Co₂, pm_{2.5} etc and other harmful aerial contaminants. Air quality of a particular state or a country is a measure on the effect of pollutants on the respected regions, as per the Indian air quality standard pollutants are indexed in terms of their scale, these air quality indexes indicates the levels of major pollutants on the atmosphere. There are various atmospheric gases which causes pollution on our environment. Each pollution has individual index and scales at different levels. The major pollutants Such as (no₂, so₂, rspm, spm) indexes AQI is acquired, with this individual AQI, the data can be categorized based on the limits. We collected the data from the Indian government database, which contains pollutant concentration occurring at various places across India. We start by calculating the individual index of the pollutant for every available datapoints and find their respective AQI for the region. We have designed a model to predict the air quality index of every available data points in the dataset, our model is capable of forecasting the air quality of India in any given area. By predicting the air quality index, we can backtrack the major pollution causing pollutant and the location affected seriously by the pollutant across India. With this forecasting model, various knowledge about the data are extracted using various techniques to obtain heavily affected regions on a particular region(cluster). This give more information and knowledge about the cause and seniority of the pollutants.

1.1 OBJECTIVE

Observing and retaining high standard air has become the crucial challenge in metropolitan areas which have more industries, companies and population. As there is a rise in population, there is an increase in the transportation, usage of electricity and fuels. There is a lot of waste dumped in the land which we are well aware of. The air is also highly contaminated which causes a more serious threat to all kinds of living organisms in the earth. This gives rise to the need for monitoring and assessing the quality of air and accordingly the government should be given alert to take necessary actions. This research work concentrates on performing an effective analysis on all the major works done in this aspect using machine learning algorithms.

1.2 Problem Description/ Problem Statements

Monitoring and preserving air quality has become one of the most essential activities in many industrial and urban areas today. The quality of air is adversely affected due to various forms of pollution caused by transportation, electricity, fuel uses etc. The deposition of harmful gases is creating a serious threat for the quality of life in smart cities. With increasing air pollution, we need to implement efficient air quality monitoring models which collect information about the concentration of air pollutants and provide assessment of air pollution in each area.

1.3 Domain overview

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data

and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm

uses this training data to give predictions on a new test data. Machine learning can be roughly separated into three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It is provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.



Fig 1.1: Process of Machine learning

Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where we have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from

the input to the output is $y = f(X)$. The goal is to approximate the mapping function so

well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include logistic regression, multi-class classification, Decision Trees and support vector machines etc. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into Classification problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for regression and categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A classification problem is when the output variable is a category, such as “red” or “blue”.

Air pollution refers to the release of pollutants into the air that are detrimental to human health and the planet as a whole. It can be described as one of the most dangerous threats that the humanity ever faced. It causes damage to animals, crops and forests. It also contributes to the depletion of the ozone layer, which protects the Earth from the sun's UV rays. Some of the other environmental effects of air pollution are haze, eutrophication, and global climate changes. Most air pollution comes from energy use and production. Burning fossil fuels releases gases and chemicals into the air. And in an especially destructive feedback loop, air pollution not only contributes to climate change but is also exacerbated by it. Air pollution in the form of carbon dioxide and methane raises the earth's temperature. Another type of air pollution is then worsened by that increased heat: Smog forms when the weather is warmer and there's more ultraviolet radiation. Climate change also increases the production of allergenic air pollutants including mold (thanks to damp conditions caused by extreme weather and increased flooding) and pollen (due to a longer pollen season and more pollen production). Air pollution can be defined as an alteration of air quality that can be characterized by measurements of chemical, biological or physical pollutants in the air. Therefore, air pollution means the undesirable presence of impurities or the abnormal rise in the

proportion of some

constituents of the atmosphere. It can be classified in 2 sections: visible and invisible air pollution.

1.4 Air pollution causes:

Air pollution is caused by the presence in the atmosphere of toxic substances, mainly produced by human activities, even though sometimes it can result from natural phenomena such as volcanic eruptions, dust storms and wildfires, also depleting the air quality.

Anthropogenic air pollution sources are:

- Combustion of fossil fuels, like coal and oil for electricity and road transport, producing air pollutants like nitrogen and sulfur dioxide.
- Emissions from industries and factories, releasing large amount of carbon monoxide, hydrocarbon, chemicals and organic compounds into the air.
- Agricultural activities, due to the use of pesticides, insecticides, and fertilizers that emit harmful chemicals.
- Waste production, mostly because of methane generation in landfills.

1.5 Scope

The scope of this project is to investigate a dataset of air pollutants records for India meteorological sector using machine learning technique. To identifying air quality is more difficult and try to reduce this risk factor behind predicting from Air Quality Index (AQI) of India to save human so as to save lots of meteorological efforts and assets and to predict whether assigning the air quality is bad or good based on given attributes.

CHAPTER 2

LITERATURE

SURVEY

1. Application of BP neural network optimized by genetic simulated annealing algorithm to prediction of air quality index in Lanzhou

Authors: Zhou Kang ; Zhiyi Qu, 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)

It is of great significance to carry out cities' air quality forecasting work for the prevention of the air pollution in urban areas and to the improvement of the living environment of urban residents. The air quality index (AQI) is a dimensionless index that quantitatively describes the state of air quality. In this paper, the data of air quality in Lanzhou released by china air quality online monitoring and analysis platform is dealt with, and then AQI prediction model based on back propagation (BP) neural network, AQI prediction model based on genetic algorithm optimization and AQI prediction model of BP neural network based on genetic simulated annealing algorithm optimization are established. By comparing and analyzing the prediction results, it is found that BP neural network based on genetic simulated annealing algorithm has strong generalization ability and global search ability, and has higher accuracy rate.

2. Air Quality Prediction Method in Urban Residential Area

Authors: RuiJun Yang ; HaiLong Zhou ; DanFeng Ding, 2018 11th International Symposium on Computational Intelligence and Design (ISCID)

According to hedonic price theory, using classification algorithm of the machine learning which include SVM, Naive Bayesian and KNN, establish the internal mapping relationship between feature variables that include prices of housing and air quality. Air quality in urban residential district is predicted by feature variables. Finally, experiments have conducted on the data set of residential district In Tianhe, Guangzhou city, showed that the most accuracy. The method is very practical and stable for air quality in urban residential district prediction. The new method has a strong practical value as reference methods for buyers.

