

AIR QUALITY PREDICTION BASED ON SUPERVISED WITH CLASSIFICATION MACHINE LEARNING APPROACH

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

Karan Parsi (37110320)

K. Prudhvinath Reddy (37110334)



**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 | 12B Status by

UGC I Approved by AICTE

**JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI - 600119**

MARCH - 2021



SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119

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 **KARAN PARSI (37110320)**, **K. PRUDHVINATH REDDY (37110334)** who carried out the project entitled “**AIR QUALITY PREDICTION BASED ON SUPERVISED WITH CLASSIFICATION MACHINE LEARNING APPROACH**” under my supervision from November 2020 to March 2021.

Internal Guide

Dr. Asha. P M.E., PhD.,

Head of the Department

Dr. L. LAKSHMANAN M.E, Ph.D.,

Dr. S. VIGNESHWARI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KARAN PARSI (37110320), K. PRUDHVINATH REDDY (37110334)** hereby declare that the Project Report entitled "**AIR QUALITY PREDICTION BASED ON SUPERVISED WITH CLASSIFICATION MACHINE LEARNING APPROACH**" done by me under the guidance of **Dr. P. ASHA M.E., Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering in **Computer Science and Engineering**.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA 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 and **Dr. S. VIGNESHWARI M.E., Ph.D.**, and **Dr. L. LAKSHMANAN M.E, Ph.D.**, Head of the Department, Dept. of Computing 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 ASHA M.E., PhD.**, for his 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 Computing 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 analyse 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.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 LITERATURE SURVEY	2
2	AIM AND SCOPE OF INVESTIGATION	7
	2.1 AIM AND SCOPE OF PROJECT	7
	2.2 PROJECT GOALS	7
	2.3 EXISTING SYSTEM	7
	2.4 PROPOSED SYSTEM	8
	2.4.1 DATA WRANGLING	8
	2.4.2 BUILDING CLASSIFICATION	
	MODEL	8
	2.4.3 PREDICTIVE MODEL	
	CONSTRUCTION	9
3	SYSTEM DESIGN AND IMPLEMENTATION	10
	3.1 PROJECT REQUIREMENTS	10
	3.1.1 FUNCTIONAL REQUIREMENT	10
	3.1.2 NON-FUNCTIONAL	
	REQUIREMENT	10
	3.1.3 ENVIRONMENTAL	
	REQUIREMENTS	10
	3.2 SOFTWARE DESCRIPTION	11
	3.2.1 CONDA	11
	3.2.2 THE JUPYTER NOTEBOOK	11

	3.2.3 KERNEL	12
	3.2.4 NOTEBOOK DASHBOARD	12
	3.3 WORKING PROCESS	13
	3.4 MODULES	13
	3.4.1 VARIABLE IDENTIFICATION PROCESS / DATA VALIDATION PROCESS	14
	3.4.2 EXPLORATION OF DATA	15
	3.4.3 PERFORMANCE MEASUREMENT OF LOGISTIC REGRESSION AND NAÏVE BAYES	18
	3.4.4 PERFORMANCE MEASUREMENT OF SVM AND RANDOM FOREST ALGORITHMS	20
	3.4.5 PERFORMANCE MEASUREMENT OF KNN AND DECISION TREE ALGORITHM	21
	3.4.6 GUI BASED AIR QUALITY	22
4	RESULTS AND DISCUSSION	27
5	CONCLUSION AND FUTURE WORK	30
	5.1 CONCLUSION	30
	5.2 FUTURE WORK	30
	REFERENCES	31
	APPENDIX	32
	A.SOURCE CODE	32
	B.SCREENSHOTS	44
	C.PUBLICATION	45

LIST OF FIGURES

- 1.1 Process of Machine Learning**
- 1.2 Data flow Diagram**
- 3.1 Data flow Diagram for Machine Learning Model**
- 3.2 Percentage of each pollutants**
- 3.3 Air Quality and Non-Air Quality**
- 4.1 Training Data**
- 4.2 Confusion Matrix Result**
- 4.3 Input Page(GUI Interface)**
- 4.4 Expected Output on GUI Interface**
- 4.5 Air Quality for state Karnataka**
- 4.6 Air Quality for state Andhra Pradesh**
- 4.7 Air Quality for state Maharashtra**

LIST OF ABBREVIATIONS

AI	ARTIFICIAL INTELLIGENCE
ML	MACHINE LEARNING
AQI	AIR QUALITY INDEX
GUI	GRAPHICAL USER INTERFACE
RNP	REQUIRE NAVIGATION PERFORMANCE
SVM	SUPPORT VECTOR MACHINE
UI	USER INTERFACE
MMH	MAXIMAL MARGIN HYPERPLANE
KNN	K-NEAREST NEIGHBORS
LR	LOGISTIC REGRESSION
FP	FALSE POSITIVE
FN	FALSE NEGATIVES
TP	TRUE POSITIVES
TN	TRUE NEGATIVES

1.1 OVERVIEW

C
H
A
P
T
E
R

1
I
N
T
R
O
D
U
C
T
I

O

N

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 in to 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 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. Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where 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 $y = f(X)$.



Fig 1.1 Process of Machine learning

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 labelled 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. 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.

1.2 LITERATURE SURVEY

Title: Air pollutant severity prediction using Bi-directional LSTM Network

Author: Ishan Verma, Rahul Ahuja and Hardik Meisheri

Year: 2018

Recurrent Neural Networks (RNN) has proved to be very efficient in processing

temporal data. However, future input information coming up later than the current time instance is also useful for prediction. RNNs can partially achieve this by delaying the output by a certain number of time frames to include future information. Theoretically, a large delay

could be used but in practice, it is found that prediction results drop if the delay is too large. While delaying the output by some frames has been used successfully to improve results for sequential data, the optimal delay is task dependent and need to be obtained by the trial and error method. Also, two separate networks, one for each direction could be trained on all input information and then the results could be merged using arithmetic or geometric averaging for final prediction. However, it is difficult to obtain optimal merging since different networks trained on the same data can no longer be regarded as independent. To overcome these limitations, it proposed bidirectional recurrent neural network (BRNN) that can be trained using all available input information in the past and future of a specific time frame. Pollutant data like any other sensor data is not free from missing data and abnormal values. The irregularities may occur due to instrumental error or some other external factors like power-shutdown or severance of connectivity etc. There were instances where pollutant data was not reported by a source monitoring station. These missing values were interpolated using rolling average of available data values of past three time instances. A value lying outside the permissible range for a parameter is treated as an abnormal value. Abnormal values are also replaced by rolling average of past three instances. It presented an effective way of predicting the severity of pollutants by leveraging various sensor data in 6,12 and 24 hour in advance using Deep learning models. We have validated this claim over the pollution data from New Delhi, India by predicting the severity of PM2.5 pollutant. We present our experimentations with comparison to baseline system over different stations and over different time periods. In addition, we have presented a Ensemble system which performs better in most of the cases, and also proves to be more robust.

Title: Air pollution monitoring and prediction using IoT

Author: Temesagan Walelign Ayele, Rutvik Mehta

Year: 2018

It is obvious that those who work in a factory or plant will be far more at risk of inhalation

of harmful chemicals and gases due to their prolonged exposure to emissions. Air pollution adds to the harmful condition that makes unfavorable impact on living things. It is one of the real concerns for the entire world. Air pollution is a worldwide issue including

international organizations, governments, and the mass media. Any utilization of natural assets at a higher rate than the nature's ability to reestablish itself can bring about contamination of plants, air, and water. Other than human exercises, there are a couple of intermittent characteristic cycles that additionally result in release of risky stuff. Beside human made activities natural disaster such as volcanic eruption may result in the contamination of air. Technology is passing around its wing in almost every walk of human life activities. Now a day it is better if every action is done using new technology in order to satisfy the demand of human being, Organization, Enterprise etc. Internet of Things (IoT) is one of the main communication developments in the last decade. Through this concept, it is possible to connect countless low-powered smart embedded objects to each other and to the Internet. The pervasive presence around us of various wireless technologies such as radio Frequency Identification (RFID) tags, sensors, actuators and mobile phones constitutes the cornerstone of the IoT concept. These objects can send and receive data autonomously, thus opening new horizons for home, health, and industrial applications. In fact, technology advances along with increasing demand will foster a widespread deployment of IoT services, which would radically transform our corporations, communities and personal lives. Now a day the air pollution in urban areas is a major issue in developed cities due to significant impacts of air pollution on public health, global environment and the whole worldwide economy. The proposed work on an air pollution monitoring and prediction system is enables us to monitor air quality with the help IoT devices. The system utilizes air sensors to detect and transmit this data to microcontroller. Then the microcontroller stores the data into the web server. For predicting the LSTM is implemented. It has a quick convergence and reduces the training cycles with a good accuracy.

