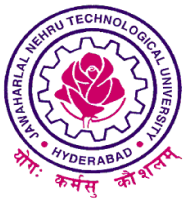
A MACHINE LEARNING MODEL FOR AIR QUALITY PREDICTION FOR SMART CITIES

A Major Project Report

Submitted to



Jawaharlal Nehru Technological University, Hyderabad

In partial fulfillment of the requirements for the

award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

PAMULPARTHI SINDHU	**(18VE1A0543)**
THIRUPATHI NITHIN	**(18VE1A0551)**
VEMUGANTI MEGHANA	**(18VE1A0554)**
YELLA SOWMYA REDDY	**(18VE1A0560)**

(2018-2022)

Under the Guidance

of

MRS. SRILATHA PULI

ASSISTANT PROFESSOR

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Affiliated to JNTUH, Approved by A.I.C.T.E and Accredited by NAAC, New Delhi)

Bandlaguda, Beside InduAranya, Nagole,

Hyderabad-500068, Ranga Reddy Dist.

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Report on **"A Machine learning model for Air Quality Prediction for Smart Cities "** submitted by **Pamulaprthi sindhu, Thirupathi Nithin, Vemuganti Meghana,Yella Sowmya Reddy** bearing Hall ticket Nos. **18VE1A0543, 18VE1A0551, 18VE1A0554, 18VE1A0560** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** from Jawaharlal Nehru Technological University, Kukatpally, Hyderabad for the academic year 2021-2022 is a record of bonafide work carried out by them under our guidance and Supervision.

Project Coordinator

Dr. V. BIKSHAM

Professor

Head of the Department

Dr. SHAIK ABDUL NABI

Professor

Internal Guide

Mrs. SRILATHA PULI

Assistant Professor

External Examiner

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **Pamulaprthi sindhu, Thirupathi Nithin, Vemuganti Meghana, Yella Sowmya Reddy** bearing **Roll Nos 18VE1A0543, 18VE1A0551, 18VE1A0554, 18VE1A0560** hereby declare that the Major Project titled **"A MACHINE LEARNING MODEL FOR AIR QUALITY PREDICTION FOR SMART CITIES"** done by us under the guidance of **Mrs. SRILATHA PULI, Assistant Professor** which is submitted in the partial fulfillment of the requirement for the award of the B. Tech degree in **Computer Science and Engineering** at **Sreyas Institute of Engineering and Technology** for Jawaharlal Nehru Technological University, Hyderabad is our original work.

Pamulaprthi Sindhu **(18VE1A0543)**

Thirupathi Nithin **(18VE1A0551)**

Vemuganti Meghana

(18VE1A0554)

Yella Sowmya Reddy

(18VE1A0560)

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without mention of the people who made it possible through their guidance and encouragement crowns all the efforts with success.

We take this opportunity to acknowledge with thanks and deep sense of gratitude to **Mrs. Srilatha Puli, Assistant Professor, Department of Computer Science and Engineering** for her constant encouragement and valuable guidance during the Project work.

A Special vote of Thanks to **Dr. Shaik Abdul Nabi, Head of the Department** and **Dr. V. Biksham, Project Co-Ordinator** who has been a source of Continuous motivation and support. They had taken time and effort to guide and correct us all through the span of this work.

We owe very much to the **Department Faculty, Principal** and the **Management** who made our term at Sreyas a stepping stone for our career. We treasure every moment we had spent in the college.

Last but not the least, our heartiest gratitude to our parents and friends for their continuous encouragement and blessings. Without their support this work would not have been possible.

Pamulaprthi Sindhu (18VE1A0543)
Thirupathi Nithin (18VE1A0551)
Vemuganti Meghana (18VE1A0554)
Yella Sowmya Reddy (18VE1A0560)

ABSTRACT

Air quality of a certain region can be used as one of the major factor determining pollution index also how well the city's industries and population is managed. Urban air quality monitoring has been a constant challenge with the advent of industrialization. Air pollution has remained a major challenge for the public and the government all over the world. Air pollution causes noticeable damage to the environment as well as to human health resulting into acid rain, global warming, heart diseases and skin cancer to the people. This project addresses the challenge of predicting the Air Quality Index (AQI), with the aim to minimize the pollution before it gets adverse, using Neural Networks, Support Vector Machines, Linear regression, K-Nearest Neighbor, Decision Tree, Random Forest. The air pollution databases were extracted from the Central Pollution Control Board (CPCB), Ministry of Environment, Forest and Climate change, Government of India. The proposed Machine Learning (ML) model is promising in prediction context for the Delhi AQI. The results show improvement of the prediction accuracy and suggest that the model can be used in other smart cities as well. Air quality has been an enormous health concern in recent decades as the place has become further industrialized and more and more of its citizens have begun driving automobiles. The occurrence of air pollution takes place in the following ways.

1. Release and generation of pollutants from their source.

2. Carry of pollutants in the atmosphere.

3. Penetrating and negatively impacting human health and ecosystems.

We tend to minimise the effects of these emissions as there is no practical, economical or technical method for zero emissions. PM 2.5 is especially dangerous because it can pass through the human body's natural filters and enter the lungs. Health concerns related to PM 2.5 include heart and lung disease, asthma, bronchitis, and other respiratory problems. Machine learning, as one of the most accepted techniques, is capable to efficiently train a model using regression models to predict the hourly air pollution concentration. Following six regressors chosen for this problem were Linear Regression, K-Nearnest Neighbor, Stochastic Gradient Descent, Decision Tree, Random Forest and Multi-layer Perception.

KEYWORDS: Pollution index, Air Quality Index (AQI), Air Pollution, Global Warming, Machine Learning, Neural Networks, Support Vector Machines, Central Pollution Control Board (PBCB).

CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Air pollution is dangerous for human health and should be decrease fast in urban and rural areas so it is necessary to predict the quality of air accurately. There are many types of pollution like water pollution, air pollution, soil pollution etc but most important among these is air pollution which should be controlled immediately as humans inhale oxygen through air. There are various causes of air pollution. Outdoor air pollution caused by industries, factories, vehicles and Indoor air pollution is caused if air inside the house is contaminated by smokes, chemicals, smell. Two types of Pollutants that is causing air pollution are Primary Pollutants and Secondary Pollutants. Primary Pollutants include: - Carbon dioxide (CO2): Carbon dioxide is playing an important role in causing air pollution. It is also named as Greenhouse gas. Global warming a major concern caused by increase in carbon dioxide in air.CO2 is exhale by Human.CO2 is also released by burning of fossil fuels. Sulphur oxide (SOX): Sulphur dioxide (SO2) released by burning coal and petroleum. It is released by various industries. When react with Catalyst (NO2), results in H2SO4 causing acid rains that forms the major cause of Air Pollution. Nitrogen oxide (NOX): Most commonly Nitrogen dioxide (NO2) that is caused by thunderstorm, rise in temperature. Carbon monoxide (CO): -Carbon monoxide is caused by burning of coal and wood. It is released by Vehicles.It is odorless, colorless, toxic gas. It forms a smog in air and thus a primary pollutant in air pollution. Toxic metals –Example are Lead and Mercury Chlorofluorocarbons (CFC): - Chlorofluorocarbons released by air conditioners, refrigerators which react with other gases and damage the Ozone Layer.

Therefore, Ultraviolet Rays reach the earth surface and thus cause harms to human beings. Garbage, Sewage and industrial Process also causes Air Pollution. Particles originating from dust storms, forest, volcanoes in the form of solid or liquid causing air pollution. Secondary Pollutants include: - Ground Level Ozone: It is just above the earth surface and forms when Hydrocarbon react with Nitrogen Oxide in the sunlight presence. Acid Rain: When Sulphur dioxide react with nitrgendioxide, oxygen and water in air thus causing acid rain and fall on ground in dry or wet form. The difference between Primary Pollutants and Secondary Pollutants

is Primary Pollutants are those which are released into air directly from Source whereas Secondary Pollutants are those which are formed by reacting with either primary pollutants or

with other atmospheric component. There are various pollutants causing air pollution but PM 2.5 being the major air pollutant as proposed by the author (J. Angelin Jebamalar & A. Sasi Kumar,2019) and comes out with the best results in predicting level of PM 2.5 in their reaSearch. Logistic regression and autoregression help in determining the level of PM2.5. The day wise prediction of pollutant level was removed by various authors further by predicting hourly wise data using different algorithm. Benzene concentration can also account into air pollution and its concentration can be determined with CO. These are the causes of air pollution. Air pollution is causing harmful effects on human beings and plants. It causes the less threatening diseases like irritation in throat, nose. Headache to most severe disease like Respiratory Problems, shortness of breath, Lungs Cancer, brain disease, kidney disease and even leads to death. There are masks which protect us from increasing air pollution and various acts are there to control air pollution. It is also necessary to create awareness among human being about air pollution. It is necessary to predict the air quality accurately. Various traditional methods are there to measure it but results are not accurate and it involves a lot of mathematical calculations. Machine learning a subset of Artificial Intelligence has an important role in predicting air quality. Various researches are being done on measuring Air quality Index by using Machine Learning algorithms. So, to control Air Pollution first necessary step is to measure accurately the Air Index Quality. Machine Learning algorithms play an important role in measuring air quality index accurately. In this paper various algorithm are compared on the basis of different condition in different areas and Neural Network comes out with best results.

Machine learning is to predict the future from past data. Computer studying (ML) is a style of artificial intelligence (AI) that delivers computers the capability to gain knowledge of without being explicitly programmed. Machine finding out makes a speciality of the progress of pc applications that can alternate when exposed to new information and the basics of laptop studying, implementation of a easy laptop finding out algorithm utilising python. Process of coaching and prediction involves use of specialised algorithms. It feed the training data to an algorithm, and the algorithm uses this training knowledge to offer predictions on a brand new

test information. Machine finding out can be roughly separated in to three classes. There are supervised learning, unsupervised finding out and reinforcement finding out. Supervised studying software is each given the input knowledge and the corresponding labeling to be trained data must be labeled with the aid of a person previously. Unsupervised learning isn't any label.