3. Revenue oriented air quality prediction microservices for smart cities

Authors: Shajulin Benedict, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)

A notion of creating smart cities with integrated high end ICT technologies, including cloud, edge, IoTs, and so forth, has increased recently in order to provide a reasonably high comfort to city residents with limited human intervention. In fact, one of the major challenges faced by city authorities of developing countries is to control air pollution, which might cause chronic diseases and financial concerns to the residents. Although a few air quality monitoring and air pollution control measures exist, these solutions are not remaining as assets to cities. In this paper, a revenue oriented, cloud microservice based air quality prediction framework is proposed for smart cities. This framework collects the air quality parameters such as SO₂, NO₂, RSPM, and so forth, for different cities in a mongodb based cloud database; it develops air quality prediction models based on Random Forests (RF); and it predicts the air quality parameter values accordingly. For each access to the proposed air quality prediction framework, revenue oriented cloud microservices, written in golang, are initiated and the credits for city stations are calculated based on the prediction queries. In the experiments, the prediction framework for predicting SO₂, NO₂, and RSPM attained over 70 to 90 percentage of prediction accuracies.

4. Big data platform for air quality analysis and prediction

Authors: Yue Shan Chang ; Kuan-Ming Lin ; Yi-Ting Tsai ; Yu-Ren Zeng ; Cheng-Xiang Hung, 2018 27th Wireless and Optical Communication Conference (WOCC)

With the advance of industry, air quality (AQ) is increasingly becoming worse. There are increasingly AQ monitors device have been deployed around country for monitoring air-quality all year long. To estimate and predict AQ, such as PM (particulate matter) 2.5, become an important issue for government to improve people's quality of life. As we can know, there are many factors can affect the AQ, such as traffic, factory exhaust emissions, weather, incineration of garbage, and so on. In most well-developed countries, these pollution sources are monitored for

future environmental policy making. In this paper, we will propose a semantic ETL (Extract-Transform-Load) framework on cloud platform for AQ prediction. In the

platform, we exploit ontology to concretize the relationship of PM 2.5 from various data sources and to merge those data with the same concept but different naming into the unified database. We implement the ETL framework on the cloud platform, which includes computing nodes and storage nodes. The computing nodes are used to execute data mining algorithms for predicting, and storage modes are used to store retrieved, preprocessed, and analyzed data. We utilize restful web service as the front end API to retrieve analyzed data, and finally we exploit browser to show the visualized result to demonstrate the estimation and prediction. It shows that the big data access framework on the cloud platform can work well for air quality analysis.

5. Prediction of Air Quality Index Based on Improved Neural Network

Authors: Wang Zhenghua ; Tian Zhihui, 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)

In order to realize the prediction of city air quality status, this paper uses the improved BP neural network to establish a prediction model of air quality index. The model uses the characteristics of nonlinear fitting approximation of BP neural network to solve the problem that air quality has many influencing factors and is nonlinear and difficult to predict. Aiming at the problem of slow convergence and easy to fall into local optimal solution of BP neural network, genetic algorithm is used to optimize. This paper takes Xuchang city as an example. Results show that the average relative error of the air quality index is 22%. The accuracy rate was 80.44%. The accuracy level of air quality is 82.5%. Compared with the BP neural network, BP neural network prediction improved has a higher accuracy. The model has certain feasibility and adaptability.

6. Prediction of Air Quality Index Based on LSTM

Authors: Yu Jiao ; Zhifeng Wang ; Yang Zhang, 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)

In view of the increasing attention paid by the state to environmental governance in recent years and the continuous deterioration of air quality, this paper proposes a prediction model of environmental quality based on Long Short Term Memory

(LSTM). This paper uses data provided by the environmental protection department

to predict Air Quality Index (AQI) through temperature, PM2.5, PM10, SO₂, wind direction, NO₂, CO and O₃. Firstly, this paper introduces the background, technical characteristics, development status and problems of air environment monitoring. Then, it will introduce the environmental prediction model. Finally, we make AQI prediction by using LSTM and analyze the error of the prediction results. The results show that LSTM can predict air quality index well.

7. Data-driven prediction model of indoor air quality by the preprocessed recurrent neural networks

Authors: MinHan Kim ; YongSu Kim ; SuWhan Sung ; ChangKyoo Yoo, 2009 ICCAS-SICE

In this study, data-driven prediction methods based on recurrent neural networks (RNN) for indoor air quality in a subway station are developed. The RNN can predict the air pollutant concentration of PM10 and PM2.5 at a platform of a subway station by using the previous information of NO, NO₂, NOX, CO, CO₂, temperature, humidity, and PM10 and PM2.5 on yesterday. For comparison, the other prediction models such as neural networks (NN) and multiple regression model are used. To optimize the prediction model, the variable importance in the projection (VIP) of the PLS is used to select key input variables as a preprocessing step. Experimental result shows that the selected key variables have positive influence on the prediction performance. The predicted result of RNN model gives better modeling performance and higher interpretability than other data-driven prediction modeling methods.

8. A Predictive Data Feature Exploration-Based Air Quality Prediction Approach

Authors: Ying Zhang ; Yanhao Wang ; Minghe Gao ; Qunfei Ma ; Jing Zhao ; Rongrong Zhang ; Qingqing Wang ; Linyan Huang, 2019 IEEE Access

In recent years, people have been paying more and more attention to air quality because it directly affects people's health and daily life. Effective air quality prediction has become one of the hot research issues. However, this paper is suffering many challenges, such as the instability of data sources and the variation of pollutant concentration along time series. Aiming at this problem, we propose

an improved air quality prediction method based on the LightGBM model to predict the

PM2.5 concentration at the 35 air quality monitoring stations in Beijing over the next 24 h. In this paper, we resolve the issue of processing the high-dimensional large-scale data by employing the LightGBM model and innovatively take the forecasting data as one of the data sources for predicting the air quality. With exploring the forecasting data feature, we could improve the prediction accuracy with making full use of the available spatial data. Given the lack of data, we employ the sliding window mechanism to deeply mine the high-dimensional temporal features for increasing the training dimensions to millions. We compare the predicted data with the actual data collected at the 35 air quality monitoring stations in Beijing. The experimental results show that the proposed method is superior to other schemes and prove the advantage of integrating the forecasting data and building up the high-dimensional statistical analysis.