Title: Adverse health effects of outdoor air pollutants

Author: Luke Curtis, William Rea, Patricia Smith-Willis

Year: 2006

Outdoor air quality plays an important role in human health. Air pollution causes large increases in medical expenses, morbidity and is estimated to cause about 800,000 annual

premature deaths worldwide (Cohen et al., 2005). The outdoor air often contains biologically significant levels of many pollutants including particulates (PM10 or PM2.5), ozone, carbon monoxide, oxides of nitrogen and sulfur, bioaerosols, metals, volatile organics and pesticides. A large percentage of these pollutants are produced by anthropogenic activities. While most people spend the majority of their time indoors, outdoor air quality can affect indoor air quality to a large degree. In addition many patients such as asthmatics, patients with allergies and chemical sensitivities, COPD patients, heart and stroke patients, diabetics, pregnant women, the elderly and children are especially susceptible to poor outdoor and indoor air quality. Much research on the health effects of outdoor air pollution has been published in the last decade. The goal of this review is to concisely summarize a wide range of the recent research on health effects of many types of outdoor air pollution. A review of the health effects of major outdoor air pollutants including particulates, carbon monoxide, sulfur and nitrogen oxides, acid gases, metals, volatile organics, solvents, pesticides, radiation and bio aerosols is presented. Numerous studies have linked atmospheric pollutants to many types of health problems of many body systems including the respiratory, cardiovascular, immunological, hematological, neurological and reproductive/ developmental systems.

Title: Urban air Pollution Monitoring System with Forecasting Model

Author: Khalid Bashir Shaban, Abdullah Kadri and Eman Rezk

Year: 2016

It is widely believed that urban air pollution has a direct impact on human health especially in developing and industrial countries, where air quality measures are not available or minimally implemented or enforced. Recent studies have shown substantial evidences that exposure to atmospheric pollutants has strong links to adverse diseases including asthma and lung inflammation. The modules are responsible for receiving and storing the data, preprocessing and converting the data into useful information, forecasting the pollutants based on historical information, and finally presenting the

acquired information through different channels, such as mobile application, Web portal, and short message service. The

focus of this paper is on the monitoring system and its forecasting module. Three machine learning (ML) algorithms are investigated to build accurate forecasting models for one-step and multi-step ahead of concentrations of ground-level ozone (O_3), nitrogen dioxide (NO_2), and sulfur dioxide (SO_2). These ML algorithms are support vector machines, M5P model trees, and artificial neural networks (ANN). Two types of modelling are pursued: 1) uni-variate and 2) multivariate. The performance evaluation measures used are prediction trend accuracy and root mean square error (RMSE). The results show that using different features in multivariate modelling with M5P algorithm yields the best forecasting performances. For example, using M5P, RMSE is at its lowest, reaching 31.4, when hydrogen sulfide (H_2S) is used to predict SO_2 . Contrarily, the worst performance, i.e., RMSE of 62.4, for SO_2 is when using ANN in uni-variate modelling. The outcome of this paper can be significantly useful for alarming applications in areas with high air pollution levels. Air quality is an important problem that directly affects human health. Air quality data are collected wirelessly from monitoring nodes that are equipped with an array of gaseous and meteorological sensors. These data are analyzed and used in forecasting concentration values of pollutants using intelligent machine to machine platform. The platform uses ML-based algorithms to build the forecasting models by learning from the collected data.

CHAPTER 2

AIM AND SCOPE OF INVESTIGATION

2.1 AIM AND SCOPE OF THE PROJECT:

This helps all others department to carry out other formalities. It has to find Accuracy of the training dataset, Accuracy of the testing dataset, Specification, False Positive rate, precision and recall by comparing algorithm using python code.

The goal is to develop a machine learning model for consumer purchase intension forecasting, potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithm.

2.2 PROJECT GOALS

Web scraping

Exploration data analysis of variable identification

Uni-variate data analysis

Exploration data analysis of bi-variate and multi-variate

Method of Outlier detection with feature engineering

2.3 EXISTING 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 (PM_{2.5}). 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 PM_{2.5} concentrations. Specifically, the gradient similarity is extracted to measure the structural information loss with the consideration that PM_{2.5} 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 PM_{2.5} concentration estimator is obtained.

Drawback

Photo graphic method is not sufficient to calculate PM_{2.5} and it taken only one pollutants of concentration.

The work is given based on only one PM_{2.5} and to collect more monitoring data from other cities to verify the generalization of our work and more factors, e.g., geomorphic conditions.

2.4 PROPOSED SYSTEM

2.4.1 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. 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.

2.4.2 Building the classification model

The predicting the consumer intension by supervised machine learning 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.

2.4.3 Predictive Model Construction:

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 accuracy.

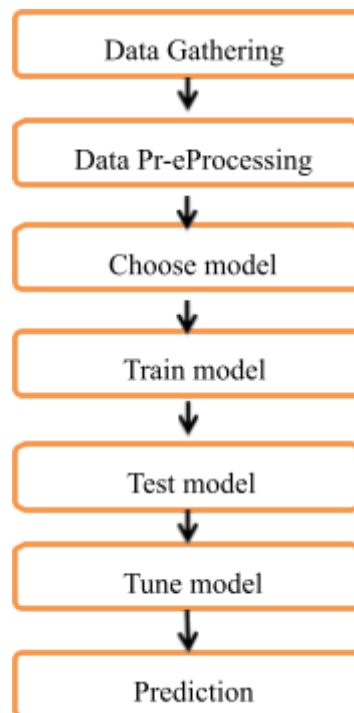


Fig 2.1 Data flow Diagram

CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

Requirements are the basic constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

3.1 PROJECT REQUIREMENTS

3.1.1 Functional Requirements

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

3.1.2 Non-Functional Requirements

Process of functional steps,

Problem define

Preparing data

Evaluating algorithms

Improving results

Prediction the result

3.1.3 Environmental Requirements

Software Requirements

Operating System : Windows

Tool : Anaconda with Jupyter Notebook, xampp

Hardware requirements

Processor : Pentium IV/III

Hard disk : minimum 80 GB

RAM : minimum 2 GB
Frontend

GUI based Tkinter using python language

Backend

Machine Learning using python language

Library Packages

Pandas, Numpy, Sk-learn, Matplotlib, Seaborn

3.2 SOFTWARE DESCRIPTION

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:

JupyterLab

Jupyter Notebook

QtConsole

Spyder

Glueviz

Orange

Rstudio

Visual Studio Code

3.2.1 Conda

Conda is an open source, cross-platform, language agnostic package manager

and environment management system that installs, runs and updates

packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multilanguages. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

3.2.2 The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much

more.

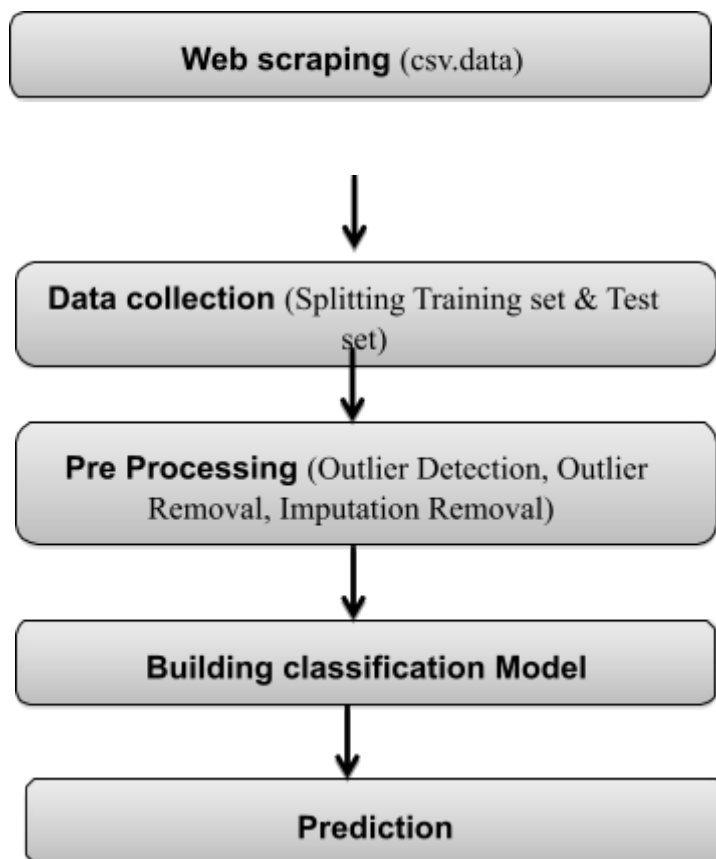


Fig 3.1 Data Flow Diagram For Machine Learning Model

3.2.3 Kernel

A notebook *kernel* is a “computational engine” that executes the code contained in a Notebook document. The *ipython kernel*, referenced in this guide, executes python code. Kernels for many other languages exist. When you open a Notebook document, the associated *kernel* is automatically launched. When the notebook is *executed*, the *kernel* performs the computation and produces the results. Depending on the type of computations, the *kernel* may consume significant CPU and RAM. Note that the RAM is not released until the *kernel* is shut-down.