It provided to the learning algorithm. This algorithm has to figure out the clustering of the input knowledge. Subsequently, Reinforcement learning dynamically interacts with its environment and it receives positive or bad suggestions to toughen its efficiency. Data scientists use many one of a kind types of computing device learning algorithms to observe patterns in python that lead to actionable insights. At a high stage, these specific algorithms can also be labeled into two companies situated on the way they "gain knowledge of" about data to make predictions: supervised and unsupervised learning. Classification is the method of guessing the class of given information points. Lessons are in many instances referred to as goals/ labels or classes. Classification predictive modeling is the task of approximating a mapping function from enters variables(X) to discrete output variables(y). In computer studying and facts, classification is a supervised studying technique in which the pc software learns from the information input given to it after which makes use of this studying to classify new statement. This data set could without problems be bi-classification (like deciding upon whether the man or woman is male or female or that the mail is unsolicited mail or non-spam) or it may be multiclassification too. Some examples of classification problems are: speech consciousness, handwriting awareness, bio metric identification, file classification and so forth.

1.1 MOTIVATION

The necessity of healthy air has always been of great importance. As air is vital for all living beings on earth, it is our responsibility to keep the air clean. The rapid urbanization and industrialization have led the world into a new era of air pollution and is seen as a modern-day curse. Air pollution refers to the contamination of the air by excessive quantities of harmful substances. Most air pollution occurs from energy use and production, where emissions from traffic and industry are major contributors.

Air quality has increasingly attracted attention from environment managers and citizens all over the world. New tools continue to emerge to raise air quality awareness worldwide. Continuously improvements in air quality mapping are happening along with the advancements of smart cities and the amount of internet-of-things sensor devices. The increase in data produced contributes to further momentum in air pollution activity. A hot research topic is air pollution forecasting, the prediction of the atmospheric composition of pollutants for a given time and location. With an accurate air quality forecast, one can decide how to act due to air pollution health effects.

.

1.2 OBJECTIVE

The main objectives of this study were

(i) (PM 2.5) and meteorological conditions (air temperature, wind speed, relative humidity) as the evaluator

(ii) to compare the predictive and generalization abilities of this mode using a limited data set Subsequently, linear and nonlinear (MLPN) developed and performance criteria parameters

1.3 SCOPE

The goal is to use state-of-the-art research results with a special focus on machine learning methods about air quality prediction, to evaluate and apply ideas and algorithms from these studies in the context of time series prediction. The ultimate goal of air quality prediction is to enable national and global decision-makers, communities, and individuals to proactively take measures to reduce the health hazards caused by air pollution.

1.4 OUTLINE

Air quality prediction for smart cities goal is to detect the quality of the air and the level of pollutants in the air.

CHAPTER 2

LITERATURE SURVEY

Air quality prediction is a hot research field, and much progress has been made to improve performance. This chapter presents the related work of air quality prediction with machine learning. Section 2. Existing system and section 2.2 shows Proposed system.
Lot of research had been done in this field. The research taken by various authors as follows: -

[1] Kostandina Veljanovska1 & Angel Dimoski (2018) in their research had compared unsupervised neural network algorithm in which output is not known with supervised algorithm K-Nearest Neighbour, Support vector machine, Decision Trees. Neural network performs better as compared to these algorithms but lacks in determining the hourly prediction of air pollutants level.

[2] Zhao et al. ,(2018) in their research had predicted the level of air pollutant with Recurrent neural network at any given time and remove the drawback of hourly prediction due to memorization power of algorithm but lacks in working without memory operation.

[3] Savita Vivek Mohurle, Dr. Richa Purohit and Manisha Patil(2018) had predicted Pm2 and Pm10 level using Fuzzy logic. Fuzzy logic helps removing the outliers that is unwanted gas present in atmosphere but in fuzzy logic clusters are created which can contain repeated data and can lead to inaccurate prediction.

[4] CR et al.(2018) in their research had used Autoregression to detect whether the air is polluted or not and linear regression can be used to determine the level of PM2.5 but the drawback is it does not able to determine the level of PM2.5 after when there is some change in atmospheric condition and it takes into account meteorological condition such as wind speed, temperature.

[5] Zhang et al. (2018) proposed Wavelet neural network considered as robust method to determine the level of Air pollutants but lack in determining the appropriate wavelet function and

no of proper hidden layers required in their research which leads to inaccurate prediction of air pollutants.

[6] Timothy M. Amado and Jennifer C. Dela Cruz(2018) in their study considered neural network with integrated sensor giving accurate level of air pollutants but slow because it take all training set and cannot work with incomplete data set.

[7] Arwa et al. (2018) in their research had determined Benzene concentration by using Artificial Neural Network(ANN) and Support Vector Machine but does not predict the error accurately between actual and predicted value and its association with CO can be determined using method of Correlation.

[8]In this paper the author Kang et al.(2018), had compared the various models like ANN, Random Forest, Decision Trees, Least squares support vector machine model, Deep belief network and Deep belief network comes out to be superior as it takes into account hourly prediction of data but drawbacks are there are lots of issues in sensor quality of data due to faults in device .

[9] Mejía et al. (2018),had determined PM10 level best with Random Forest but does not accurately predict the level of dangerous pollutant but can work with incomplete data set. .

[10] Lidia et al.(2015) had determined Airvlc Model not only to predict the air quality level CO,NO,PM2.5 but also warns the public about dangerous air pollutant outside using his model through sensors.

[11]Recurrent Neural Networks (RNN) has shown its effectiveness in dealing with temporal data. But, data from future which may come up later that the present time is needed for prediction. RNNs can partly acquire this via postponing the output with the aid of a specific amount of time frames to incorporate future understanding. Hypothetically, a huge lengthen could be applied but in observe, it's located that prediction outcomes drip if the extend is too big. At the same time deferring the yield by way of some borders has been applied efficiently to make stronger outcome for consecutive information, the surest extend is duty elegant and have to be received by way of the experimental and blunder process. Likewise, two distinct networks, one for each and every track would be informed on all input expertise and then the outcomes can

be combined by applying arithmetic or geometric mean for ultimate forecast. Nevertheless, it's complex to receive most excellent combining considering the fact that exceptional networks trained on the identical knowledge can now not be viewed as unbiased. To overcome these obstacles, it proposed bidirectional recurrent neural community (BRNN) that may be proficient utilizing all on hand input know-how previously and future of a special time frame. Pollutant knowledge like every other sensor data shouldn't be permitted from lacking normal and anomalous values. The indiscretions may just occur due to influential mistake or any other outside reasons similar to energy-blocking or compensation of connectivity and so on. There have been occasions the place pollutant data used to be no longer suggested by a source observing post. These lacking values were incorporated making use of systematic data values of previous occasions. If a value lies out of scope, the allowable variety for an attribute is handled as an anomalous worth. Irregular values can be changed with the aid of rolling natural of prior three instances. It offered a robust manner of predicting the severity of pollution through the readings received from different sensors in 6, 12 and 24 hour upfront making use of Deep learning items. This claim is verified over the data from pollution Department from New Delhi, India via forecasting the concentration of PM2.5 pollutant. We gift our investigations with assessment to reference line method over distinct posts and over exceptional time phases. Additionally, we have now awarded a Collaborative process which achieves good in lots of the circumstances, and also attests to be extra effective.

[12] It's apparent that folks that labor in a company or concern are possible to be exposed to the hazard of breathing damaging chemical compounds and air because of their constant acquaintance to pollutants. Air pollution provides to the hazardous situation that creates negative influence on dwelling items. It is likely an actual consideration for the whole earth. Contamination of the air is a global trouble even for multi-national companies, governments, and the mass media. Making use of ordinary possessions at a better fee than the character's capability to rebuild it can cause pollution of vegetation, air, and water. As opposed to the works done by people, there are other causes that result in the release of harmful toxins. Apart from quests made by men, natural calamities reminiscent of many kinds may have an outcome in infecting the air. Technology has advanced in almost all areas and movements of living organisms. In the current world everything is completed making use of new science with a view to satisfy the demand of person, institution, manufacturer and so on. Internet of matters (IoT) is without doubt one of the

predominant exchanging information traits within the last ten years. By means of this thought, it's feasible to attach numerous embedded objects that consume less power.

[13] Outside air nice performs an essential role in human well being. Air pollution reasons colossal raises in scientific costs, disease and is assessed to intent nearly 800,000 yearly hastydemises global (Cohen et al., 2005). The outside air commonly includes physically expressively phases of numerous pollution together with particulates (PM10 or PM2.5), ozone, carbon monoxide, oxides of nitrogen and sulfur, bioaerosols, metals, volatile organics and pesticides. A gigantic proportion of those pollution are formed by way of anthropogenic events. Though most persons devote nearly every hour inside their houses, outside air exceptional may disturb the air inside to a tremendous measure. Furthermore many sufferers corresponding to asthmatics, patients with allergies and chemical sensitivities, COPD patients, heart and stroke sufferers, diabetics, pregnant women, the aged and children are mainly vulnerable.

[14] It is extensively believed that city air pollution has a right away have an impact on human wellbeing mainly in constructing and industrial nations, where air pleasant measures usually are not available or minimally applied or enforced. Contemporary reports have shown vast evidences that publicity to atmospheric pollutants has robust hyperlinks to hostile diseases together with asthma and lung irritation. The modules are in charge for getting and loading the info, preprocessing and translating the info into knowledge, estimating the pollution centered on ancient know-how, and subsequently offering the bought understanding by means of distinctive channels, corresponding to cellular application and net gateway. The focal point of the research work is on the observing and estimating.

Different results are obtained on the basis of different pollutants. Methods are chosen on the basis of different pollutants and different areas whether urban or rural and accuracy and error is predicted and seen how much predicted value is closer to exact value Lots of research had been done and different algorithms comes out to be superior with different conditions like pollutant chosen and area selection. Various algorithms had taken meteorological data like temperature, wind speed, humidity in predicting accurately the upcoming pollutant level. Neural Network and boosting model comes out to be superior than other algorithms.