9. Spatio-temporal-Aware Sparse Denoising Autoencoder Neural Network for Air Quality Prediction

Authors: Yan Li ; Xiajiong Shen ; Daojun Han ; Jun Sun ; Yatian Shen, 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)

Nowadays, air quality prediction is an important task. The main way to solve this problem is to use neural network, which are robust enough to handle time sensitive data. In this paper, we propose a novel spatio-temporal-aware sparse denoising autoencoding neural network architecture for this task. Our model makes full use of the spatio-temporal characteristics of the data, thus the accuracy of the prediction is improved. Experiments on historical dataset released by China National Environmental Monitoring Station and China Meteorological Administration show that our model achieves better performances than other models.

10. A Sequence-to-Sequence Air Quality Predictor Based on the n-Step Recurrent Prediction

Authors: Bo Liu ; Shuo Yan ; Jianqiang Li ; Guangzhi Qu ; Yong Li ; Jianlei Lang ; Rentao Gu, 2019 IEEE Access

Increasingly, more people are suffering from the effects of air pollution. This study

took Beijing as an example and proposed an attention-based air quality predictor (AAQP) that could better protect people from air pollution. The AAQP is a seq2seq

model, and it exploits historical air quality data and weather data to predict future air quality indexes. Although existing research has promoted seq2seq for air quality prediction, there are still two problems. First, the seq2seq has a slow training speed so the original RNN in the encoder was replaced with a fully connected encoder to accelerate the training process. Position embedding was also introduced to help the fully connected encoder find the sequential relationships among source sequences. Another problem is error accumulation caused by recurrent prediction. Accordingly, the n-step recurrent prediction was proposed to solve this problem. The experimental results validated that the AAQP with n-step recurrent prediction had better performance than the related arts since the error accumulation was reduced, and the training time was significantly decreased compared with the original seq2seq attention model.

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 SYSTEM

To estimate the PM2.5 concentration by designing a photograph-based method. By observation, it is found that the saturation map is sensitive to air quality, exhibiting entirely different appearances under high and low PM2.5 concentrations. Specifically, it loses structures and most pixel values tend to be 0 under a high PM2.5 concentration. To compute the gradient similarity between the saturation and gray-scale maps to quantify the structural information loss. Then, utilizing the Weibull distribution to fit the saturation map and able to derive a value to estimate the color information. Finally, the PM2.5 concentration of an image can be estimated via the combination of the aforementioned two features followed by a nonlinear mapping procedure. Both numerical and visualized results on real captured data validate the effectiveness and superiority of the proposed method in comparison with the relevant state-of-the-art methods. Air pollution has become a worldwide concerned issue and automatic estimation of air quality can provide a positive guidance to both individual and industrial behaviors. Given that the traditional instrument-based method requires high economic, labor costs on instrument purchase and maintenance, proposes an effective, efficient, and cheap photo-based method for the air quality estimation in the case of particulate matter (PM2.5). This method lies in extracting two categories of features (including the gradient similarity and distribution shape of pixel values in the saturation map) by observing the photo appearances captured under different PM2.5 concentrations. Specifically, the gradient similarity is extracted to measure the structural information loss with the consideration that PM2.5 attenuates the light rays emitted from the objects and accordingly distorts the structures of the formed photo. Meanwhile, the saturation map is fit by the Weibull distribution to quantify the color information loss. By combining two features, a primary PM2.5 concentration estimator is obtained. Next, a nonlinear function is adopted to map the primary one to the real PM2.5 concentration. Sufficient experiments on real data captured by

professional PM2.5 instrument demonstrate the effectiveness and efficiency of the proposed method.

Specifically, it is highly consistent with real sensor's measures and requires low implementation time.

3.2 Drawbacks

- Photo graphic method is not sufficient to calculate PM2.5 and it taken only one pollutants of concentration.
- the work is given based on only one PM2.5 and to collect more monitoring data from other cities to verify the generalization of our work and more factors, e.g., geomorphic conditions.
- It can't thereby better determine the regularity of air pollutant data and achieve more accurate prediction results.

3.3 Proposed System

In proposed we apply decision tree & linear regression algorithm.

We will increase high accuracy

- While applying photo graphic based method is critical to evaluate parameters and it's taken data size is high
- To overcome this method to implement machine learning approach by user interface of GUI application
- Multiple datasets from different sources would be combined to form a generalized dataset, and then different machine learning algorithms would be applied to extract patterns and to obtain results with maximum accuracy.

3.4 Logistic Regression:

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest

(dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables .

Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts $P(Y=1)$ as a function of X . Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

3.5 Random Forest:

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines

multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*,

hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm :

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
- In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

Chapter 4

Software development methodology

4.1 System Specification

HARDWARE SPECIFICATION

System : Pentium IV 2.4 GHz

Hard Disk : 40 GB

Monitor : 15 VGA Colour

Mouse :

Logitech Ram : 8gb

SOFTWARE SPECIFICATION

Operating system : Windows 10

Coding Language :

Python

Dataset : csv

4.2 Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

Jupyter Lab

Jupyter Notebook

QtConsole

Spyder

Glueviz

4.3 LANGUAGES USED IN THE SYSTEM

A language is a formal constructed symbol designed to communicate instructions to a machine, particularly a computer. Languages can be used to create programs and control the behavior of the machines or for expressing the algorithms. This work uses python language.