3.2.4 Notebook Dashboard

The *Notebook Dashboard* is the component which is shown first when you launch Jupyter Notebook App. The *Notebook Dashboard* is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown). The *Notebook Dashboard* has other features similar to a file manager, namely navigating folders and renaming/deleting files.

3.3 WORKING PROCESS

Download and install anaconda and get the most useful package for machine learning in Python.

Load a dataset and understand its structure using statistical summaries and data visualization.

machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming. The best way to get started using Python for machine learning is to complete a project.

It will force you to install and start the Python interpreter (at the very least).

It will give you a bird's eye view of how to step through a small project.

It will give you confidence, maybe to go on to your own small projects.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

3.4 MODULES

Data Validation and Preprocessing Technique (Module-01)

Exploration data analysis of visualization and training a model by given attributes (Module-02)

Performance measurement of Logistic regression and Naïve Bayes (Module-03)

Performance measurements of Support vector classifier and Random Forest algorithm (Module-04)

Performance measurements of KNN and Decision Tree algorithm (Module-05)

GUI based prediction of air quality (Module-06)

3.4.1 Variable Identification process / data validation process (Module-01):

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. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers uses this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. For example, time series data can be analyzed by regression algorithms; classification algorithms can be used to analyze discrete data. (For example to show the data type format of given dataset)

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. Some specified Machine Learning model needs information in a specified format; for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

3.4.2 Exploration of data (Module-02)

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.

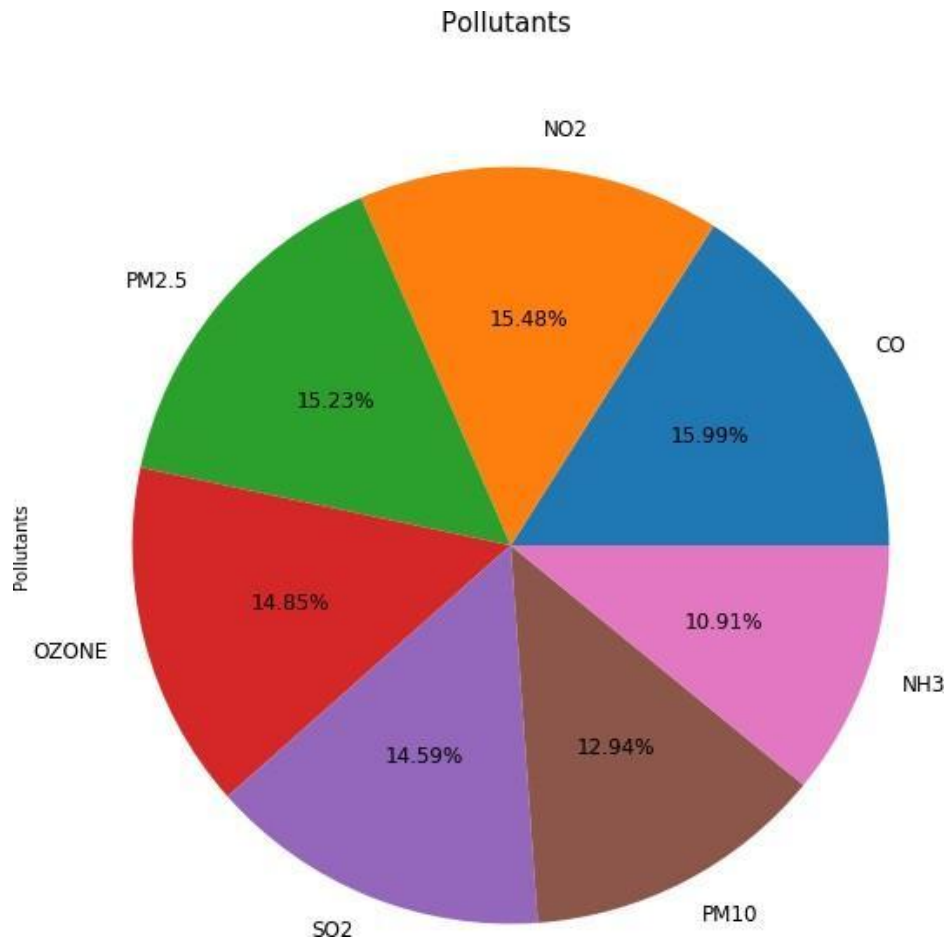


Fig 3.2 Percentage of each pollutants

Many machine learning algorithms are sensitive to the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the training process of machine learning algorithms resulting in longer training times, less accurate models and ultimately poorer results.

Even before predictive models are prepared on training data, outliers can result in misleading representations and in turn misleading interpretations of collected data. Outliers can skew the summary distribution of attribute values in descriptive statistics like mean and standard deviation and in plots such as histograms and scatterplots, compressing the body of the data. Finally, outliers can represent examples of data instances that are relevant to the problem such as anomalies in the case of fraud detection and computer security.

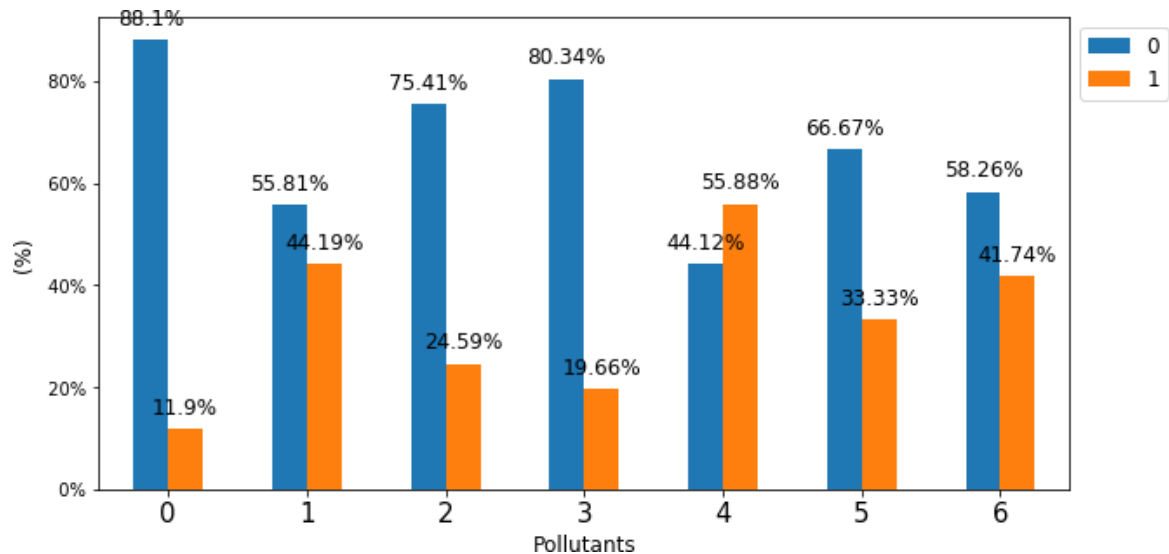


Fig 3.3 Air quality vs No Air quality

It couldn't fit the model on the training data and can't say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

The three steps involved in cross-validation are as follows:

1. Reserve some portion of sample data-set.
2. Using the rest data-set train the model.
3. Test the model using the reserve portion of the data-set.

Advantages of train/test split:

1. This runs K times faster than Leave One Out cross-validation because K- fold cross-validation repeats the train/test split K-times.
2. Simpler to examine the detailed results of the testing process.

Advantages of cross-validation:

1. More accurate estimate of out-of-sample accuracy.
2. More "efficient" use of data as every observation is used for both

training and testing.

Training the Dataset:

The first line imports iris data set which is already predefined in sklearn module and raw data set is basically a table which contains information about various varieties. For example, to import any algorithm and train_test_split class from sklearn and numpy module for use in this program. To encapsulate load_data() method in data_dataset variable. Further divide the dataset into training data and test data using train_test_split method. The X prefix in variable denotes the feature values and y prefix denotes target values.