2.1 EXISTING SYSTEM

In first approach monitoring of real-time Air Quality Monitoring and another is developing statistical models using historical data. This paper summarizes Air Quality Prediction studies in two categories. The first study is on prediction of PM2.5 / PM10 concentration and on prediction of air pollutants like CO2, O3, NO2 and then inferring Air Quality Index (AQI) using machine learning techniques whereas second study is on monitoring real-time AQI using sensor devices.

Urban air pollutant attention forecast is coping with a surge of large ecological monitoring data and intricate alterations in air pollution. This necessitates effective estimating methods to strengthen prediction accuracy and avoid grave contamination episodes, thereby improving ecological administration resolution-making capacity. A brand new contaminant concentration estimation process is established on sizeable amounts of ecological knowledge and deep learning approaches. This integrates colossal data using two forms of deep networks. This system is situated on a design that uses a Convolutional Neural community as the bottom layer, routinely extracting features of enter information. An extended quick term reminiscence network is used for the output layer to keep in mind the time dependence of pollution. It consists of these two deep networks. With performance optimization, the model can predict future particulate topic (PM2:5) concentrations as time series. Sooner or later, the estimation outcome are related with the outcome of numerical models. The applicability and benefits of the mannequin are also analyzed.

Urban air pollutant attention forecast is coping with a surge of large ecological monitoring data and intricate alterations in air pollution. This necessitates effective estimating methods to strengthen prediction accuracy and avoid grave contamination episodes, thereby improving ecological administration resolution-making capacity. A brand new contaminant concentration estimation process is established on sizeable amounts of ecological knowledge and deep learning approaches. This integrates colossal data using two forms of deep networks. This system is situated on a design that uses a Convolutional Neural community as the bottom layer, routinely extracting features of enter information. An extended quick term reminiscence network is used

for the output layer to keep in mind the time dependence of pollution. It consists of these two deep networks. With performance optimization, the model can predict future particulate topic (PM2:5) concentrations as time series. Sooner or later, the estimation outcome are related with the outcome of numerical models. The applicability and benefits of the mannequin are also analyzed. Air Quality Prediction based on Supervised Machine Learning Methods K. Mahesh Babu, J. Rene Beulah Air Quality Prediction based on Supervised Machine Learning Methods 207 Published By: Blue Eyes Intelligence Engineering & Sciences Publication Retrieval Number: I11320789S419/19©BEIESP DOI:10.35940/ijitee.I1132.0789S419 Experimental outcome show that it improves prediction efficiency in comparison with basic models. Air pollution has attracted huge concentration involving the everyday lifetime of men and women. It has terrible influence on human health and everyday lifestyles throughout episodes of severe air pollution with the broaden of reasons and kinds of air pollutants, the complication of pollutant attention prediction has elevated. As a result, it's imperative to make use of ecological analyzing data to more appropriately guess city air pollution levels. Conventional prediction methods, such as numerical evaluation and machine learning, are commonly used in this kind of prediction. Nevertheless, a few drawbacks of those methods were recently recognized as given below. First, numerical prediction ways are situated on knowledge as abridged with the aid of historic information or the nature of pollutant exchange.

2.2 PROPOSED SYSTEM

Supervised learning repressors Time series forecasting can also be cast as a supervised learning regression problem where the target variable ’y’ is the feature pm25 and the predictive variables ’x and y’ are ’y’ at various lags. A significant benefit to using this approach is the ability to incorporate multivariate time series into the predictive variable. For this project we use pm25 at hourly lags up to one week, and deep at the same lags as our predictive variables. We have chosen the following six repressors for this problem.

A couple of datasets from exclusive sources can be combined to type a generalized dataset, and then one of a kind computing device finding out algorithms can be utilized to extract patterns and to receive outcome with maximum accuracy.

Algorithm:

Linear Regression

The benefit of linear regression is that it is the simplest repressor and is readily interpretable. Linear Regression 20is used for finding linear relationship between target and one or more predictors. It is generally applied to one Independent variable and one dependent variable. The regression has five important assumptions:

1. Linear relationship: Linear regression assumes the relationship between the independent and dependent variables to be linear.

2. Multivariate normality: The data is normally distributed.

3.Little or No multicollinearity: Independent variables are not correlated to one another.

4. No Auto correlation: Auto correlation occurs when the residuals are not independent from each other.

5. Homoscedasticity: If the residuals are equal along the regression line then the scatter plot is a exemplary way to check homoscedasticity. .

Linear Regression is all about fitting a line into data point.

The mathematical equation of a line is y = mx + c where m is the slope of gradient and c is y intercept where the line crosses y axis.

To train a model we first check how well the model fits the training data. Its parameters are then set to fir the training set. Generally RMSE value is used to evaluate the performance of a regression model .Although practically it is easier to minimise the MSE value than RMSE value and they both lead to the same ratio.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

A software requirement specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven project, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

3.2 Software Requirements:

- Programming Language/Platform : Python
- IDE : pycharm / jupyter
- Libraries : panda, Seaborn, SkLearn ,MatPlotLib

3.3 Hardware Requirements:

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum

3.4 Functional Requirements

Functional requirement should include function performed by a specific screen outline work-flows performed by the system and other business or compliance requirement the system must meet.

Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified.

The functional specification describes what the system must do, how the system does it is described in the design specification.

If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements.

1. Load the dataset

Data Preprocessing

==========

2. Handling null values (delete record, replace with zero, and replace with column mean, mode, median)

3. Handling duplicate values (delete duplicate records)

4. Handling Categorical Values (label encoder, one hot encoding)

Feature Engineering

==========

5. Outlier detection and removal

6. Feature selection

7. Feature Scaling

8. Handling Imbalanced Dataset

9. Data Visualization

10. Train, Test Split

Working ML Models

==========

11. Create Model

12. Train Model

13. Evaluate Model

Optimization Techniques

==========

Hyper parameter Selection

Cross Validation

Model Deployment

==========

Save Model

Deploy model in Web Server

Test Model

3.4 Non-Functional Requirements

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy (.e. how precise are the systems numerical answers.).

- Portability
- Reliability
- Usability
- Time Constraints
- Error messages
- Performance
- Standards
- Interoperability
- Scalability

CHAPTER 4

SYSTEM DESIGN

4.1 IMPORTANCE OF DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. It is the process of defining software methods, functions, objects and overall structure and interaction of your code so that the resulting functionality will satisfy your users requirements. It allows you to do the best abstraction, to understand the requirements better and meet them better.This prevents redundancy and increases reusability. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us towards how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design sign aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. During, Detailed Design, the internal logic of each of the modules specification in system design is decided. During this phase, the details of the data is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented.

In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules. During the system design activities, Developers bridge the gap between the requirements specification, produced during requirements elicitation and analysis, and the system that is delivered to the user.

4.2 UML DIAGRAMS

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- Actors
- Business processes
- (logical) Components
- Activities
- Programming Language Statements
- Database Schemes
- Reusable software components

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

As a visual language, UML is used to model a software system. However, the software industry has been divided in its view on the use of UML diagrams. Even though some consider it an essential part of software systems and development, a significant number of people believe it is entirely unnecessary. This article explores the various UML diagram advantages and attempts to understand the software industry's relationship with UML design diagrams.

- **It's Flexible & Well Known**

There is no doubt that UML is an established platform for designing software. Many software developers use it as a standard notation. UML diagrams are commonly used to explain software design models. As a result, most software professionals will at least be familiar with them, if not well versed in them.

What, then, makes UML so advantageous for software development? In particular, the flexibility of UML diagrams makes them well suited for software development. You can modify the modeling elements and their interactions according to the domain or technologies you are using. This ability to transfer knowledge quickly and comprehensively is one of the most significant UML diagram advantages.

- **Effective Communication of the Software Architecture is Essential**

Software architecture is the blueprint of a system. It lays the foundation for system efficiency and process efficiency. However, this framework is only useful when communicated with everyone working on it. That's where UML comes into play.

UML is a comprehensive language that allows for modeling object-oriented software engineering and application structure, behavior, and business processes. It assists in assessing performance, security, tracking, and it provides general guidelines about the assignment in operation.

Software developers agree that architecture documentation is essential, and UML plays a vital role in architecture documentation.

Therefore, UML is an ideal visual language for communicating details about software architecture due to its broad reach. Communicating details across teams is crucial to developing a shared understanding of the material, making for a significant UML diagram advantage.

- **UML Is Easy to Understand**

Although there are 14 different types of UML diagrams, developers tend to use only three or four types of UML diagrams to document software systems. Class, sequence, and use case diagrams remain the most popular.

What does this imply? This implies that you only need to understand 20% of the UML language to model 80% of your projects. You do not need to comprehend the entire notation to use UML diagrams effectively. Knowing a little of the notation will do just fine for you, and it also means you can effectively communicate architecture systems to people that might not have an in-depth understanding of the code. This transferability is helpful and a big reason UML diagrams are beneficial to teams.

- **The Abundance of UML Tools**

There are many reasons why UML is so widely used, one of which is it's very straightforward to create a UML diagram. UML tools range from free, open-source software to expensive commercial products. Besides that, these tools go far beyond just drawing diagrams. Their coding capabilities include:

- Creating code from designs.
- Applying design patterns.
- Mining requirements.
- Reverse engineering code.
- Determining impact and complexity.

Accessing free tools that can create powerful and simple architecture diagrams is one of the reasons people use them so frequently and one of the prominent UML diagram advantages.

- **Readability and Re-usability of UML Tools**

A UML diagram is well readable because it is meant to be understood by any programmer, and it explains program relationships straightforwardly and understandably.

Traditionally, to understand a program, programmers would read the code directly. That could mean thousands of lines of code, which could increase exponentially for extensive programs.

A UML diagram helps to illustrate relationships between those lines of code more quickly. Additionally, by using UML diagrams to show the code running in a program, a programmer can identify redundant code and reuse existing portions of code rather than rewriting them. This helps increase efficiency across the board and communicates information in a much more digestible format.