4.3.1 About Python Language

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Programming Characteristics

It provides rich data types and easier to read syntax than any other programming languages

It is a platform independent scripted language with full access to operating system API's

Compared to other programming languages, it allows more run-time flexibility

It includes the basic text manipulation facilities of Perl and Awk

A module in Python may have one or more classes and free functions

Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows

For building large applications, Python can be compiled to byte-code

Python supports functional and structured programming as well as OOP

It supports interactive mode that allows interacting Testing and debugging of snippets of code

In Python, since there is no compilation step, editing, debugging and testing is fast.

Applications of Python

Programming Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the

prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.

Python GUI Tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app:

- Importing the module – tkinter

- Create the main window (container)

- Add any number of widgets to the main window

- Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

- **pack() method:**It organizes the widgets in blocks before placing in the parent widget.
- **grid() method:**It organizes the widgets in grid (table-like structure) before placing in the parent widget.
- **place() method:**It organizes the widgets by placing them on specific

positions directed by the programmer.

About Opencv Package

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

4.4 METHODOLOGY

4.4.1 SYSTEM DESIGN

Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

Data collection

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, and logistic regression are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

Building the classification model

The predicting the air quality problem, decision tree algorithm prediction model is effective because of the following reasons: It provides better results in classification problem.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.
- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

4.5 Construction of a Predictive Model

Machine learning needs data gathering have lot of past data's. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can't be used directly. It's used to preprocess then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

Architecture diagram

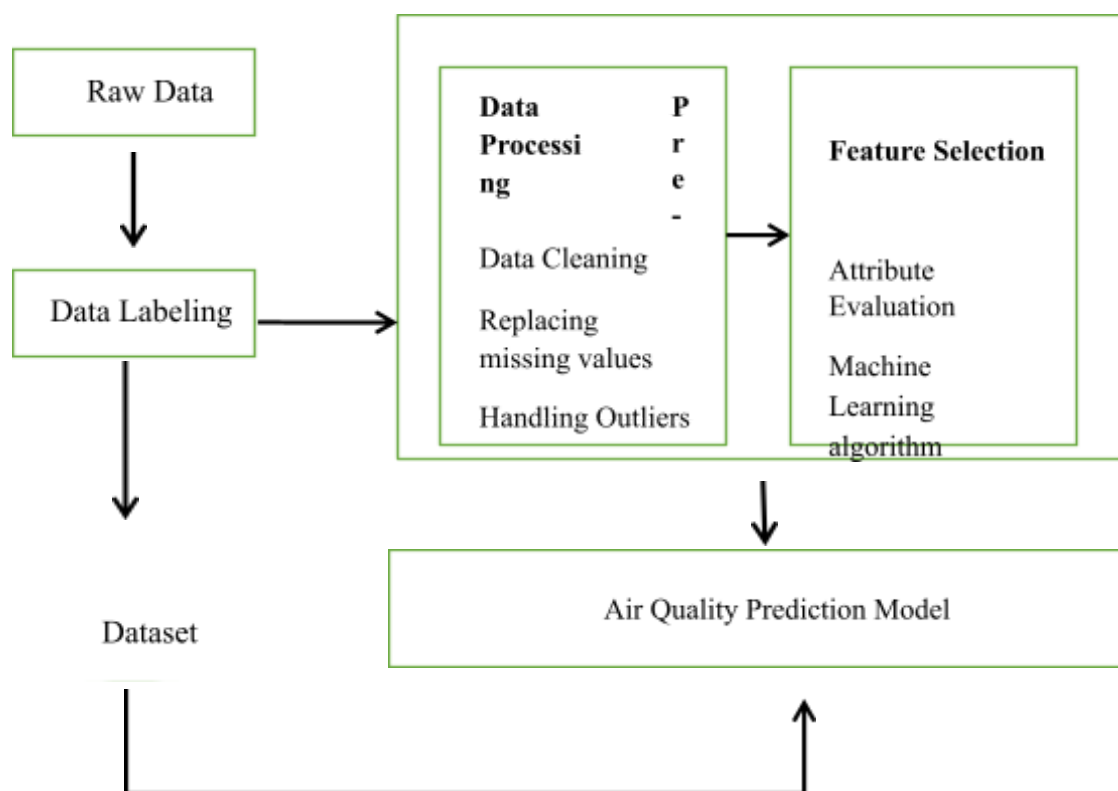


FIG NO 4.1 : ARCHITECTURE DIAGRAM

4.6 UML DIAGRAMS

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full life cycle of object oriented development. Unified Modeling Language is a general purpose developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of the system.

UML is a standard language for specifying, visualizing, and documenting of software systems and created by Object Management Group (OMG) in 1997. There are three important type of UML modeling are Structural model, Behavioral model, and Architecture model. To model a system the most important aspect is to capture the dynamic behavior which has some internal or external factors for making the interaction. These internal or external agents are known as actors. It consists of actors, use cases and their relationships. In this fig we represent the Use Case diagram for our project.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core

concepts.

3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations , frameworks, patterns and components.

4.7 USECASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components a user or another system that will interact with the system modeled. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