- This method divides dataset into training and test data randomly in ratio of 67:33 / 70:30. Then we encapsulate any algorithm.
- In the next line, we fit our training data into this algorithm so that computer can get trained using this data. Now the training part is complete.

Testing the Dataset:

- Now, the dimensions of new features in a numpy array called 'n' and it want to predict the species of this features and to do using the predict method which takes this array as input and spits out predicted target value as output.
- So, the predicted target value comes out to be 0. Finally to find the test score which is the ratio of no. of predictions found correct and total predictions made and finding accuracy score method which basically compares the actual values of the test set with the predicted values.

3.4.3 Performance measurement of Logistic regression and Naïve Bayes (Module-03)

LR (Logistic Regression):

It could help to use predict whether the purchase rating increased or not and it predictions are discrete (only specific values or categories are allowed). We can

also view probability scores underlying the model's classifications and it is multi type of classification process.

Steps

Instead of $y=0,1$ (Predicted variables), we will expand our definition so that $y=0,1,...,n$. Basically we re-run binary classification multiple times. Divide the problem of classification of good customer behavior into $n+1$ binary classification problems (+1 because the index starts at 0).

- ☐ For each class...
- ☐ Predict the probability the observations are in that single class.
- ☐ $\text{prediction} = \max(\text{probability of the classes})$

For each sub-problem, we select one class (YES) and lump all the others into a second class (NO). Then we take the class with the highest predicted value of multi class of product purchases.

Naïve Bayes

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically. Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method. The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.

3.4.4 Performance measurement of support vector machine and Random Forest algorithm (Module-04)

SVM (Support Vector Machine) :

Each customer behaviours as a point in n-dimensional space (where n is number of columns) with the value of each column being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes. To separate the two classes of data points, there are many possible hyper- planes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes.

Steps

It representation of different classes in a hyper-plane in multidimensional space. The hyper-plane will be generated in an iterative manner. So, that the error can be minimized and to divide the given datasets into classes to find a maximum marginal hyper-plane (MMH).

It will generate hyper-planes iteratively that segregates the classes in best way.

It will choose the hyper-plane that separates the classes correctly.

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.

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.

3.4.5 Performance measurement of KNN and Decision Tree algorithm (Module-05)

KNN (K-nearest neighbour)

It is done by a majority vote to its neighbors and the data of network connection behaviors is assigned to the class which has the nearest neighbors. To increase the number of nearest neighbors, the value of k, accuracy might increase.

Steps

Load the given dataset and initialize 'k' value by chosen number of neighbors.

Calculate the distance between trained customer behavior's and new customer behavior.

Sort the collection of distances in ascending order of customer behaviors by the distances and pick the first 'k' entries from sorted collection.

So, get the labels of the selected 'k' entries and return the mode of the 'k' labels.

Decision Tree:

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

3.4.6 GUI based air quality (Module-06)

Tkinter is a python library for developing GUI (Graphical User Interfaces). We use the tkinter library for creating an application of UI (User Interface), to create windows and all other graphical user interface and Tkinter will come with Python as a standard package, it can be used for security purpose of each users or accountants. There will be two kinds of pages like registration user purpose and login entry purpose of users.

Accuracy calculation:

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class is no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

Comparing Algorithm with prediction in the form of best accuracy result:

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine

learning algorithms in order to choose the one or two to finalize. A way to do this is to use

different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 6 different algorithms are compared:

Logistic Regression

Naïve Bayes

Support Vector Machine

Random Forest

K-Nearest Algorithm

Decision Tree Algorithm

- Now, the dimensions of new features in a numpy array called 'n' and it want to predict the species of this features and to do using the predict method which takes this array as input and spits out predicted target value as output.
- So, the predicted target value comes out to be 0. Finally to find the test score which is the ratio of no. of predictions found correct and total predictions made and finding accuracy score method which basically compares the actual values of the test set with the predicted values.

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes

and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate(TPR) = $TP / (TP + FN)$

False Positive rate(FPR) = $FP / (FP + TN)$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct. (When

the model predicts default: how often is correct?)

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

General Formula:

$$\text{F-Measure} = 2\text{TP} / (2\text{TP} + \text{FP} + \text{FN})$$

F1-Score Formula:

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

CHAPTER 4

RESULTS AND DISCUSSION

The data obtain from online resources and then we save it as csv format and process it then after we train some data and test it so that it will learn itself and we can predict the output using this data by applying the ml algorithms and create a graph to represent it.

TRAINING DATA:

	Country	State	city	place	lastupdate	Avg	Max	Min	Pollutants
0	India	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	21-12-2018 03:00:00	70.0	108.0	42.0	PM2.5
1	India	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	21-12-2018 03:00:00	76.0	102.0	43.0	PM10
2	India	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	21-12-2018 03:00:00	73.0	118.0	46.0	NO2
3	India	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	21-12-2018 03:00:00	5.0	6.0	4.0	NH3
4	India	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	21-12-2018 03:00:00	41.0	109.0	2.0	SO2

Fig 4.1 Training Data

The above Fig 4.1 represent the training data and test data it will be like 0.7 as training data and 0.3 as testing data where by using this data it will learn to do as for the new data and predict it.

CONFUSION MATRIX RESULT:

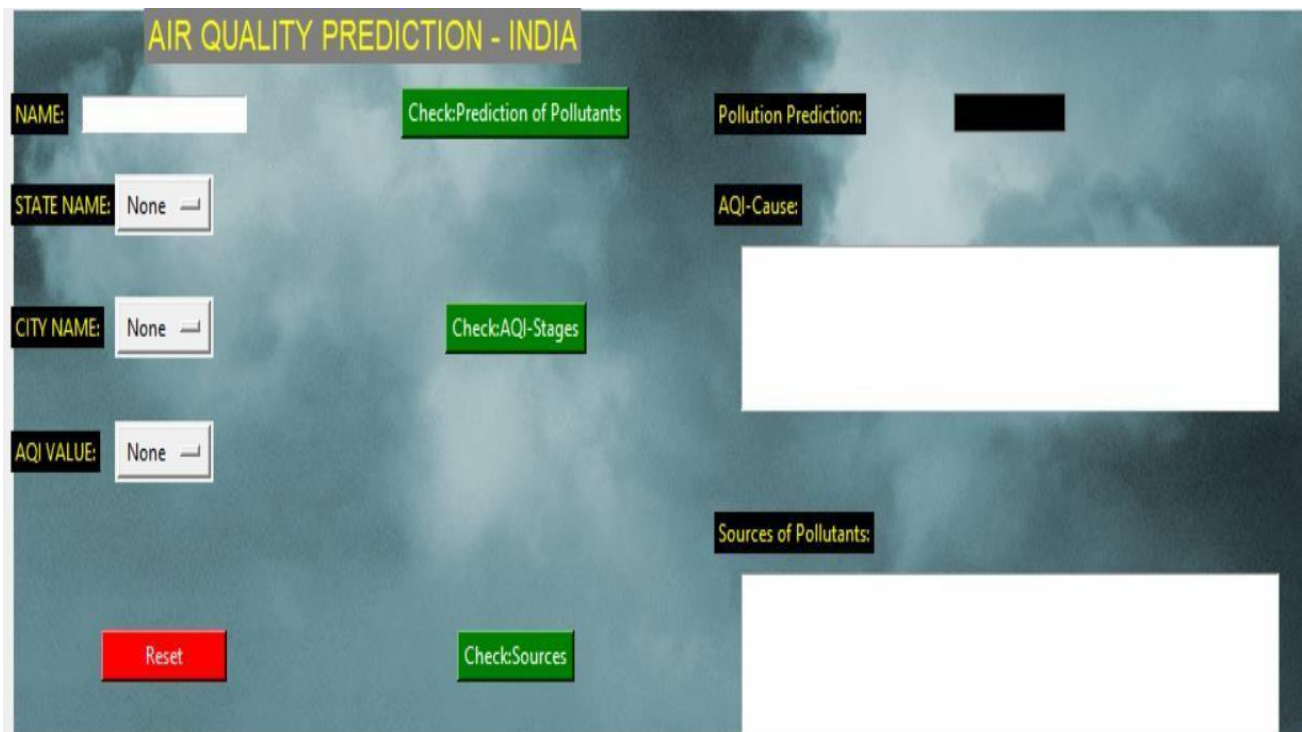
Classification report of Decision Tree Classifier Results:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	73
1	1.00	1.00	1.00	175
accuracy			1.00	248
macro avg	1.00	1.00	1.00	248
weighted avg	1.00	1.00	1.00	248

Fig 4.2 Confusion Matrix Result

The above Fig 4.2 represent the result of the confusion matrix like it is the output for every algorithm you used in your machine learning where it gives results of accuracy, specificity, precision, recall and averages for every factor there will be a specified formula by using the formula all the other algorithms also can predicted.