- **Serves as a Visual Representation between Classes and Entities**

In a UML diagram, the relationships between classes and entities in a computer program are visualized. A class is an object that combines similar functions and variables into one place. To understand a program, you must understand how each class object stores information and relates to the other classes. It is easy to comprehend and envision a program's relationship with others by depicting this information in a diagram.

- **Helps to Plan a Program before the Programming Takes Place**

Using UML, you can plan a program before implementing it. Some tools to model UML generate code based on the classes set up in the model. This helps to reduce overhead during the implementation process. Furthermore, modifying a UML diagram is much easier than reprogramming a code section.

These UML diagram advantages, as well as the abundance of UML tools themselves, make UML the most widely used modeling and development language among software engineers.

- **Conclusion**

Creating a UML diagram is the best tool to use when communicating the meaning and content behind a codebase. Stay tuned to Fresco for more collaborative content if you liked this article.

4.2.1 Use-Case Diagrams

A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

- The purpose of use case diagram is to capture the dynamic aspect of a system. This is used to gather the requirements of a system including internal and external influences.
- The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.
- 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.

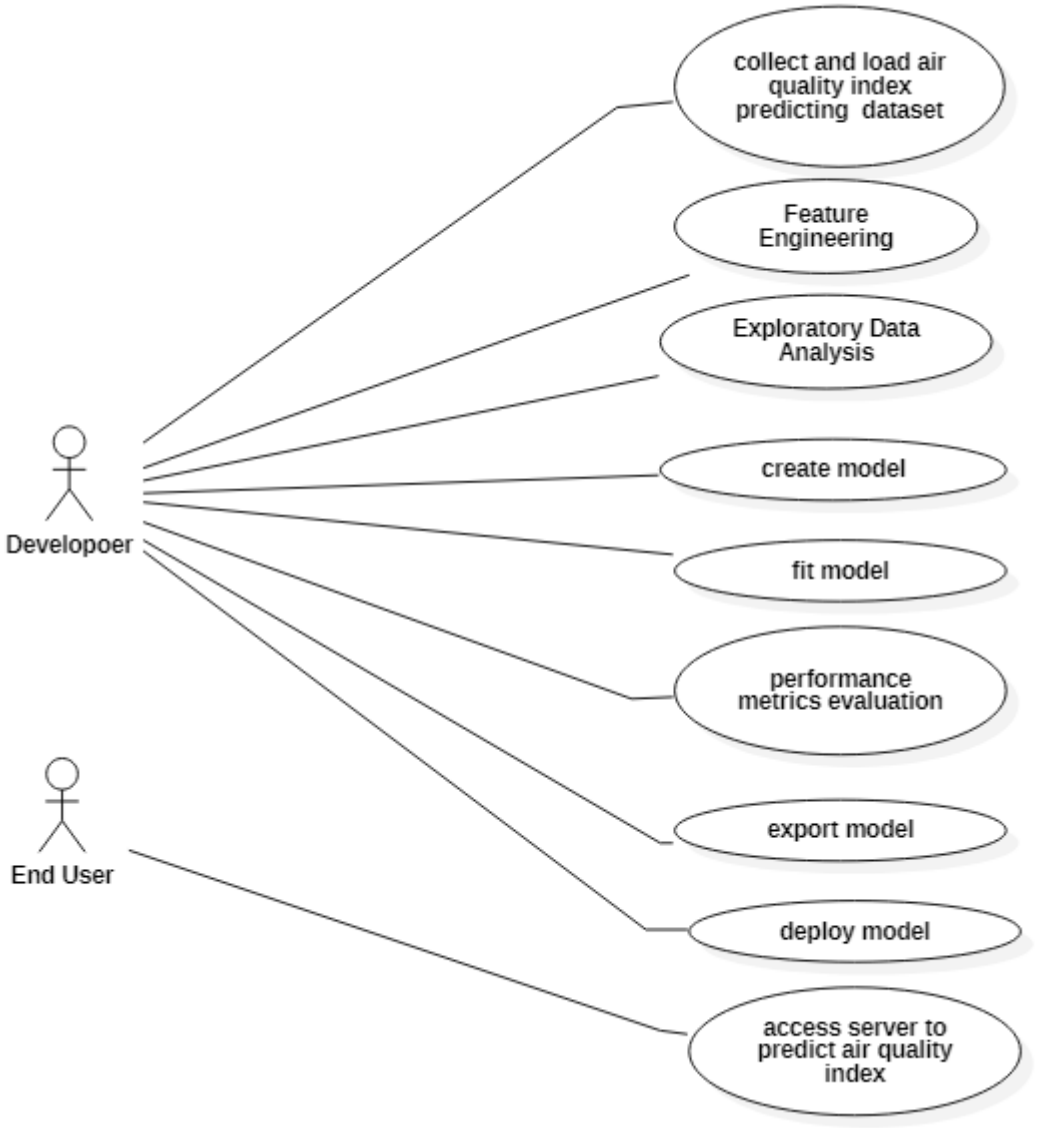


Fig.4.1. Use Case Diagram

4.2.2 Sequence diagram

- A sequence diagram details the interaction between objects in a sequential order i.e. the order in which these interactions take place.
- These diagrams sometimes known as event diagrams or event scenarios. This helps in understanding how the objects and component interacts to execute the process.

- This has two dimensions which represents time (Vertical) and different objects (Horizontal).

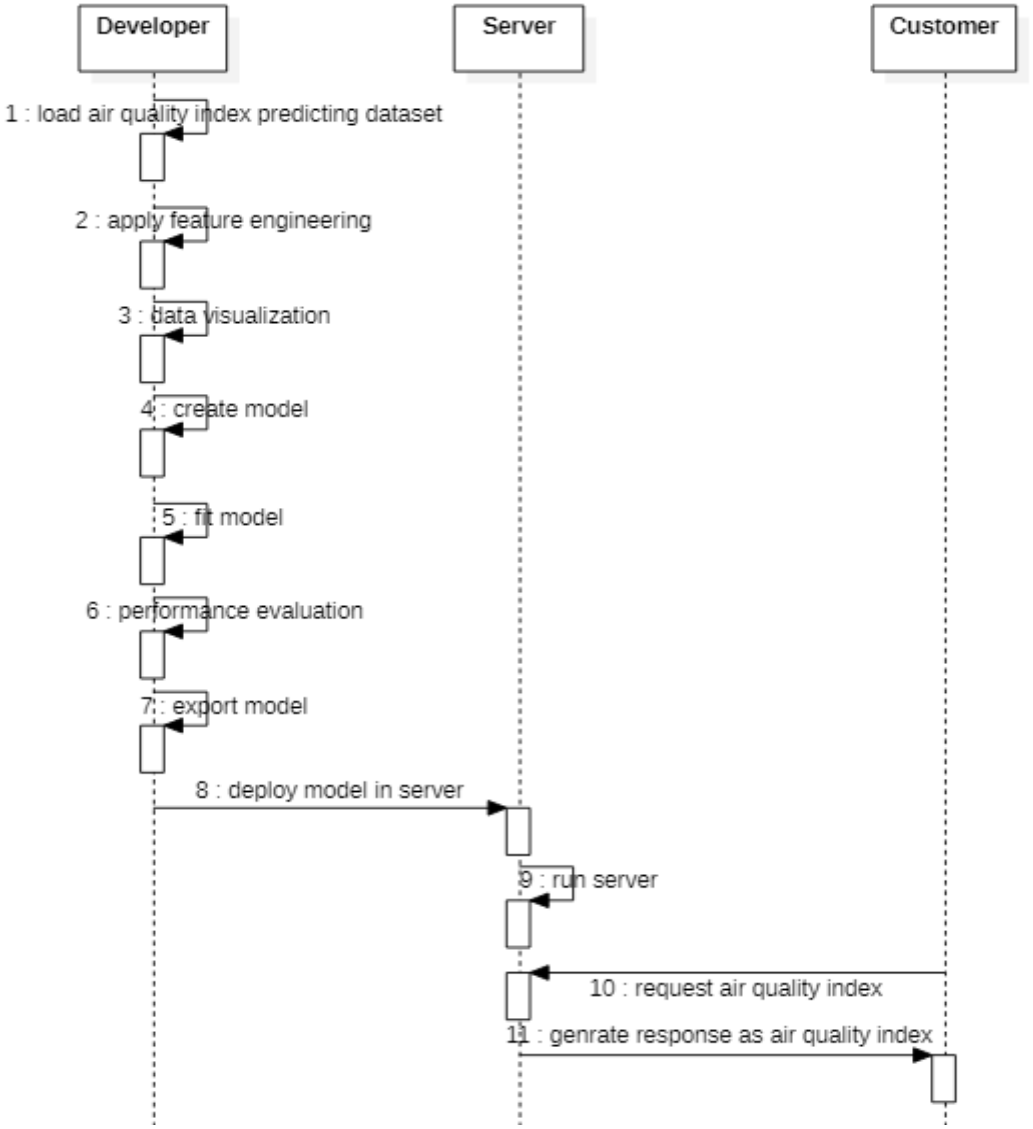


Fig.4.2. Sequence Diagram

4.2.3Activity Diagram

- It is behavioral diagram which reveals the behavior of a system. it sketches the control flow from initiation point to a finish point showing the several decision paths that exist while the activity is being executed.
- This doesn't show any message flow from one activity to another, it is sometimes treated as the flowchart. Despite they look like a flowchart, they are not.
- In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.
- We use **Activity Diagrams** to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behavior diagrams.