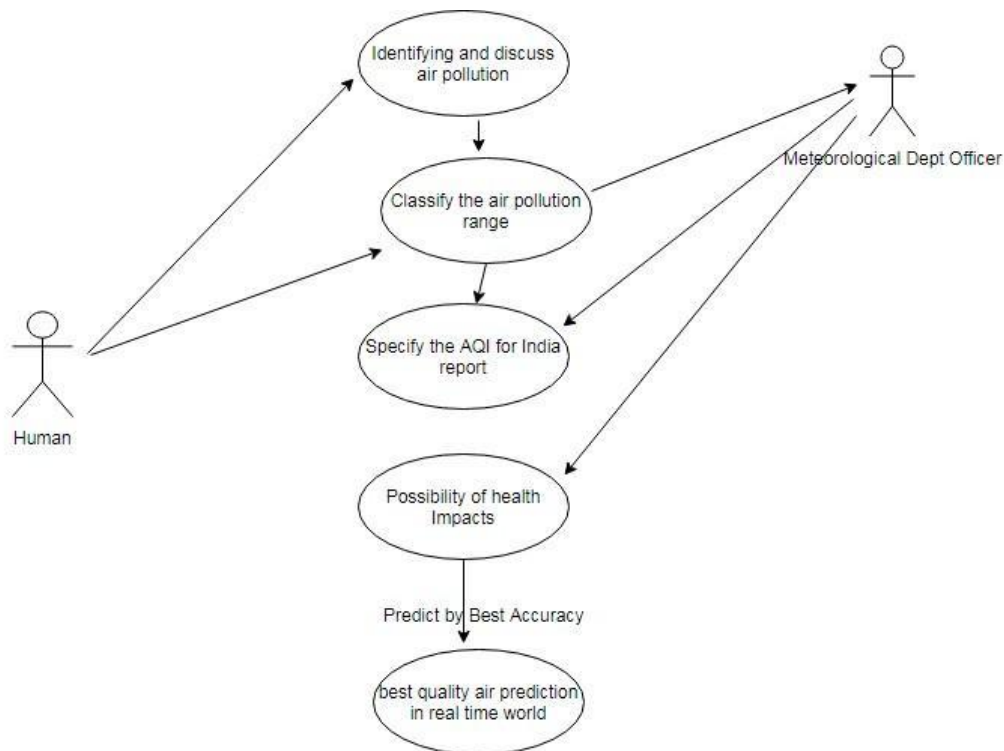


FIG 4.2 USECASE

Sequence Diagram:

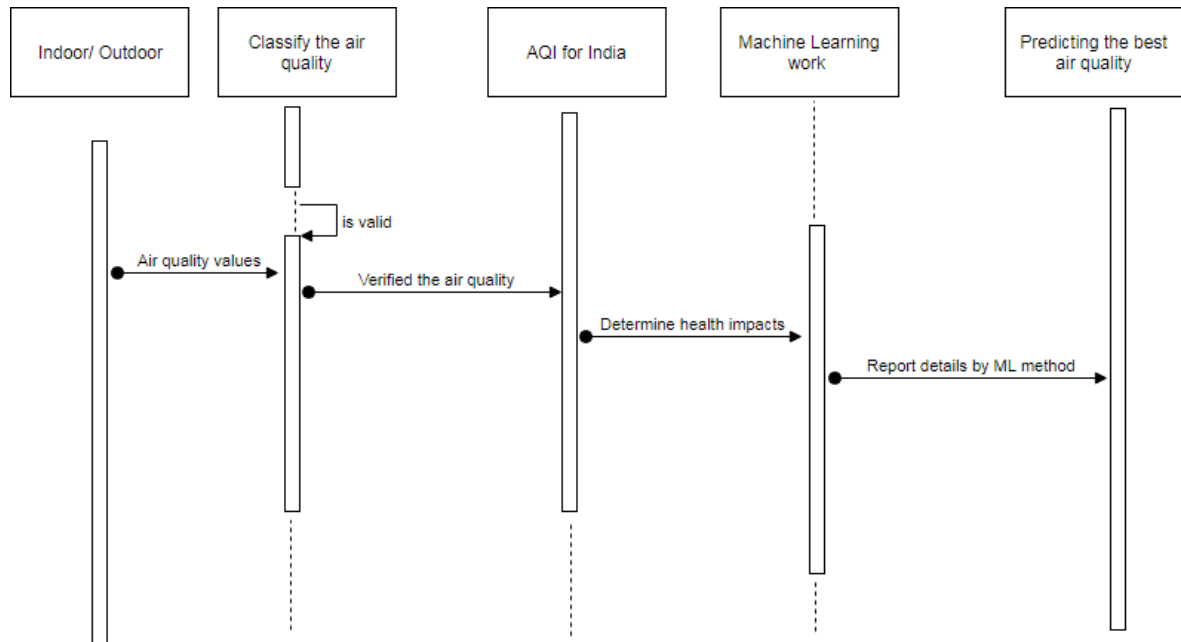


FIG 4.3 : SEQUENCE DIAGRAM

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Other dynamic modeling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design- level models for modern business application development.

Entity Relationship Diagram (ERD)

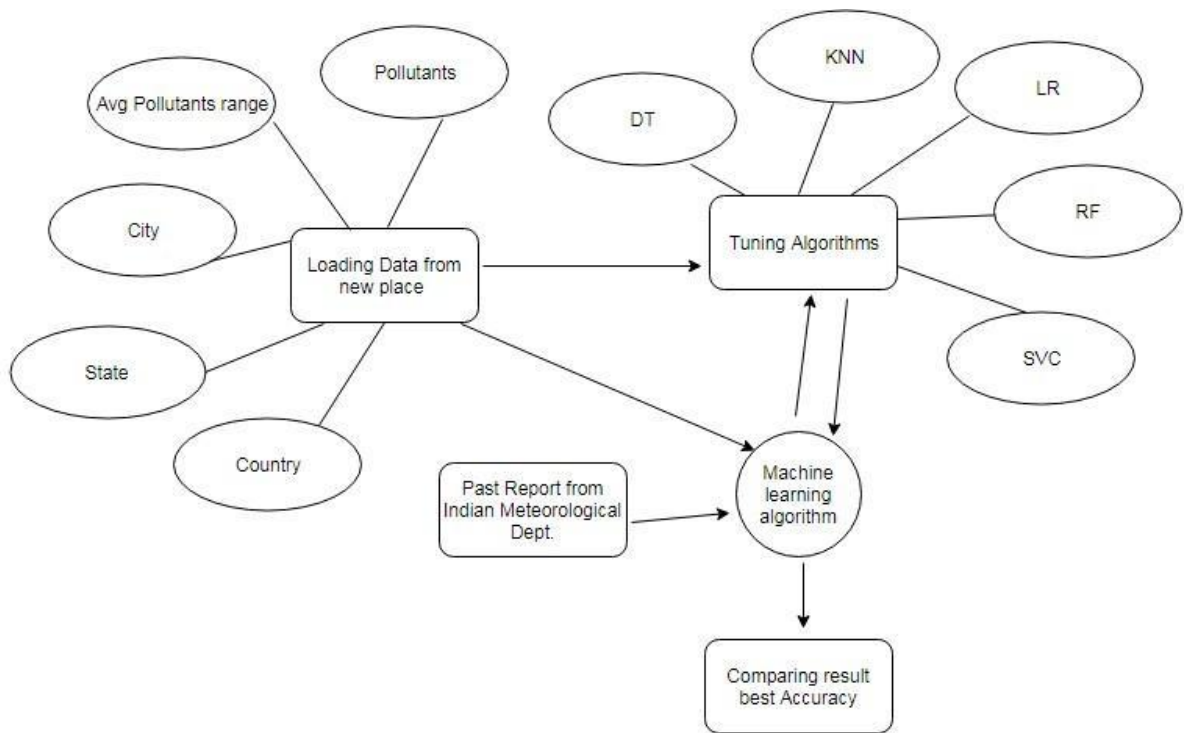


FIG 4.4 : ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

Modules:

- Data validation and pre-processing technique (Module-01)
- Exploration data analysis of visualization and training a model by given attributes (Module-02)
- Performance measurements of Logistic regression and decision tree algorithm (Module-03)
- Performance measurements of Random forest and support vector machine (Module-04)
- Performance measurements of Naive Bayes and K-Nearest neighbor (Module-05)
- GUI based prediction of air quality (Module-06)

4.8 Model description:

Module-01:

Variable Identification Process / data validation process:

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

Data Validation/ Cleaning/Preparing Process:

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets

when evaluating your models. Data cleaning / preparing by rename the given

dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

Data Pre-processing:

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner.

Module-02:

Exploration data analysis of visualization:

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

4.9 TESTING

4.9.1 Software Testing

In a generalized way, we can say that the system testing is a type of testing in

which the main aim is to make sure that system performs efficiently and seamlessly. The

process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

4.9.2 TEST CASE

Testing, as already explained earlier, is the process of discovering all possible weak- points in the finalized software product. Testing helps to counter the working of sub- assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

4.9.3 Testing Techniques

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineers.

(I) UNIT TESTING

In unit testing, the design of the test cases is involved that helps in the validation of

the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken

into account after the individual unit is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and contains clearly defined inputs with the results which are expected.

(II) INTEGRATION TESTING

These tests are designed to test the integrated software items to

determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the component are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.

(III) FUNCTIONAL TESTING

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

(IV) SYSTEM TESTING

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

(V) WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

(VI) BLACK BOX TESTING

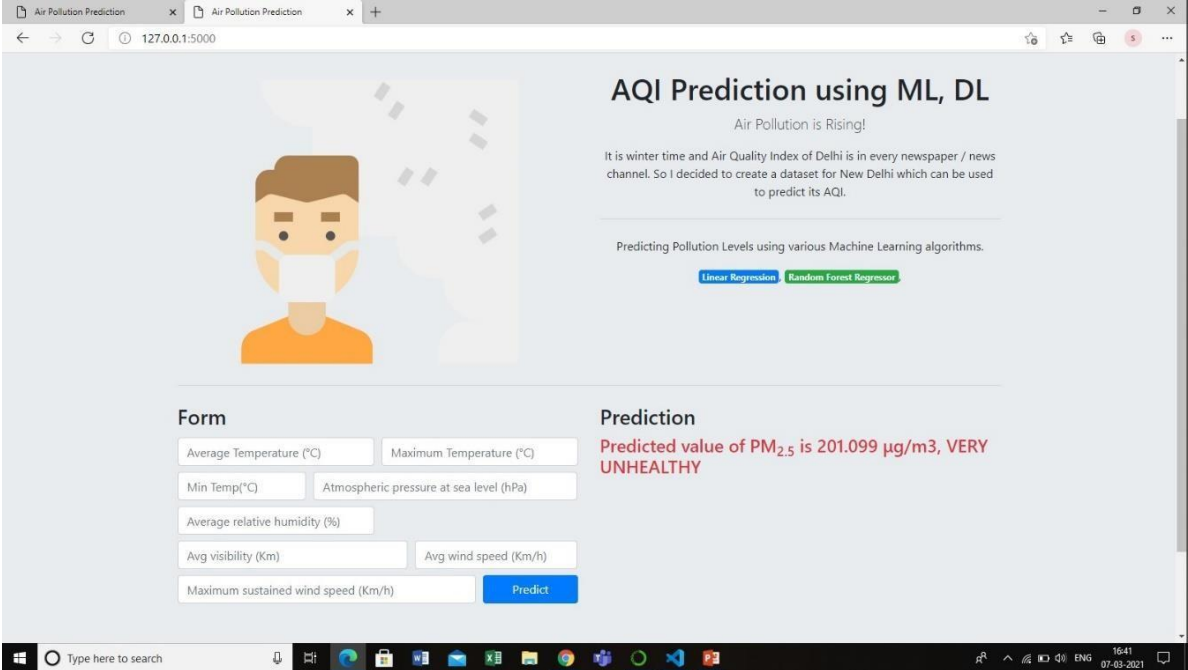
The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are straight-forward in nature. It is also less costly than white box testing.

(V) 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 5

RESULTS AND DISCUSSION



AQI Prediction using ML, DL

Air Pollution is Rising!

It is winter time and Air Quality Index of Delhi is in every newspaper / news channel. So I decided to create a dataset for New Delhi which can be used to predict its AQI.

Predicting Pollution Levels using various Machine Learning algorithms.

[Linear Regression](#) [Random Forest Regressor](#)

Form

Average Temperature (°C) Maximum Temperature (°C)

Min Temp(°C) Atmospheric pressure at sea level (hPa)

Average relative humidity (%)

Avg visibility (Km) Avg wind speed (Km/h)

Maximum sustained wind speed (Km/h) [Predict](#)

Prediction

Predicted value of PM_{2.5} is 201.099 µg/m3, **VERY UNHEALTHY**

FIG 5.1 : FORM

In this form we need to give the following information temperature average, minimum and maximum, relative humidity, visibility and wind speed. Once this information is this data will be loaded and it will predict the amount of PM 2.5 which defines whether the air quality is healthy or not.