AIR INPUT PAGE



The screenshot displays the 'AIR QUALITY PREDICTION - INDIA' GUI. It features a dark blue background with a cloudy sky image. On the left, there are four input fields: 'NAME:' with a text box, 'STATE NAME:' with a dropdown menu showing 'None', 'CITY NAME:' with a dropdown menu showing 'None', and 'AQI VALUE:' with a dropdown menu showing 'None'. Below these is a red 'Reset' button. In the center, there are two green buttons: 'Check:Prediction of Pollutants' at the top and 'Check:AQI-Stages' below it. At the bottom center is a green 'Check:Sources' button. On the right, there are two output areas: 'Pollution Prediction:' with a black rectangular box, and 'AQI-Cause:' with a large white rectangular box. Below these is a label 'Sources of Pollutants:' followed by another large white rectangular box.

Fig 4.3 Input Page (GUI Interface)

The above figure, states that we have to give the input for the respective data such as Name, State name, City name, AQI Value.

AIR OUTPUT PAGE

The screenshot displays the 'AIR QUALITY PREDICTION - INDIA' application. On the left, there are four input fields: 'NAME:' with 'India', 'STATE NAME:' with 'TamilNadu', 'CITY NAME:' with 'Chennai', and 'AQI VALUE:' with 'Above_175'. Below these is a red 'Reset' button. To the right of the inputs are two green buttons: 'Check:Prediction of Pollutants' and 'Check:AQI-Stages'. Further right, the 'Pollution Prediction:' field shows 'OZONE'. Below it, the 'AQI-Cause:' field displays a text box with the message: '(Moderate)-May cause breathing discomfort to people with lung disease such as asthma, and discomfort to people with heart disease, children and older adults.' At the bottom right, the 'Sources of Pollutants:' field shows a text box with the message: 'Pollutants emitted by cars, Power plants, Industrial boilers, Refineries, Chemical plants, Other sources react chemically in the presence of sunlight.' A green 'Check:Sources' button is located at the bottom center.

Fig 4.4 Expected Output on GUI Interface

The above fig 4.5 is about the pollutants in the respective area and the causes, adverse effect of the pollutants and the source of pollutants. Fig 4.4 is the final result.

Here are some few screenshots regarding different states.

This screenshot shows the same application but with different inputs for Karnataka. The 'NAME:' field is 'INDIA', 'STATE NAME:' is 'Karnataka', 'CITY NAME:' is 'Bengaluru', and 'AQI VALUE:' is 'Above_175'. The 'Check:Prediction of Pollutants' button is now active, and the 'Pollution Prediction:' field shows 'PM10'. The 'AQI-Cause:' field displays the same message as in Fig 4.4. The 'Sources of Pollutants:' field shows a new message: 'Burning of fuels, Wood stoves, Fires, Wind generated dust, Brick works, Refineries, Cement Works, Iron and Steel making.' The 'Check:AQI-Stages' and 'Check:Sources' buttons remain visible.

Fig 4.5 Air Quality for state Karnataka

AIR QUALITY PREDICTION - INDIA

NAME:

Pollution Prediction:

STATE NAME:

CITY NAME:

AQI VALUE:

(Moderate)-May cause breathing discomfort to people with lung disease such as asthma, and discomfort to people with heart disease, children and older adults.

Sources of Pollutants:

Burning of fossil fuels by power plants and Other industrial facilities.

Fig 4.6 Air Quality for state Andhra Pradesh

AIR QUALITY PREDICTION - INDIA

NAME:

Pollution Prediction:

STATE NAME:

CITY NAME:

AQI VALUE:

(Poor)-May cause breathing discomfort to people on prolonged exposure, and discomfort to people with heart disease.

Sources of Pollutants:

Combustion, Fireplaces, Car engines and Coal or Natural gas fired power plants.

Fig 4.7 Air Quality for state Maharashtra

CH

APTER 5

5.1 CONCLUSION

CONCLUSION

AND FUTURE

WORK

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. This application can help India meteorological department in predicting the future of air quality and its status and depends on that they can take action.

5.2 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.

REFERENCES

- [1] A. W. Joshi, "When does customer orientation hinder (Help) radical product innovation? The role of organizational rewards," *J. Product Innovation Manage.*, vol. 33, no. 4, pp. 435–454, 2018.
- [2] B. Schlender and R. Tetzeli, *Becoming Steve Jobs: The Evolution of a Reckless Upstart into a Visionary Leader*. London, U.K.: HodderStoughton, 2018.
- [3] W. Biemans and F. Langerak, "More research priorities," *J. Product Innovation Manage.*, vol. 32, no. 1, pp. 2–3, 2017.
- [4] F. Eggers, S. Kraus, and J. G. Covin, "Traveling into unexplored territory: Radical innovativeness and the role of networking, customers, and technologically turbulent environments," *Ind. Marketing Manage.*, vol. 43, no. 8, pp. 1385–1393, 2017.
- [5] G. Barczak, "The future of NPD/innovation research," *J. Product Innovation Manage.*, vol. 29, no. 3, pp. 355–357, May 2016.
- [6] Y. M. Antorini, A. M. J. Muniz, and T. Askildsen, "Collaborating with customer communities: Lessons from the lego group," *MIT Sloan Manage. Rev.*, vol. 53, no. 3, pp. 73–79, 2016.
- [7] M. K. Poetz and M. Schreier, "The value of crowdsourcing: Can users really compete with professionals in generating new product ideas?," *J. Product Innovation Manage.*, vol. 29, no. 2, pp. 245–256, Mar. 2016.
- [8] R. J. Calantone, N. Harmancioglu, and C. Droge, "Inconclusive innovation 'returns': A meta-analysis of research on innovation in new product development," *J. Product Innovation Manage.*, vol. 27, no. 7, pp. 1065– 1081, 2015.

APPENDIX

A. SOURCE CODE

```
#import library packages
import pandas as p
import numpy as n
#Load given dataset
data = p.read_excel("AirQuality.xlsx")
df=data.dropna()
#shape
df.shape
#show columns
df.columns
#To describe the dataframe
df.describe()
#Checking datatype and information about dataset
df.info()
#Checking for duplicate data
df.duplicated()
#find sum of duplicate data
sum(df.duplicated())
#Checking sum of missing values
df.isnull().sum()
#Checking minimum or maximum of average pollution values
print("Minimum value of average pollution is:", df.Avg.min())
print("Maximum value of average pollution is:", df.Avg.max())
#check the average pollution range
print("Avg pollution range :", sorted(df['Avg'].unique()))
```

```
PMkitchen = df.loc[df['Pollutants'] == 'PM2.5']
```

```

PMclean = df.loc[df['Pollutants'] ==
'PM10'] N02 = df.loc[df['Pollutants'] ==
'NO2']
NH3 = df.loc[df['Pollutants'] ==
'NH3'] S02 = df.loc[df['Pollutants'] ==
'SO2'] CO = df.loc[df['Pollutants'] ==
'CO']
O3 = df.loc[df['Pollutants'] == 'OZONE']
p.Categorical(df['Pollutants']).describe()
df.corr()

from sklearn.preprocessing import LabelEncoder
var_mod = ['Country', 'State', 'city', 'place', 'lastupdate', 'Avg', 'Max', 'Min', 'Pollutants']
le = LabelEncoder()
for i in var_mod:
    df[i] = le.fit_transform(df[i]).astype(str)
p.crosstab(df.city, df.Pollutants)
fig, ax = plt.subplots(figsize = (16,10))
ax = s.barplot(x='Pollutants', y='Avg', data=df)
ax.set(ylabel = 'Pollutants details', title = 'Pollutants by average pollution values')
plt.show()

#Propagation by variable
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' \n', fontsize = 15)
    return n.round(dataframe_pie/df.shape[0]*100,2)
PropByVar(df, 'Pollutants')
df.boxplot(column="Avg", by="Pollutants")