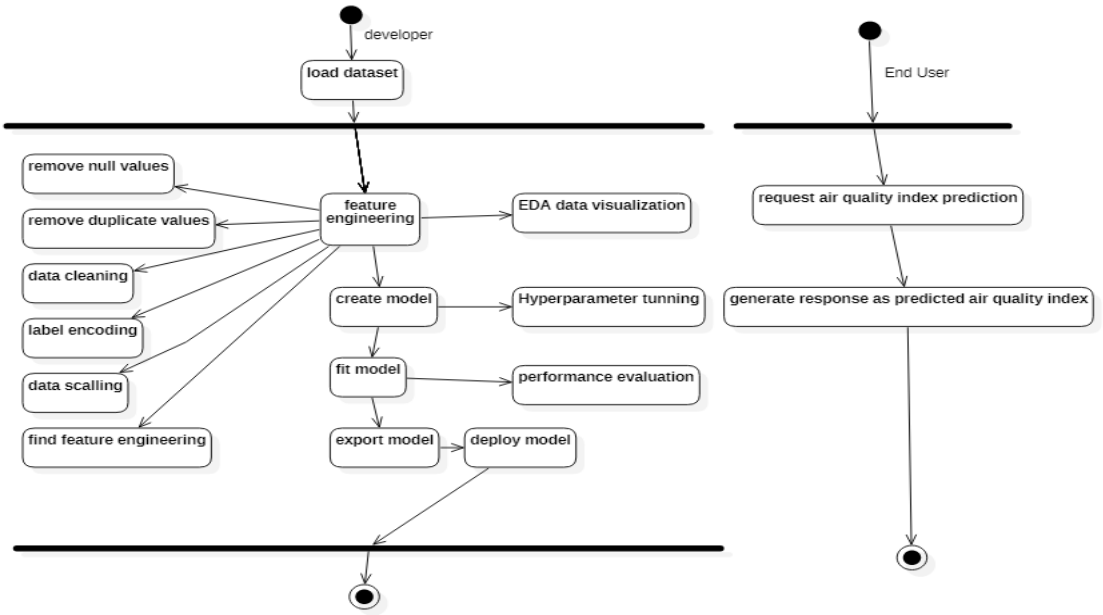


Fig.4.3. Activity Diagram

4.2.4 Class Diagram

The **class diagram** is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. **Class diagrams** can also be used for data modeling.
The class diagram describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among the classes.

It explains which class contains information and also describes responsibilities of the system. This is also known as structural diagram.

The advantages of the class diagram

Class diagrams **give you a sense of orientation**. They provide detailed insight into the structure of your systems. At the same time they offer a quick overview of the synergy happening among the different system elements as well as their properties and relationships.

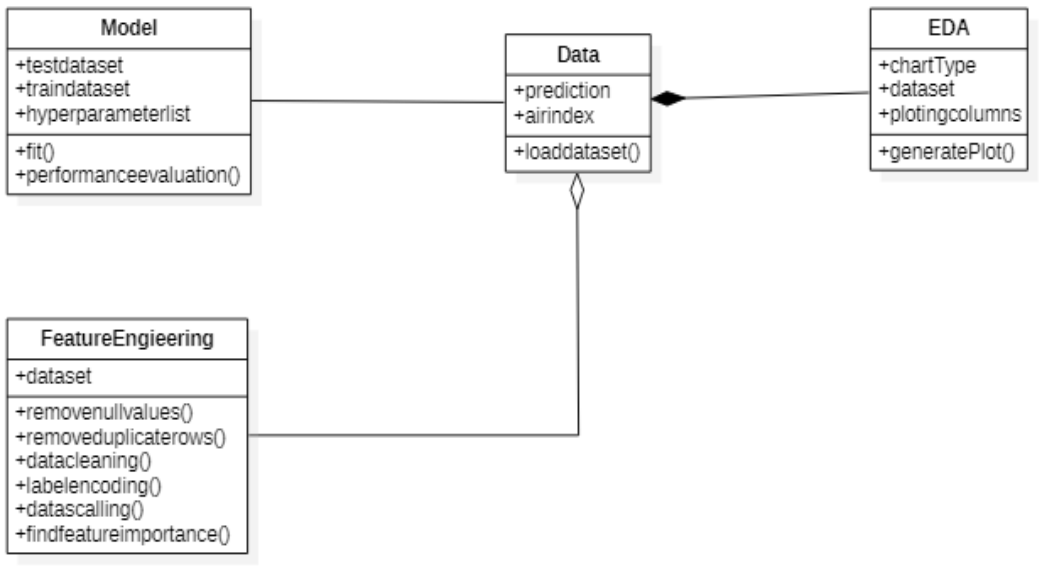


Fig.4.4. Class Diagram

4.2.5 Deployement diagram

A deployment diagram in the Unified Modelling Language models the physical deployment of artic rafts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI). The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers. Device nodes are physically computing resources with processing memory and services to execute software, such as typical computer or mobile phones. An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

Follow the simple steps below to draw a deployment diagram. You can either use the deployment diagram examples below to get a head start or use our UML diagram tool to start from the beginning.

Step 1: Identify the purpose of your deployment diagram. And to do so, you need to identify the nodes and devices within the system you'll be visualizing with the diagram.

Step 2: Figure out the relationships between the nodes and devices. Once you know how they are connected, proceed to add the communication associations to the diagram.

Step 3: Identify what other elements like components, active objects you need to add to complete the diagram.

Step 4: Add dependencies between components and objects as required.

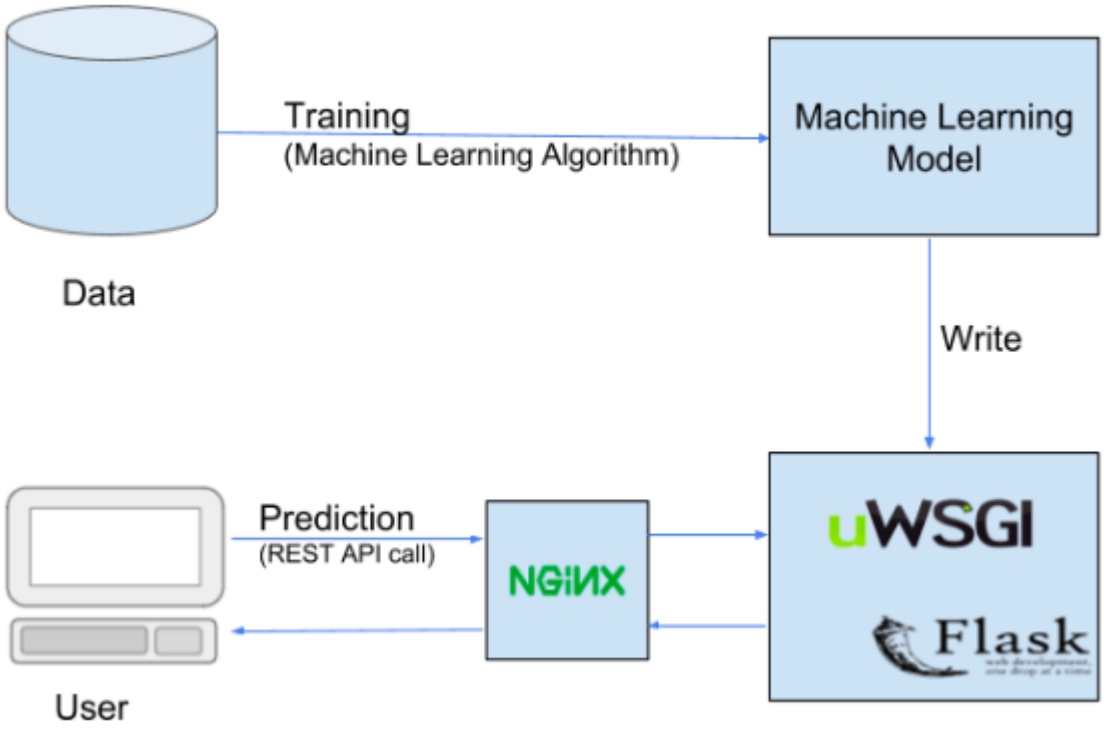


Fig 4.5. Deployement diagram

4.2.6 System Architecture

A system architecture diagram would be used to show the relationship between different components. Usually they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them. However, it can also be created for web applications.

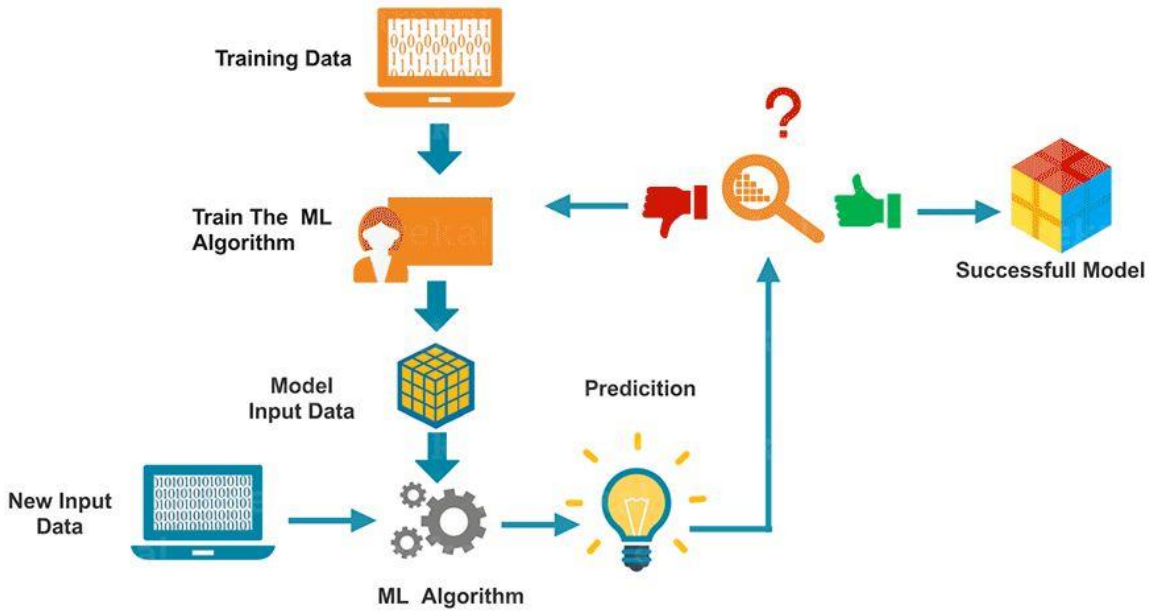


Fig.4.6. Architecture Diagram

CHAPTER 5

IMPLEMENTATION

Implementation Methodology:

The methodology in this study consists of the following procedures: data collection and preprocessing, feature selection, time windowing, and model building. All the machine learning models exploited in this study will be constructed on the open-source data mining platform, Orange, a software programmed under the python script. In this section, the details of procedures will be discussed respectively.