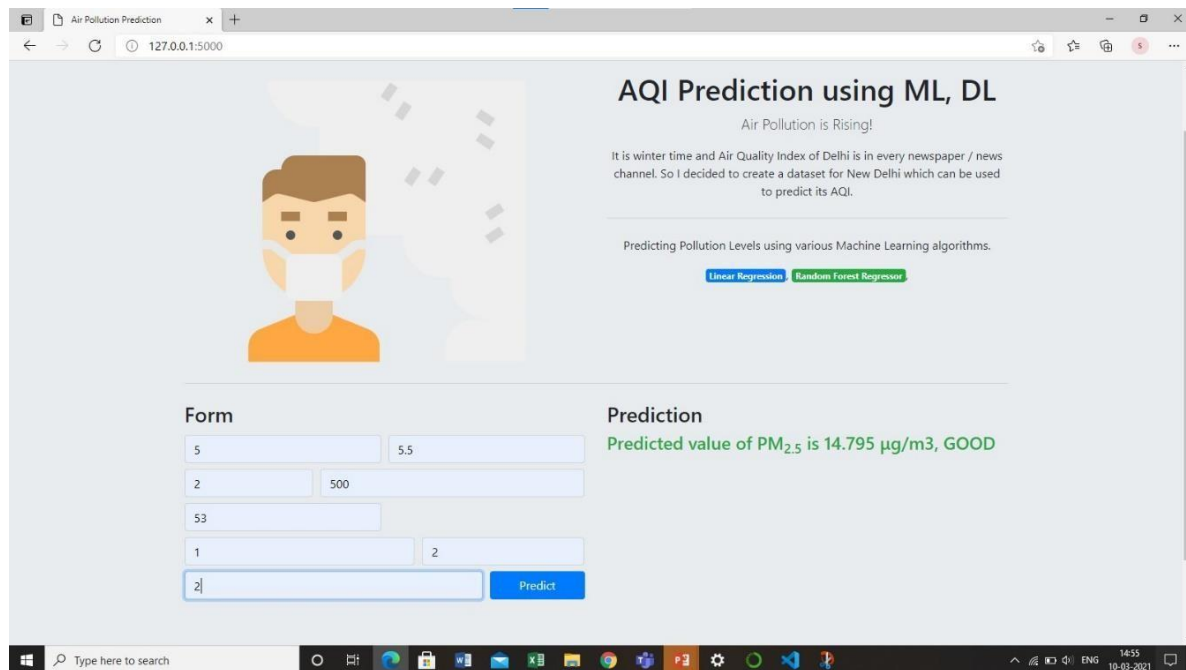


FIG 5.2 : PREDICTED RESULT

When the PM 2.5 is less than 15 microgram per cubic metre it can be said as good. When the PM 2.5 is between 15 to 35 microgram per cubic metre then it can be said as moderate. When the PM 2.5 is between 15 to 35 microgram per cubic metre then it can be said as moderate and if it is between 35 to 55 microgram per cubic metre it is unhealthy for sensitive people. If the predicted value is greater than 55 microgram per cubic metre then it is unhealthy. If the predicted PM 2.5 is above 150 microgram per cubic metre then it is very unhealthy and if it is greater than 250 microgram per cubic metre then it is hazardous.

CHAPTER 6

CONCLUSIO

N

The analytical procedure began from information cleaning and processing, incomplete records, detailed evaluation and in the end model constructing and evaluation. The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score for predicting the air quality by given attributes. The first-rate accuracy on public test set is good parameter values of decision tree method process by way of accuracy with classification record. This application can help India meteorological division in predicting the way forward for air nice and its reputation and will depend on that they are able to take motion.

FUTURE WORK □

India meteorological department wants to automate the detecting the air quality is good or not from eligibility process (real time). To automate this process by show the prediction result in web application or desktop application. To optimize the work to implement in Artificial Intelligence environment.

India meteorological department wants to automate the detecting the air quality is good or not from eligibility process (real time).

To automate this process by show the prediction result in web application or desktop application.

To optimize the work to implement in Artificial Intelligence environment.

REFERENCE

1. Application of BP neural network optimized by genetic simulated annealing algorithm to prediction of air quality index in Lanzhou, Zhou Kang ; Zhiyi Qu, 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)
2. Air Quality Prediction Method in Urban Residential Area, RuiJun Yang ; HaiLong Zhou ; DanFeng Ding, 2018 11th International Symposium on Computational Intelligence and Design (ISCID)
3. Revenue oriented air quality prediction microservices for smart cities, Shajulin Benedict, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)
4. Big data platform for air quality analysis and prediction, Yue Shan Chang ; Kuan-Ming Lin ; Yi-Ting Tsai ; Yu-Ren Zeng ; Cheng-Xiang Hung, 2018 27th Wireless and Optical Communication Conference (WOCC)
5. Prediction of Air Quality Index Based on Improved Neural Network
Authors: Wang Zhenghua ; Tian Zhihui, 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)
6. Prediction of Air Quality Index Based on LSTM, Yu Jiao ; Zhifeng Wang ; Yang Zhang, 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)

7. Data-driven prediction model of indoor air quality by the preprocessed recurrent neural networks, MinHan Kim ; YongSu Kim ; SuWhan Sung ; ChangKyoo Yoo, 2009 ICCAS-SICE

8.A Predictive Data Feature Exploration-Based Air Quality Prediction Approach, Ying Zhang ; Yanhao Wang ; Minghe Gao ; Qunfei Ma ; Jing Zhao ; Rongrong Zhang ; Qingqing Wang ; Linyan Huang, 2019 IEEE Access

9.Spatio-temporal-Aware Sparse Denoising Autoencoder Neural Network for Air Quality Prediction, Yan Li ; Xiajiong Shen ; Daojun Han ; Jun Sun ; Yatian Shen, 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)

10.A Sequence-to-Sequence Air Quality Predictor Based on the n-Step Recurrent Prediction, Bo Liu ; Shuo Yan ; Jianqiang Li ; Guangzhi Qu ; Yong Li ; Jianlei Lang ; Rentao Gu, 2019 IEEE Access