```

```
def qul_No_qul_bar_plot(df, bygroup):
```

```
    dataframe_by_Group = p.crosstab(df[bygroup], columns=df["class"], normalize = 'index')
```

```

dataframe_by_Group = n.round((dataframe_by_Group * 100), decimals=2)
ax = dataframe_by_Group.plot.bar(figsize=(10,5));
vals = ax.get_yticks()
ax.set_yticklabels(['{:3.0f}%'.format(x) for x in vals]);
ax.set_xticklabels(dataframe_by_Group.index,rotation = 0, fontsize = 15);
ax.set_title('Air Quality Vs No Air Quality (%) (by ' + dataframe_by_Group.index.name +
')\n', fontsize = 15)
ax.set_xlabel(dataframe_by_Group.index.name, fontsize = 12)
ax.set_ylabel('(%)', fontsize = 12)
ax.legend(loc = 'upper left',bbox_to_anchor=(1.0,1.0), fontsize= 12)
rects = ax.patches

# Add Data Labels

for rect in rects:
    height = rect.get_height()
    ax.text(rect.get_x() + rect.get_width()/2,
            height + 2,
            str(height)+'%',
            , ha='center',
            va='bottom',
            fontsize = 12)

return dataframe_by_Group

#According to the cross-validated MCC scores, the random forest is the best-performing
model, so now let's evaluate its performance on the test set.

from sklearn.metrics import confusion_matrix, classification_report, matthews_corrcoef,
cohen_kappa_score, accuracy_score, average_precision_score, roc_auc_score
X = df.drop(labels='class', axis=1)

#Response variable

```

```
y = df.loc[:, 'class']
```

```

del df

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very
important to do because there are so few fraudulent transactions.

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1,
stratify=y)

#for our convenient we delete X,y variable for differentiate confusion
del X, y

# Prevent view warnings
X_train.is_copy = False
X_test.is_copy = False

from sklearn.linear_model import LogisticRegression
logR= LogisticRegression()

logR.fit(X_train,y_train)

predictR = logR.predict(X_test)

print("")
print('Classification report of Logistic Regression Results:')
print("")

print(classification_report(y_test,predictR))
x = (accuracy_score(y_test,predictR)*100)

print('Accuracy result of Logistic Regression is:', x)
print("")
cm1=confusion_matrix(y_test,predictR)

```

```
print('Confusion Matrix result of Logistic Regression is:\n',cm1)
```

```

sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', sensitivity1 )
print("")

specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', specificity1)
print("")

from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()

dtree.fit(X_train, y_train)

predictDT = dtree.predict(X_test)

print("")
print('Classification report of Decision Tree Classifier Results:')
print("")

print(classification_report(y_test,predictDT))
x = (accuracy_score(y_test,predictDT)*100)

print('Accuracy result of Decision Tree Classifier is', x)
print("")
cm2=confusion_matrix(y_test,predictDT)
print('Confusion Matrix result of Decision Tree Classifier is:\n',
confusion_matrix(y_test,predictDT))
print("")

```

```
sensitivity1 = cm2[0,0]/(cm2[0,0]+cm2[0,1])  
print('Sensitivity : ', sensitivity1 )
```

```

specificity1 = cm2[1,1]/(cm2[1,0]+cm2[1,1])
print('Specificity : ', specificity1)
TN = cm2[0][0]
FN = cm2[1][0]
TP = cm2[1][1]
FP = cm2[0][1]
print("True Positive :",TP)
print("True Negative :",TN)
print("False Positive :",FP)
print("False Negative :",FN)
print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN)
FNR = FN/(TP+FN)
print("True Positive Rate :",TPR)
print("True Negative Rate :",TNR)
print("False Positive Rate :",FPR)
print("False Negative Rate :",FNR)
print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()

rfc.fit(X_train,y_train)

```

```

predictR = rfc.predict(X_test)

print("")
print('Classification report of Random Forest Results:')
print("")

print(classification_report(y_test,predictR))
x = (accuracy_score(y_test,predictR)*100)

print('Accuracy result of Random Forest is:', x)
print("")
cm1=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Random Forest is:\n',cm1)
print("")
sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', specificity1)
print("")
from sklearn.svm import SVC
s = SVC()

s.fit(X_train,y_train)

predicts = s.predict(X_test)

print("")

```

```
print('Classification report of Support Vector Machines Results:')
```

```

print("")

print(classification_report(y_test,predicts))
x = (accuracy_score(y_test,predicts)*100)

print('Accuracy result of Support Vector Machines is:', x)
print("")
cm2=confusion_matrix(y_test,predicts)
print('Confusion Matrix result of Support Vector Machines is:\n',cm2)
print("")
sensitivity1 = cm2[0,0]/(cm2[0,0]+cm2[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm2[1,1]/(cm2[1,0]+cm2[1,1])
print('Specificity : ', specificity1)
print("")
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()

gnb.fit(X_train,y_train)

predictR = gnb.predict(X_test)

print("")
print('Classification report of Naive Bayes Results:')
print("")

print(classification_report(y_test,predictR))

```

```
x = (accuracy_score(y_test,predictR)*100)
```

```

print('Accuracy result of Naive Bayes is:', x)
print("")
cm1=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Naive Bayes is:\n',cm1)
print("")
sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', specificity1)
print("")
from sklearn.neighbors import KNeighborsClassifier
knnc = KNeighborsClassifier()

knnc.fit(X_train,y_train)

predictR = knnc.predict(X_test)

print("")
print('Classification report of K-Nearest Neighbor Results:')
print("")
print(classification_report(y_test,predictR))
x = (accuracy_score(y_test,predictR)*100)

print('Accuracy result of K-Nearest Neighbor is:', x)
print("")
cm2=confusion_matrix(y_test,predictR)

```

```
print('Confusion Matrix result of K-Nearest Neighbor is:\n',cm2)
```

```
print("")
sensitivity1 = cm2[0,0]/(cm2[0,0]+cm2[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm2[1,1]/(cm2[1,0]+cm2[1,1])
print('Specificity : ', specificity1)
print("")
```

B. SCREENSHOTS

AIR QUALITY PREDICTION - INDIA

NAME: Check:Prediction of Pollutants Pollution Prediction:

STATE NAME: AQI-Cause:

CITY NAME: Check:AQI-Stages

AQI VALUE: Sources of Pollutants:

AIR QUALITY PREDICTION - INDIA

NAME: Check:Prediction of Pollutants Pollution Prediction:

STATE NAME: AQI-Cause:

(Good)-Minimal impact

CITY NAME: Check:AQI-Stages

AQI VALUE: Sources of Pollutants:

Burning of fossil fuels by power plants and Other industrial facilities.

C. PUBLICATION

GUI BASED AIR QUALITY PREDICTION BASED ON SUPERVISED MACHINE LEARNING APPROACH

KARAN PARSI [1], KATHA PRUDHVINATH REDDY [2] , ASHA P[3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3]Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

ABSTRACT: By and large, fog alludes to the arrival of the poisons up in air that are unfavourable to person wellbeing and the world in general. This tends to a depicted as quite possibly the max perilous dangers that mankind at any point confronted. It makes harm creatures, crops, woods, so on to forestall this issue around transport areas need to foresee air raid from toxins manoeuvring AI procedures. In subsequential, air raid assessment and forecast have become a significant exploration region. Our point is to explore AI based procedures for air raid guaging by expectation brings about best value. The breakdown of dataset by supervised machine learning (SMLT) where the data caught are cleanly organised by breaking down the data in a 70-30 manner and investigation is done on full dataset. Our investigation gives an extensive major change in values by examining the model boundaries with respect to completing in forecast of air raid pollution by rigor count. To put forward an AI based strategy by precisely anticipate the Air Quality Index(AQI) an incentive alongside expectation brings about the type of benefit exactness in contrasting direct order AI calculations. Also, to analyze and checking the various AI numbers we can analyze the GUI made UI air raid expectation by credits.

KEYWORDS: ML- Machine learning, dataset, GUI results, Classification, python.

I INTRODUCTION

Air pollution demonstrating with profound learning is another idea. The future

forecast of air toxins directed with profound learning may arrive at almost careful outcomes. With the assistance of

profound learning the absence of importance to air pollution guaging might be diminished and, around there, bigger ventures might be created. Since the mechanical upheaval air nature of the world has been evolving quickly. The vast majority of the modern exercises

discharges gigantic measures of poisonous or destructive pollutions to the environment, for example, SO₂, NO₂, CO, PM and harmful organics. The majority of these air poisons influences individuals genuinely alongside creatures and plants. For instance, respiratory, cardiological and pneumonic issues by inward breath are a portion of these issues made via air poisons. They may likewise prompt more major issues influencing entire world, for example, an Earth-wide temperature boost and environmental change. At the point when all negative impacts of air pollutions considered, it is vital to show air pollution to decide future focuses and toxin sources.