5.1 Data Collection

The main pollutant emissions in Taiwan are due to energy production industry, traffic, waste incineration and agriculture. In Taiwan, six pollutants (O3, PM2.5, PM10, CO, SO2, and NO2) are monitored and controlled based on their concentration time-series. Types of data used as predictors to perform analysis involve AQ: air quality data, MET: meteorological data, and TIME: the day of the month, day of the week, and the hour of the day. From 1 January 2008 to 31 December 2018, air quality data are collected from several monitoring stations across Taiwan and reported via the EPA's website. With the same timeframe, meteorological data are provided in 1-h intervals by Taiwan's Central Weather Bureau (CWB) from three air monitoring stations: Zhongli (Northern Taiwan), Chuanghua (Central Taiwan), and Fengshan (Southern Taiwan). The datasets represent different environmental conditions related to air pollutant concentration.

5.1.2 Data Pre-Processing

The number of raw data points for the Zhongli, Changhua, and Fengshan monitoring stations includes 91,672, 94,453, and 94,145, respectively. The analysis of these readings begins with a crucial phase – data preprocessing. Various preprocessing operations precede the learning phase. At any particular time, one invalid variable will not affect the whole data group, and thus it will just be either marked blank or, where available, replaced by a value sourced from the CWB, without eliminating the full row. The missing values are treated by imputation to recover the corresponding values. Given the lack of spatial proximity of the readings to the original monitoring stations, the missing values are imputed for relative humidity, temperature, and rainfall, without using wind speed or wind direction. The next imputation process used the k-NN algorithm to substitute the rest of the invalid or missing data that did not qualify for the previous imputation process. Note that the percentage of missing values is lower than 1.3% in all three-station datasets. Then, input and target data are normalized to eliminate potential biases; thus, variable significance won't be affected by their ranges or their units. All raw data values are normalized to the range of [0, 1]. Inputs with a higher scale than others will tend to dominate the measurement and are consequently given greater priority. Normalization not only improves the model learning rate, but also supports k-NN algorithm performance because the imputation is decided by the distance measure.

5.1.3 Feature Engineering

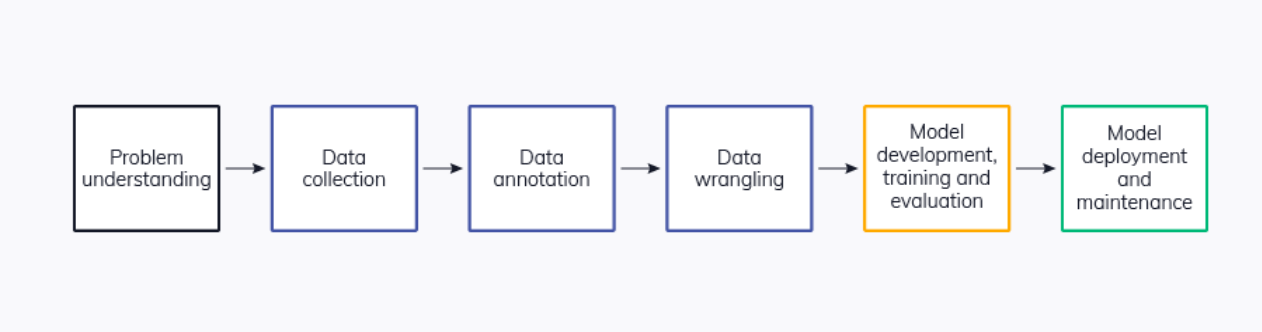
In regard to selecting features in the predictive models, the hourly AQI readings with the highest index out of 6 pollutants: O3, PM2.5, PM10, CO, SO2, and NO2 are selected. To convert the time-window-specific concentration of 6 pollutants, the AQI Taiwan Guidelines are adopted and the AQI is manually calculated, where index values of O3, PM2.5, and PM10 are needed to define AQI in Taiwan, and the lack of one or more of these values will significantly reduce the accurate assessment of current air quality

The AQI mechanism introduces several new variables to train the prediction model. For several pollutants, time windows other than hourly are more sensitive in determining AQI; therefore, the prediction interval related to the accuracy of long-term predictions is under investigation to clarify the time dependency between consecutive data points. As the AQI calculation is already established, the future value of the AQI readings in three different time intervals will be regarded as target variables

5.1.4 Performance Evaluation

According to Isakndaryan et al., the most used metrics are RMSE (root mean squared error) and MAE (mean average error), calculated based on the difference between the prediction result and the true value, while another metric, R 2 (R-squared) is essential to explain the strength of the relationship between predictive models and target variables. These three metrics provide a baseline for comparative analysis across different parameter settings for each model and across different methods. However, performance validation leads to a bias when the data set is split, trained, and tested only one time. This also means the result drawn from the testing dataset may no longer be valid after the testing subset is changed. To overcome this problem, each model is re-built 20 times using different random subsets of training and testing samples. The splitting proportion remains the same (80:20) all metrics report only a single value from the average performance of 20 identical models validated into 20 different subsets of testing instances.

Data Flow



Screenshot 5. 1 Data flow

5.2 MODULE DESCRIPTION

Implementation includes all those activities that take place to covert from old system to new system. The old system consists of manual operations, which is operated in a very difficult manner from the proposed system. A proper implementation is essential to provide a reliable system to meet the requirements of the organization.

5.2.1 TECHONLOGY DESCRIPTION

Python

The Python programming language is an Open Source, cross-platform, high level, dynamic, interpreted language.

Python is a very flexible language. It is widely used for many different purposes. Typical uses include :

- Web application programming with frameworks like Zope, Django and Turbogears
- System administration tasks via simple scripts
- Desktop applications using GUI toolkits like Tkinter or wxPython (and recently Windows Forms and IronPython)
- Creating windows applications, using the Pywin32 extension for full windows integration and possibly Py2exe to create standalone programs
- Scientific research using packages like Scipy and Matplotlib

DJANGO

Django is an open-source framework for backend web applications based on Python — one of the top web development languages. Its main goals are simplicity, flexibility, reliability, and scalability.

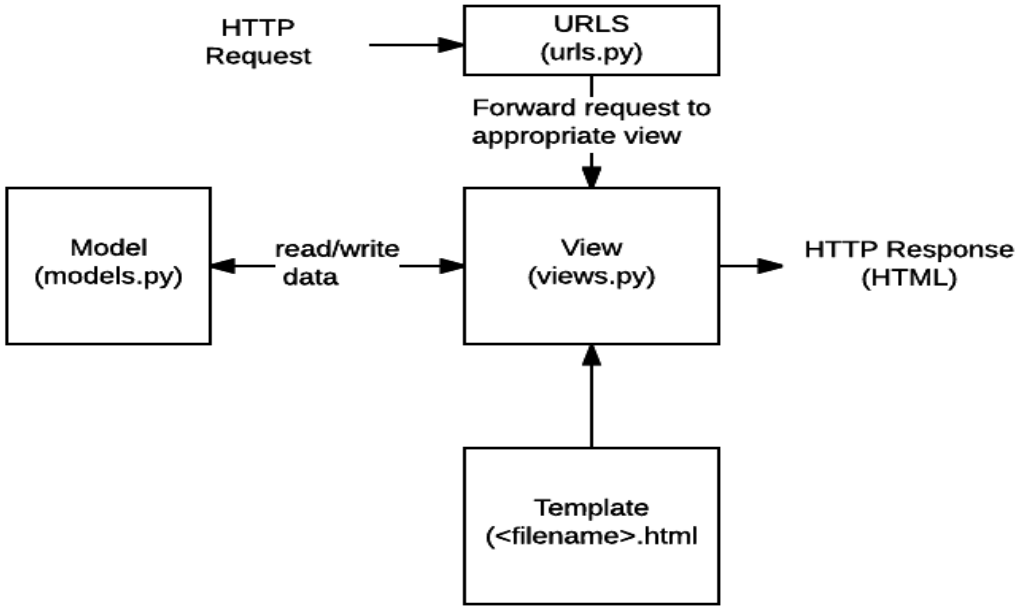
Django has its own naming system for all functions and components (e.g., HTTP responses are called “views”). It also has an admin panel, which is deemed easier to work with than in Lavarel or Yii, and other technical features, including:

- Simple syntax;
- Its own web server;
- MVC (Model-View-Controller) core architecture;
- “Batteries included” (comes with all the essentials needed to solve solving common cases);
- An ORM (Object Relational Mapper);
- HTTP libraries;
- Middleware support; and
- A Python unit test framework

What does Django code look like?

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or other client). When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template.

Django web applications typically group the code that handles each of these steps into separate files:



Screenshot 5.2 Structure of Django code

- URLs: While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in an URL, and pass these to a view function as data.

- View: A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.

- Models: Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.

- Templates: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically create an HTML page using an HTML template, populating it with data from a model. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

IDE

- PYCHARM IDE
- JetBrains has developed PyCharm as a cross-platform IDE for Python. In addition to supporting versions 2.x and 3.x of Python, PyCharm is also compatible with Windows, Linux, and macOS. At the same time, the tools and features provided by PyCharm help programmers to write a variety of software applications in Python quickly and efficiently. The developers can even customize the PyCharm UI according to their specific needs and preferences. Also, they can extend the IDE by choosing from over 50 plug-ins to meet complex project requirements.
- Overview of Important Features and Tools Provided by Py Charm
- Code Editor
- The intelligent code editor provided by PyCharm enables programmers to write high quality Python code. The editor enables programmers to read code easily through colour schemes, insert indents on new lines automatically, pick the appropriate coding style, and avail context-aware code completion suggestions. At the same time, the programmers can also use the editor to expand a code block to an expression or logical block, avail code snippets, format the code base, identify errors and misspellings, detect duplicate code, and auto-generate code. Also, the editor makes it easier for developers to analyze the code and identify the errors while writing code.
- Code Navigation
- The smart code navigation options provided by PyCharm help programmers to edit and improve code without putting extra time and effort. The IDE makes it easier for programmers to go to a class, file and symbols, along with the go to declarations invoked from a reference. The user can even find an item in the source code, code snippet, UI element, or user action almost immediately. They can further locate usage of various symbols, and set bookmarks in the code. At the same time, the developers can even take

advantage of the code navigation feature to scrutinize the code thoroughly in the lens mode.