APPENDIX

Source Code

```
from flask import Flask, render_template, request
import joblib
import numpy as np

app = Flask(__name__)

lr = joblib.load("Models/linear_regression_model.pkl")

@app.route("/")
def home():
    return render_template('index.html')

@app.route('/', methods = ['POST'])
def main():
    if request.method == 'POST':
        T, TM, Tm, SLP, H, VV, V, VM = float(request.form['T']), float(request.form['TM']),
float(request.form['Tm']), float(request.form['SLP']),
float(request.form['H']), float(request.form['VV']), float(request.form['V']),
float(request.form['VM'])
        lr_pm = lr.predict([[T, TM, Tm, SLP, H, VV, V, VM]])

        # print(lr_pm)

    return render_template("index.html", lr_pm = np.round(lr_pm,3))
```

```
if_name_____== "_main_":
    app.run(debug = True)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink- to-fit=no">

    <title>Air Pollution Prediction</title>

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.c
s s"
    integrity="sha384-
Vkoo8x4CGsO3+Hhvxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23
Q 9Ifjh" crossorigin="anonymous">

</head>

<body class="bg-dark">

    <!-- Navbar -->

    <nav class="navbar navbar-dark bg-dark">

        <div class="container">

            <span class="navbar-brand mb-0 h1">Air Pollution Predictor</span>

        </div>

    </nav>

    <!-- Header -->

    <div class="jumbotron jumbotron-fluid">
```

```

<div class="container">

  <div class="row">

    <div class="col-md-6">

      <div class="text-center">

        </div>

      </div>

      <div class="col-md-6 text-center">

        <h1 class="display-5">AQI Prediction using ML, DL</h1>

        <p class="lead">Air Pollution is Rising!</p>

        <p>It is winter time and Air Quality Index of Delhi is in every
newspaper / news channel. So I

          decided to create a dataset for New Delhi which can be used to
predict its AQI.</p>

        <hr class="my-4">

        <p>Predicting Pollution Levels using various Machine
Learning algorithms.</p>

        <p><span class="badge badge-primary">Linear
Regression</span>,

          <span class="badge badge-success">Random Forest
Regressor</span>,

        </p>

      </div>

    </div>

  <hr>

<!-- Row Ends -->

```

<!--T Average Temperature (°C)
TM Maximum temperature (°C)
Tm Minimum temperature (°C)
SLP Atmospheric pressure at sea level (hPa)
H Average relative humidity (%)
PP Total rainfall and / or snowmelt (mm)
VV Average visibility (Km)
V Average wind speed (Km/h)
VM Maximum sustained wind speed (Km/h) -->

```
<div class="row my-4">  
  <div class="col-md-6">  
    <h3>Form</h3>  
    <form method="POST" , action="/">  
      <div class="form-row">  
        <div class="col-6">  
          <input type="text" class="form-control my-1"  
placeholder="Average Temperature (°C)"  
          name='T' required>  
        </div>  
        <div class="col-6">  
          <input type="text" class="form-control my-1"  
placeholder="Maximum Temperature (°C)"  
          name='TM' required>  
        </div>  
        <div class="col-4">  
          <input type="text" class="form-control my-1"  
placeholder="Min Temp(°C)" name='Tm'  
          required>
```

```
</div>

<div class="col-8">
  <input type="text" class="form-control my-1"
    placeholder="Atmospheric pressure at sea level (hPa)"
    name='SLP' required>
</div>

<div class="col-6">
  <input type="text" class="form-control my-1"
    placeholder="Average relative humidity (%)"
    name='H' required>
</div>

<div class="col-7">
  <input type="text" class="form-control my-1"
    placeholder="Avg visibility (Km)" name='VV'
    required>
</div>

<div class="col-5">
  <input type="text" class="form-control my-1"
    placeholder="Avg wind speed (Km/h)"
    name='V' required>
</div>

<div class="col-9">
  <input type="text" class="form-control my-1"
    placeholder="Maximum sustained wind speed (Km/h)"
    name='VM' required>
</div>

<div class="col-3">
  <button type="submit" class="btn btn-primary my-1 btn-
    block">Predict</button>
</div>
```

```
</div>

</form>

</div>
```

```
<div class="col-md-6">

  <h3>Prediction </h3>

  {% if lr_pm %}

  {% if lr_pm < 51 %}

    <h4 class="text text-success">Predicted value of
PM<sub>2.5</sub> is {{ lr_pm[0] }} µg/m3, GOOD</h4>

    {% elif lr_pm < 101 %}

      <h4 class="text text-info">Predicted value of PM<sub>2.5</sub>
is {{ lr_pm[0] }} µg/m3, MODERATE

      </h4>

      {% elif lr_pm < 151 %}

        <h4 class="text text-primary">Predicted value of
PM<sub>2.5</sub> is {{ lr_pm[0] }} µg/m3, UNHEALTHY

        FOR KIDS</h4>

        {% elif lr_pm < 201 %}

          <h4 class="text text-warning">Predicted value of
PM<sub>2.5</sub> is {{ lr_pm[0] }} µg/m3, UNHEALTHY

          </h4>

          {% elif lr_pm < 301 %}

            <h4 class="text text-danger">Predicted value of
PM<sub>2.5</sub> is {{ lr_pm[0] }} µg/m3, VERY

            UNHEALTHY</h4>

            {% else %}

              <h4 class="text text-danger">Predicted value of
PM<sub>2.5</sub> is {{ lr_pm[0] }} µg/m3, HAZARDOUS

              </h4>
```

```
        {% endif %}
        {% endif %}
    </div>
</div>
</div>
</div>

<!-- Footer -->
<footer id="sticky-footer" class="py-4 bg-dark text-white-50">
    <div class="container text-center text-white">
        <a href="https://avikasliwal.github.io/Portfolio/" target="_blank">Avi
Kasliwal | </a>
        <a href="https://github.com/AviKasliwal/" target="_blank">Github | </a>
        <a href="https://www.linkedin.com/in/avi-kasliwal/"
target="_blank">Linkedin | </a>
        <a href="https://github.com/AviKasliwal/AirQualityPrediction"
target="_blank">Code </a>
    </div>
</footer>
</body>

</html>
```