The climate is getting dirtied because of human exercises and cataclysmic event, serious among them is air pollution. The grouping of air toxins in encompassing air is represented by the meteorological boundaries, for example, air wind speed, wind heading, relative mugginess, and temperature. In the event that the dampness is more, we feel a lot sultrier on the grounds that sweat won't dissipate into the air. Urbanization is one of the principal explanations behind air pollution since, increment in the transportation offices transmits more toxins into the environment and another fundamental purpose behind air pollution is Industrialization. The significant poisons are Nitrogen Oxide (NO), Carbon Monoxide (CO), Particulate matter (PM), SO₂ and so forth Carbon Monoxide is delivered because of the insufficient Oxidization of fuel, for example, oil, gas, and so forth Nitrogen Oxide is created

because of the start of warm fuel; Carbon

monoxide causes cerebral pains, spewing; Benzene is delivered because of smoking, it causes respiratory issues; Nitrogen oxides causes unsteadiness, queasiness; Particulate matter with a breadth 2.5 micrometer or not exactly that influences more to human wellbeing. Measures should be taken to limit air smog, in the climate.

Air Quality Index(AQI), is utilized to gauge the nature of air. Prior traditional strategies, for example, likelihood, insights were utilized to anticipate the nature of air, however those techniques are mind boggling to foresee the nature of air. Because of progression of innovation, presently it is not difficult to bring the information about the pollutions of air utilizing sensors. Appraisal of crude information to identify the toxins needs incredible investigation. Convolution Neural organizations, Recursive Neural organizations, Deep Learning, Machine learning calculations guarantees in achieving the expectation of future AQI so that measures can be taken properly. PM2.5 is naturally called fine particulate matter. Fine particulate matter, this is basic among the pollution list since it is a significant touch on to people's prosperity when its height perceptible all around is modestly high. PM2.5 also implies little flakes observable all around that decrease detectable quality and make the air to seem as to be faint when heights are raised. Unmistakable AI techniques have been suplicated perceive air smog and envision PM2.5 heights subject to

an enlightening list including step by step cools. A cushy deduction structure was

introduced in[4] to perform limit game plan using a reasoning about cooperation and

consolidating it into an air smog record. Because of weakness of the actual value PM2.5 height, the issue is improved to be a twofold course of action 1, which is to arrange the PM2.5 height in to "High" ($> 115 \text{ ug/m}^3$) and "low" ($\leq 115 \text{ ug/m}^3$). The constraint is picked reliant on the Air Smog Level surface, which set 115 ug/m^3 to be smooth level corruption [2]. the current structures perceive the air idea of a wished to be city picked by an customer and social events it into dissimilar arrangements like incredible, acceptable, midsize, bust, astoundingly bust, genuine reliant on AQI (Air Quality Index). Thus data is appeared as a month to month, after a long time after week or standard timetable. The characteristics are perceived for the whole city, and can't be checked for the precision of the decided characteristics a brief timeframe later. The proposed structure finishes two endeavors the first one, Distinguishes the levels of PM2.5 reliant on given barometrical characteristics. Determined backslide is used to recognize if a data test is either dirtied sullied. There are a few displaying strategies have been actualized for this reason. Counterfeit neural organizations are one of the generally picked methods which dependent on AI. Profound learning is sub-group of AIs. It conveys counterfeit neural organizations one stage past with utilizing enormous informational collection, taking care of issues without isolating, utilizing more layers, handling all the while with successive layers and giving more trustable outcomes. All these positive highlights of profound learning make it reasonable technique for air pollution

demonstrating. To create fruitful

model, a few stages ought to be followed. In this work, demonstrating with profound learning designs on genuine air pollution information was clarified in detail by considering the new advances around there and the distinctions with other man-made brainpower models.

nitrogen oxides. They have considered the factors, for example, the speed of the air, air bearing, temperature, and dampness and the harmful components from the modern site, for example, Skikda. They have

II RELATED WORK

Dixian et.al [1] utilized the Bias organization to discover the air quality and framed DAG from the informational collection of the town called as shanghai. The dataset is plunged for the preparation and testing model. The inconvenience of this methodology is they have not thought about topographical and social climate attributes, so the outcomes may change dependent on these variables.

Gong et al [2] Introduced the most awesome aspect our insight, the nearest to their work is multitasked PM2.5 expectation utilizing drawn out momentary memory cartridge organization expanded bot. His contend that current measurable and AI models for foreseeing air poison fixation neglect to successfully display long haul conditions. Furthermore, it was additionally indicated that the utilization of assistant information improved model execution.

Sachit et.al [3] proposed fake based Regressive model which is nonlinear to foresee 2 significant air toxins 2, for example, carbon monoxide and

utilized RMSE and MAE to decide the exhibition, yet this strategy considered just 2 toxins, for example, NO and CO the other significant pollutions, for example, sulfur dioxide, PM2.5, PM10 are not thought of. Jose et.al [10] proposed an IoT based method to acquire air quality informational collection. They have utilized Long Short-term Memory [LSTM] procedure to foresee the air quality the proposed method accomplished better precision by decreasing the time taken to prepare the model. Yet at the same time, the exactness can be improved by looked at different strategies.

exactness's of a various

Pandey et.al [9] described the benefits of the Bidirectional Long - Short `Memory [BiLSTM] method to forecast the severity of air pollution. The proposed technique achieved better prediction which models the far term, near term, with the critical consequence of the PM2.5 severity levels. In a proposed method prediction is made at 6h, 12h, 24h. The results obtained for 12h is consistent, but the result obtained for 6h, and 24h are not consistent.

Caselli et al[11] tended to the issue of a foreseeing ozonosphere and PM10 for the city of Milan, Italy utilizing pruned neural and feed-forward cartridge and LL. PNNs comprise a boundary stingy methodology, in light of the expulsion of repetitive boundaries from completely associated neural organizations; LL, then again, is a neighborhood straight expectation calculation, which plays out a nearby learning system every time a forecast is decided to investigate. No huge contrasts were open up flanked by the conjecture

modules. Ioannis et al[12] introduced an online air smog anticipating framework within Greater Istanbul Area. These framework anticipated three air smog problem pointer levels for the following three days will be +1, +2, and +3 days utilizing neural organizations. It was demonstrated that aggregate forecast utilizing earlier day's anticipated qualities delivered preferable outcomes over autonomous expectations. Significance of day of the week as an information boundary was likewise examined.

solitary record utilized by Environmental organizations to report every day air

III PROBLEM STATEMENT

Monitoring and saving air quality have gotten quite possibly the most fundamental investigation in several civic and also the metropolitan regions this very day. The quintessence of gale is antagonistically influenced in account of different types of smog immersed about by sending materials, virtuosity, fuel use in and suchlike the testimony of destructive murk is making a veridical danger for the especial consummation in keen civic communities. With beefing up air pollution, we need to execute proficient air parameter observing bot which bring up the data about the team of air ills and give reckoning of air smog in every barony.

IV EXISTING SYSTEM

Existing measurable and AI models for anticipating air poison focus neglect to successfully show long haul conditions. Moreover, it was likewise demonstrated that the utilization of helper information improved model execution. It is a

parameter. The huskier the AQI esteem, the more prominent the degree of gale smog and the more noteworthy the wellbeing perplex. Since AQI is figured as a 24-hour normal, it's anything but an appropriate variable for hourly forecast. Or maybe anticipating PM2.5 or PM10 fills in as helpful strand for foreseeing pollution heights, whereas one of their gales is compulsory promoting registering AQI. The information is shown on a month to month, week after week or regular routine. Likewise, when the qualities are anticipated, the qualities don't change regarding the abrupt change in the air conditions or startling expansion in rush hour gridlock. The values are distinguished for the entire city, and can't be confirmed for the precision of the anticipated qualities a while later.

V PROPOSED SYSTEM

The proposing framework is to examine a dataset of air poisons records for India meteorological area utilizing AI procedure. To recognizing air quality is more troublesome and attempt to lessen this danger factor behind anticipating from AQI of India to safe human to save bunches of meteorological endeavors and resources and to foresee whether allocating the air quality is awful or acceptable dependent on given ascribes. The proposed framework completes two assignments that distinguishes the degrees of water level dependent on given barometrical qualities and predicts the degree of water level for a specific date. Strategic relapse is utilized to identify if an information test is either dirtied contaminated.