- Refactoring
- PyCharm makes it easier for developers to implement both local and global changes quickly and efficiently. The developers can even take advantage of the refactoring options provided by the IDE while writing plain Python code and working with Python frameworks. They can avail the rename and move refactoring for files, classes, functions, methods, properties, parameters, and local/global variables. Likewise, they can improve code quality by extracting variables, fields, constants, and parameters. Also, PyCharm allows programmers to break up longer classes and methods through extract method.

Jupyter Notebook:

- The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.
- Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

5.3 EXECUTABLE CODE

```
import pandas as pd
df = pd.read_csv('dataset.csv')
df.info()
# convert 'Date' to datetime format
df['Date'] = pd.to_datetime(df['Date'])
df.head()
df.describe()
df.isnull().sum()
#df=df.dropna(how = 'any')



df = df.copy()
df['PM2.5']=df['PM2.5'].fillna((df['PM2.5'].median()))
df['PM10']=df['PM10'].fillna((df['PM10'].median()))
df['NO']=df['NO'].fillna((df['NO'].median()))
df['NO2']=df['NO2'].fillna((df['NO2'].median()))
df['NOx']=df['NOx'].fillna((df['NOx'].median()))
df['NH3']=df['NH3'].fillna((df['NH3'].median()))
df['CO']=df['CO'].fillna((df['CO'].median()))
df['SO2']=df['SO2'].fillna((df['SO2'].median()))
df['O3']=df['O3'].fillna((df['O3'].median()))
df['Benzene']=df['Benzene'].fillna((df['Benzene'].median()))
df['Toluene']=df['Toluene'].fillna((df['Toluene'].median()))
```

```
df['Xylene']=df['Xylene'].fillna((df['Xylene'].median()))
df['AQI']=df['AQI'].fillna((df['AQI'].median()))
df['AQI_Bucket']=df['AQI_Bucket'].fillna('Moderate')
df.isnull.sum()
df.duplicate().sum()
df=df.drop_duplicates(keep='first')
import seaborn as sns
import mathplotlib.pyplot as plt
city_day = df.copy()
city_day['BTX'] = city_day['Benzene']+city_day['Toluene']+city_day['Xylene']
city_day.drop(['Benzene','Toluene','Xylene'],axis=1)
city_day['Particulate_Matter'] = city_day['PM2.5']+city_day['PM10']
pollutants = ['PM2.5','PM10','NO2', 'CO', 'SO2','O3', 'BTX']
city_day.set_index('Date',inplace=True)
axes = city_day[pollutants].plot(marker='.', alpha=0.5, linestyle='None', figsize=(16, 20), subplots=True)
for ax in axes:

    ax.set_xlabel('Years')
    ax.set_ylabel('ug / m3')
def max_polluted_city(pollutant):
    x1 = city_day[[pollutant,'City']].groupby(["City"]).mean().sort_values(by=pollutant,ascending=False).reset_index()
    x1[pollutant] = round(x1[pollutant],2)
```

```
    return x1[:10].style.background_gradient(cmap='OrRd')


from IPython.display import display_html
def display_side_by_side(*args):
    html_str=''
    for df in args:
        html_str+=df.render()
    display_html(html_str.replace('table','table style="display:inline"'),raw=True)


pm2_5 = max_polluted_city('PM2.5')
pm10 = max_polluted_city('PM10')
no2 = max_polluted_city('NO2')
so2 = max_polluted_city('SO2')
co = max_polluted_city('CO')
btx = max_polluted_city('BTX')



display_side_by_side(pm2_5,pm10,no2,so2,co,btx)
import cufflinks
cufflinks.go_offline()
cufflinks.set_config_file(world_readable=True, theme='pearl')


df1 = df.copy()
```

```
df1['Pollution content'] = df1['PM2.5']+df1['PM10']+df1['NO']+df1['NO2']+df1['NOx']+df1['NH3']+df1['CO']+df1['SO2']+df1['O3']+df1['Benzene']+df1['Toluene']+df1['Xylene']


def ploting(var):
    df1[var].iplot(title=var,xTitle='Cities',yTitle=var, linecolor='black', )
    plt.show()
ploting('Pollution content')



def max_bar_plot(var):
    x1 = df1[['City',var]].groupby(["City"]).median().sort_values(by = var,
    ascending = True).tail(10).iplot(kind='bar', xTitle='Cities',yTitle=var,
                        linecolor='black', title='{2} {1} {0}'.format('')",var,' Most polluted cities('))

p = max_bar_plot('Pollution content')



def min_bar_plot(var):
    x1 = df1[['City',var]].groupby(["City"]).mean().sort_values(by = var,
    ascending = True).head(10).iplot(kind='bar', xTitle='Cities',yTitle=var, linecolor='black',title='{2} {1} {0}'.format('')",var,' Minimum polluted cities('))
p1 = min_bar_plot('Pollution content')
```

```
or = df.corr()
cor.style.background_gradient(cmap='coolwarm')
rom sklearn.preprocessing import StandardScaler, LabelEncoder
df1['AQI_Bucket']
y = df1["AQI"]
x = df1[['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2',
      'O3', 'Benzene', 'Toluene', 'Xylene']]

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 = 0)

print(X_train.shape,y_train.shape)
print(X_test.shape,y_test.shape)
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error

reg=GradientBoostingRegressor()
reg.fit(X_train, y_train)
y_pred=reg.predict(X_test)

mse = mean_squared_error(y_test,y_pred)
print(mse)
from sklearn.ensemble import AdaBoostRegressor
regr=AdaBoostRegressor()
```

```
regr.fit(X_train,y_train)
y_pred=regr.predict(X_test)

mse = mean_squared_error(y_test,y_pred)
print(mse)
from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor()
regressor.fit(x, y)
y_pred=regressor.predict(X_test)

mse = mean_squared_error(y_test,y_pred)
print(mse)
y_pred=regressor.predict([[46.89,154.0,4.0,8.97,9.65,3.05,0.57,7.86,36.5,2.34,3.84,4.72]])
print(y_pred)
```

CHAPTER 6

TESTING

6.1 IMPORTANCE

Testing is a process, which reveals error in the program. It is the major quality measure employee during software development during software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for we planned, through testing. Testing is the process of executing a program with the intent of finding an error. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself.

6.2 TYPES OF TESTING

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at different phases of software development are:

Unit testing

Unit testing is done on individual modules as a computer and become executable. It is confined only to the designer's requirements. Each module can be tested using the following strategies.

Black box testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This Testing has been uses to find errors in the following categories:

- Incorrect or missing functions

- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors

In this testing the output is checked for correctness. The logical flow of the data is not checked.

White box testing

In test cases are generated on the logic of each module by drawing flow graphs of that module and logical decision are tested on all the cases. It has been uses to generates the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false slides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to ensure their validity.

Integrating Testing

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user system meets all requirements of the client's specifications.

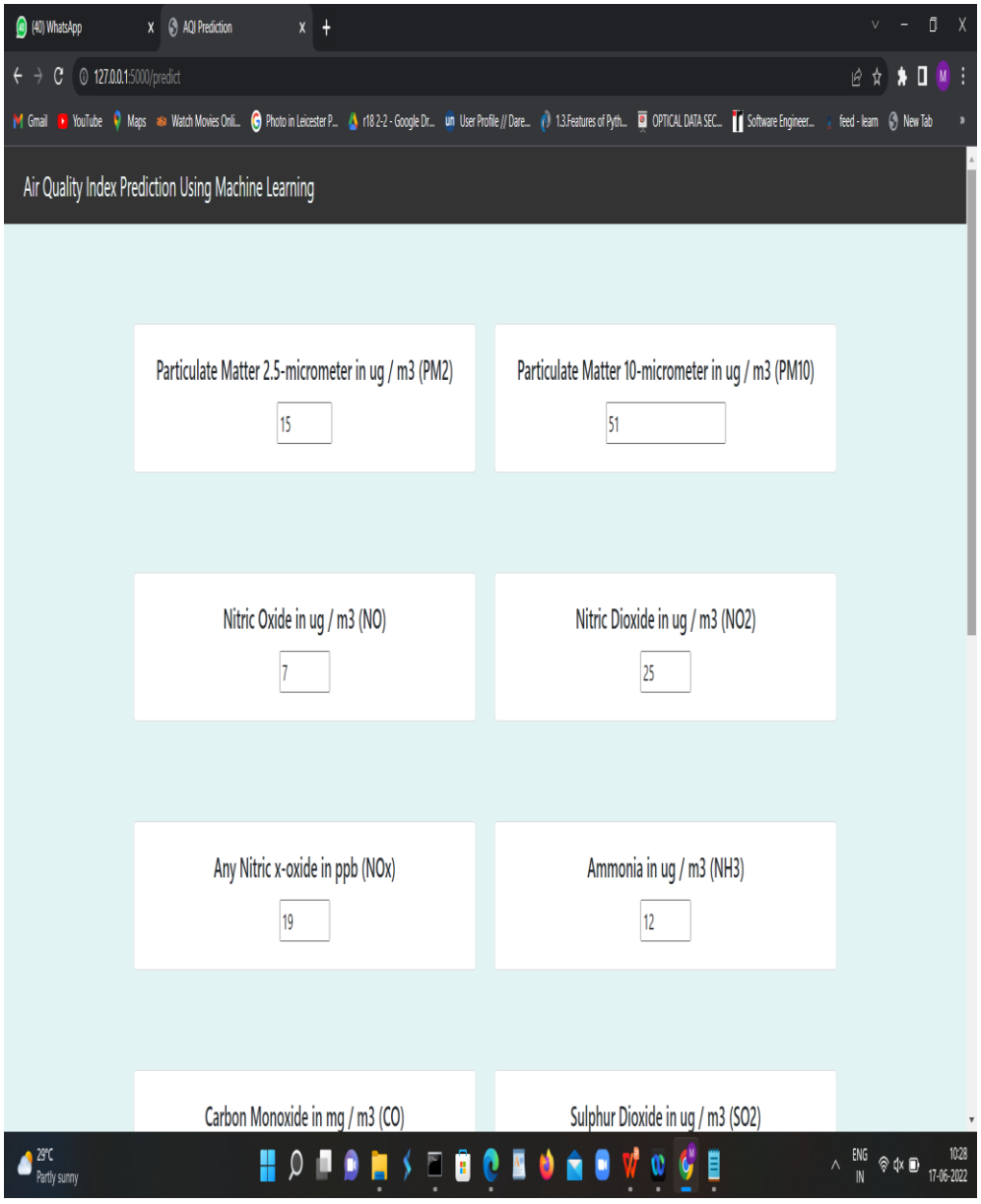
6.3 TEST CASES

Req_id	Tkt_id	Pollutants Values	Expected Output	Actual Output	Req_tckt_status
1	ID1	user gives pollutants values of a city(city with high pollution rate)	poor	poor	APPROVED
2	ID2	user gives pollutants values of a city(city with moderate pollution rate)	moderate	moderate	APPROVED
3	ID3	user gives pollutants values of a city(city with low pollution rate)	satisfactory	satisfactory	APPROVED
4	ID4	user gives pollutants values of a city(city with low pollution rate)	satisfactory	satisfactory	FAILED

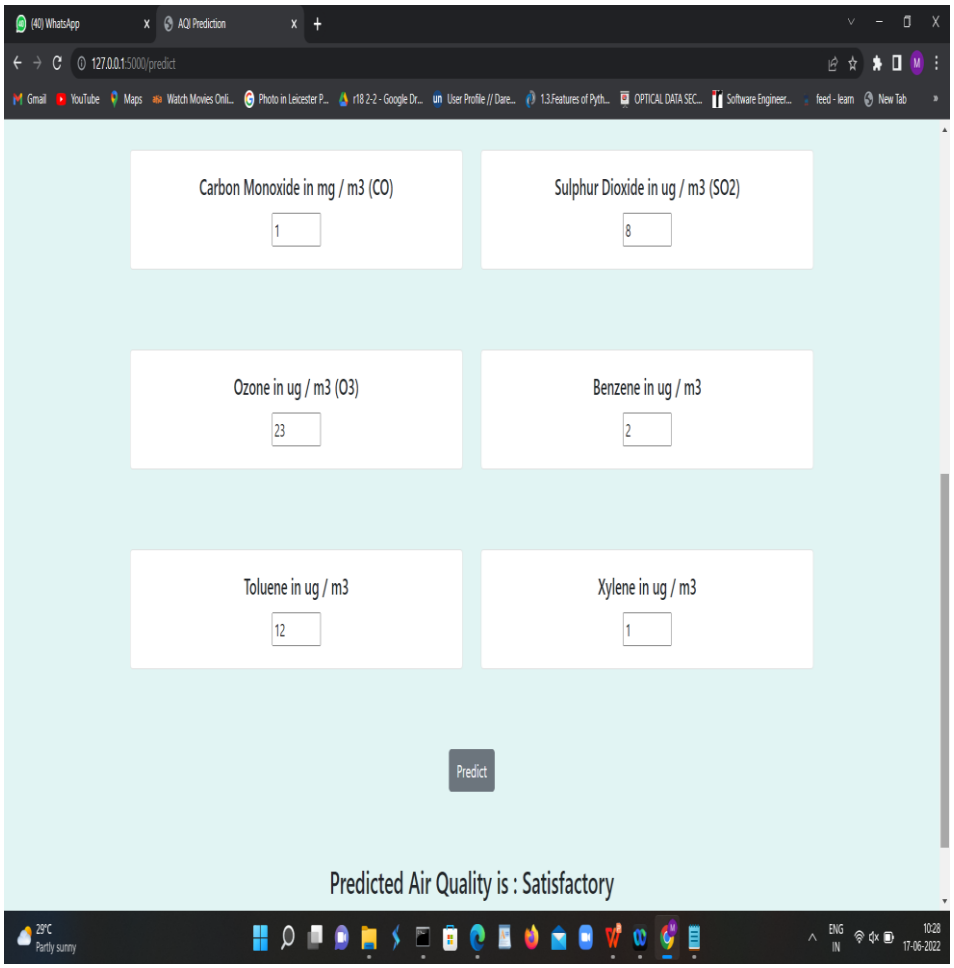
Table.6.3 Test Cases with their respective results

CHAPTER 7

RESULTS



Screenshot 7. 1 : Air quality check



Screenshot 7. 2 :Satisfactory

This section is organized into three parts. First, a general description of the dataset is provided. The datasets are mainly based on geographic distribution across Taiwan. The second part discusses the detailed development of AQI prediction models following their parameter setting and imputation. The last part evaluates the performance of the AQI forecasting models. 4.1. Data Summary In the Zhongli dataset, moderate is the most frequent AQI level in any given month (Figure 5a). Unhealthy occurs more frequently in December through April, indicating that peak pollution usually occurs in winter and spring. The year-based grouping (Figure 5b) clearly shows a general drop in pollution levels from 2014 to 2018, with a small uptick in 2016. In general, the moderate class accounts for 51% cases while good and unhealthy, respectively, account for 28% and 21%. Figure 5. Composition of AQI classes in Zhongli: (a) Month-based; (b) Year-based and

Overall-based. Similar to the Zhongli AQI pattern, pollution in Changhua peaks in March (Figure 6a). However, the degree of air pollution is more severe in Changhua, with unhealthy accounting for 59% of March readings, as opposed to 39% for Zhongli. Like Zhongli, higher AQI levels in Changhua are also clustered in winter and spring, but September, October, and November also featured significant instances of the unhealthy class (respectively 35%, 38%, and 41%). In general, Changhua has poorer air Appl. Sci. 2020, 10, 9151 9 of 17 quality than Zhongli, with more frequent AQI > 100 incidents both monthly and annually. However, the full-year AQI readings in Figure 6b show that air quality has gradually improved over time, with a 34% drop in instances of unhealthy from 2008 to 2018. Figure 6. Composition of AQI classes in Changhua: (a) Month-based; (b) Year-based and Overall-based. Southern Taiwan, especially Kaohsiung City, is notorious for its poor air quality due not only to emissions from nearby industrial parks but from particulate matter blowing in from China and Southeast Asia. Figure 7a,b shows significant instances of the unhealthy class (red bars) air quality for most of the year, with reduced pollution levels only in May to September. The worst air quality is concentrated in December and January (respectively 78% and 80% unhealthy). Figure 7. Composition of AQI classes in Fengshan: (a) Month-based; (b) Year-based and Overall-based. The winter spike in air pollution is partly due to seasonal atmospheric phenomena that trap air pollution closer to the ground for extended periods. From October to March, Fengshan air quality readings are good less than 5% of the time. In terms of year-based AQI class composition, not much improvement is seen until in 2014–2015 with a sharply declined unhealthy scores after which levels remain relatively stable. Overall, for the 11 years, the Fengshan dataset is dominated by AQI > 100 (46%) followed by 51 ≤ AQI ≤ 100 (37%), and AQI ≤ 50 (17%

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Conclusion

The objective of the current study is to establish an efficient forecasting model for AQI in Delhi.. This work can be extended by predicting a higher number of future timesteps for all the eight pollutants considered in calculating AQI.In this project, we developed an air quality prediction system for the city of Trivandrum obtaining datasets from Pollution Control Board, India. This model was developed overcoming the limitations of the existing systems, and adapted to the Indian scenario by considering the data from delhi . It can be further developed to predict the ambient air quality of multiple regions together. In future, we can introduce more meteorological factors like precipitation, minimum and maximum temperature, solar radiation, vapor pressure etc. to increase the accuracy of the system. The unclear trend and wide fluctuations of air pollutants is also attributed to the emissions from pollution sources like transportation, industrial emissions etc. Those factors need to be considered as well. The analytical procedure began from information cleaning and processing, incomplete records, detailed evaluation and in the end model constructing and evaluation. 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 scope

- 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.
- This model can be advanced by developing it to predict the ambient air quality of multiple regions simultaneously.

- In future, we can introduce more meteorological factors like precipitation, minimum and maximum temperature, solar radiation,apor pressure etc. to multiply the precision of the system.
- The ambiguous trends andimmensevariation of air pollutants is also imputed to the discharge from pollution sources liketransportation, industrial emissions etc.
- These factors also must be taken into deliberation.

CHAPTER – 9

REFERENCES

- International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 11, 2019
- Baklanov, A., Zhang, Y. (2020) Advances in air quality modeling and forecasting. Glob. Trans., 2: 261-270
- Athira, V., Geetha, P., Vinayakumar, R. (2018) DeepAirNet: Applying recurrent networks for air quality prediction. Proc. Comp. Sci., 132: 1394-1403.
- A.Gnana Soundari, Akshaya A.C, J.Gnana Jeslin. "Indian Air Quality Prediction and Analysis Using Machine Learning". International Journal of Applied Engineering Research ISSN 0973- 4562 Volume 14, Number 11, 2019.
- Venkat Rao Pasupuleti, Uhasri, Pavan Kalyan, Srikanth, Hari Kiran Reddy. "Air Quality Prediction of Data Log by Machine Learning". 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE (2020).
- E. Kalapanidas and N. Avouris, "Applying machine learning techniques in air quality prediction," in Proc. ACAI, vol. 99, September 2017.
- Machine learning with decision trees. [Online]. Available: https://blog.knoldus.com/2017/08/14/machine-learning-with-decisiontrees/