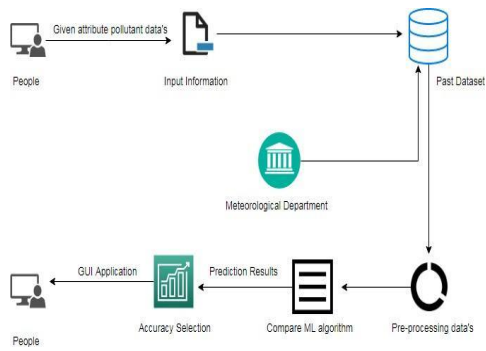


FIG 1 OVERVIEW OF THE PROPOSED SYSTEM

While applying photograph realistic based technique is basic to assess boundaries and it's taken information size is high. To conquer this strategy to actualize AI approach by UI of GUI application. Numerous data procurement from various founts would be joined to frame a summed up data procurement, and afterward unique AI calculations would be applied to separate examples and to acquire results with most extreme exactness.

VI MODULE DESCRIPTION

6.1.1 DATA WRANGLING

In this segment of the report will stack in the findings, check up on tidiness, and afterward retrenching and clean given data procurement up with examination (Fig 2 input data). Ensure that the record steps cautiously and legitimize for cleaning choices.

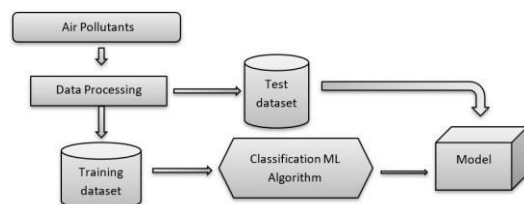


FIG 2 MODULE

DESCRIPTION

6.1.2 DATA COLLECTION

Fig 2 presents the Data Collection. The informational index gathered for anticipating given information is subsection into Training set and Test set. For the most part, 7:3 proportions are applied to part the investigation set and investigating set. Data Model which was made utilizing Random Forest, and calculated relapse are applied on the investigating set and dependent on the investigated outcome exactness, Investigated set forecast is finished.

6.1.3 DATA PRE-PROCESSING

Thus the Data which has been gathered may engross unlocatable qualities that would prompt irregularity. Then we get appropriate forecast result, the dataset should be cleaned, missing qualities should be dealt with either by erasing or by loading up with mean qualities or some other technique. Fig 2 describes about the data preprocessing. Additionally, repetitive information should be taken out or dispensed with to try not to inclination of the outcomes. Some dataset may have some anomaly or extraordinary qualities which likewise must be eliminated to get great expectation precision. Grouping and bunching calculations and other information mining techniques will function admirably just if such a lot of preprocessing is done on the information.

6.1.4 BUILDING UP THE MODEL CLASSIFICATION

The anticipating the gale parameter issue, motley tree calculation expectation model is compelling a direct result of the

accompanying reasons: It gives better outcomes in characterization issue. It is

solid in preprocessing exceptions, unessential factors, and a blend of persistent, all out and discrete factors. It creates out of pack gauge mistake which has demonstrated to be impartial in numerous tests and it is generally simple to tune with. (Fig 2 Building the Classification Model)

6.1.5 CONSTRUCTION OF A PREDICTIVE MODEL

Fig 2 includes the construction of a predictive model. AI needs information gathering have parcel of past information. Information jamboree have middling authentic information and crude information. Before information pre- handling, crude information wont be utilized straightforwardly. It is utilized for pre-process at that point, which sort come up on a calculation with model. Preparing and testing this model operable and anticipating deftly with least blunders. Rehashed model required by tuned crunch to crunch with improving their exactness.

VII FINAL OUTCOME

Thus, after investigating all the algorithms and the process, the final outcome is displayed on the GUI.

Classification report of Decision Tree Classifier Results:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	73
1	1.00	1.00	1.00	175
accuracy			1.00	248
macro avg	1.00	1.00	1.00	248
weighted avg	1.00	1.00	1.00	248

FIG 3 CONFUSION MATRIX RESULT

```
Accuracy result of Decision Tree Classifier is: 99.88095238095237
Confusion Matrix result of Decision Tree Classifier is:
[[ 73  0]
 [ 0 175]]
Sensitivity : 1.0
Specificity : 1.0
```

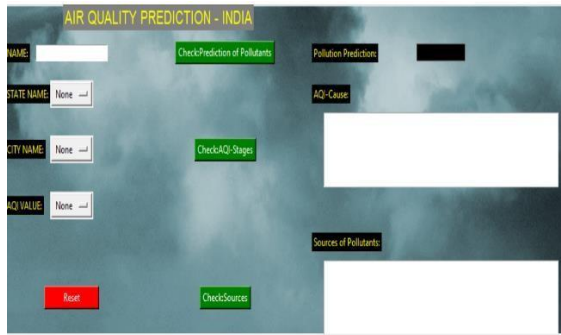


FIG 4 INPUT PAGE (GUI RESULT)



FIG 5 EXPECTED OUTPUT ON GUI

VIII CONCLUSION

The logical cycle started from deets vacuuming and tooling, jejune worth, preludial investigation in final technique structure and reckoning. The primo clearness on open investigated set is higher heed status for anticipating the air quality by given ascribes. This vaporware can help India meteorological office up with anticipating the fate of air smog and its laurels and relies upon that they are able to make a migration. In this proposed work the forecast of PM2.5 is finished. Expectation is one of the utilizations of counterfeit neural organizations. The fundamental target of this examination was to build up the model to anticipate of PM2.5. The created model can be utilized as a dynamic device to make early notice about the pollution of air in the specific

territory. In view of the investigation, the

model having move work in the neural organization structure creates the best presentation in the forecast of air quality contrasted with the organization structure that utilizations move work dependent on the estimations of R and the expectation precision. This model produces R of 0.965 which shows a decent arrangement between the objectives and anticipated yields. Be that as it may, this model created great outcomes for air quality guaging. This kind of model is basic and cost proficient, the model ability is related with their presentation. The created model is more solid for metropolitan air quality portrayal.

At San Francisco, USA [2017]

[4]. E. Kalapanidas and N. Avouris, "Applying machine learning techniques in

REFERENCES:

- [1] Dixian Zhu, Changjie Cai, Tianbao Yang and Xun Zhou: A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization. Big data and cognitive computing, Big Data Cogn. Comput. 2018, 2, 5; doi:10.3390/bdcc2010005
- [2] J. He, S. Gong, Y. Yu, L. Yu, L. Wu, H. Mao, C. Song, S. Zhao, H. Liu, X. Li et al., "Air pollution characteristics and their relation to meteorological conditions during 2014–2015 in major chinese cities," Environmental pollution, vol. 223, pp. 484–496, 2017.
- [3] Sachit Mahajan, Ling-Jyh Chen, Tzu- Chieh Tsai: An Empirical Study of PM2.5 Forecasting Using neural network. IEEE Smart World Congress,

air quality prediction,” in Proc. ACAI, vol. 99, September 2017.

[5] . T. Chiwewe and J. Ditsela, “Machine learning based estimation of Ozone using spatio-temporal data from air quality monitoring stations,” presented at 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), IEEE, 2016.

[6] . Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, “Forecasting fine-grained air quality based on big data,” in Proc. The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2267- 2276, August 10, 2015.

[7]. D. J. Nowak, D. E. Crane, and J. C. Stevens, “Air pollution removal by urban trees and shrubs in the United States,” Urban Forestry & Urban Greening, vol. 4, no. 3, pp. 115-123, 2014`.

[8]. V. M. Niharika and P. S. Rao, “A survey on air quality forecasting techniques,” International Journal of Computer Science and Information Technologies, vol. 5, no. 1, pp.103-107, 2014.

[9] Pandey, Gaurav, Bin Zhang, and Le Jian. " Predicting sub-micron air pollution indicators: a machine learning approach." ; Environmental Science: Processes & amp; Impacts 15.5 (2013): 996-1005.

[10] José Juan Carbajal-Hernándezab Luis P.Sánchez-Fernándeza Jesús A.Carrasco-OchoabJosé Fco.Martínez-Trinidadb: Assessment and prediction of

air quality using fuzzy logic and autoregressive models: (2012) Doi

:<https://doi.org/10.1016/j.atmosenv.2012.06.004>

[11] M. Caselli, L. Trizio , G. de Gennaro, P. Ielpo. "A Simple Feedforward Neural Network for the PM10 Forecasting: Comparison with a Radial Basis Function Network and a Multivariate Linear Regression Model" *Water Air Soil Pollut* (2009) 201:365–377.

[12] Ioannis N. Athanasiadis, "Classification techniques for air quality forecasting", Fifth ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence, 17th European Conference on Artificial Intelligence, Riva del Garda, Italy, August 2006.

[13] Athanasiadis, Ioannis N., et al. "Applying machine learning techniques on air quality data for real-time decision support." 1st NAISO symposium on information technologies in environmental engineering (ITEE'2003), Gdansk, Poland. 2003.