Kuwait University

Air Pollution Assessment and Computational Air Quality Modeling in North Kuwait

Submitted by:

Ahmed Ewis

A Thesis Submitted to the College of Graduate Studies in Partial Fulfillment of the Requirement for the Master's Degree in:

Chemical Engineering

Supervised by:

Supervisor: Dr. Fahad M. Al-Fadhli

Co-Supervisor: Dr. Nawaf Alhajeri

Kuwait

January/2024

© 2024

ALL RIGHTS RESERVED

Kuwait University

College of Graduate Studies

Signatory Page

(Thesis Examination Committee)

The undersigned certify that they have read, and recommend to the College of Graduate Studies for acceptance, a M.Sc. Thesis entitled Thesis Title "**Air Pollution Assessment and Computational Air Quality Modeling in North Kuwait**" submitted by Ahmed Ewis in partial fulfillment of the requirements for the Master's degree in Chemical Engineering, College of Engineering and Petroleum.

Signatures of committee members	**Date**
_____	_____
Prof. Hisham Ettouney, Professor (Convenor)	
_____	_____
Dr. Fahad Al-Fadhli, Associate Professor (Supervisor)	
_____	_____
Dr. Nawaf Alhajeri, Associate Professor (Co-Supervisor)	
_____	_____
Dr. Fadhel Azeez, Associate Professor (Member)	

Thesis/Dissertation Non-Exclusive License Publication Authorization

I, Ahmed Ewis, confirm that my thesis (dissertation) is my original work , does not infringe any rights of others, and that I hereby grant to Kuwait University a non-exclusive, worldwide, irrevocable, royalty free license, in respect of my thesis (dissertation), to reproduce, convert, publish, archive, preserve, conserve, communicate and distribute, and loan, in paper form, and electronically by telecommunication or on the internet, and /or any other formats as may be adopted for such use from time to time without breaching to the State of Kuwait Law No.25 of 2019 regarding copyright and related rights.

Signature: 

Date: 25/10/2023

© 2023
All Rights Reserved

Abstract

The issue of air quality has become a pressing matter in various regions globally, including Kuwait, where industrial activities are known to emit pollutants into the atmosphere. This study aimed to evaluate the hourly data of air quality in the urban localities of Al Jahra and Saad Al Abdullah by examining the levels of the pollutants nitrogen dioxide (NO_2) and ozone (O_3). The objective was to identify the hourly pollution levels of these pollutants and establish correlations between them. To this end, advanced statistical methods were applied using the openair package in R programming language to delve into the relationship between O_3 and NO_2. The study also highlights the number of exceedances of the pollutants against the Kuwait Environmental Public Authority (KEPA) standard. The results of the data analysis and exploration revealed a clear seasonal pattern in the concentration levels of NO_2 and O_3. The study found that the concentrations of NO_2 were consistently higher in the winter season as compared to the summer season. Conversely, the concentrations of O_3 were consistently higher in the summer season as compared to the winter season. An essential aspect of the research was modeling, which involved diagnostic analysis to examine the performance of the machine learning models used. A comparative analysis was conducted to determine the most appropriate machine learning models and deep learning models for this type of data, including boosting, bagging, and neural network techniques. The results of the study indicated a robust negative correlation between NO_2 and ozone O_3. The use of the boosting method, specifically the CatBoost algorithm, was found to exhibit the highest level of accuracy for this type of data, with a coefficient of determination (R^2) of 0.91 achieved on test dataset after hyperparameter tuning. This result provides a higher level of accuracy for the hourly prediction of the pollutants. The results were in line with the recommendations of experts who suggest using boosting techniques for tabular data and deep learning for larger datasets or different types of inputs like images, text, or audio. The findings of the study underscore the importance of monitoring and improving air quality in Kuwait and provide a basis for future research in the field. They also facilitate forecasting and allow authorities to take necessary precautions and raise awareness if needed. The thesis provides a comprehensive understanding of the relationship between NO_2 and O_3 and serves as a valuable contribution to the field of air quality assessment.

Keywords: Air pollution, Air quality, NO_2, O_3, Al Jahra, Saad Al Abdullah, Kuwait

Table of Contents

ABSTRACT ..... **V**

LIST OF TABLES ..... **IX**

LIST OF FIGURES ..... **X**

ACKNOWLEDGMENT ..... **XII**

CHAPTER 1 INTRODUCTION & LITERATURE REVIEW **1**

CHAPTER 2 METHODOLOGY **16**

2.1 Data collection **16**

Data Preprocessing **17**

2.2.1 Data Cleaning 17

2.2.2 Data Transformation 25

Data Analysis **30**

Choosing a model **41**

2.4.1 Regression Loss function 42

2.4.2 Linear Regression Predictive Model 43

2.4.2.1 Multiple linear regression 43

2.4.2.2 Assumptions of Multiple Linear Regression 44

2.4.2.3 Least square method 44

2.4.3 The K-nearest neighbors (KNN) algorithm 46

2.4.4 Ensemble Techniques 47

2.4.5 Artificial Neural Network with Deep Learning 49

2.4.6 Forecasting Time series algorithms 50

2.4.6.1 SARIMAX Model 51

2.4.6.2 Long short-term memory (LSTM) 51

2.4.6.3 NeuralProphet 52

Training a model **52**

2.5. Hyperparameter tuning.....53
2.5.1 Grid-Search-Cv.....54
2.5.2 Randomized-Search-Cv.....54

CHAPTER 3 RESULTS.....55

3.1 Hyperparameter tunning for Machine Learning & Deep learning algorithms.....55
3.1.1 KNN Algorithm Architecture.....55
3.1.2 KNN Algorithm Metrics after Hyperparameter tuning.....56
3.2.1 Random Forest Algorithm Architecture.....56
3.2.2 Random Forest Algorithm Metrics after Hyperparameter tuning.....57
3.3.1 XGBoost Algorithm Architecture.....58
3.3.2 XGBoost Forest Algorithm Metrics after Hyperparameter tuning.....58
3.4.1 CatBoost Algorithm Architecture.....59
3.4.2 CatBoost Forest Algorithm Metrics after Hyperparameter tuning.....61
3.5.1 Artificial Neural Network Architecture.....62
3.5.2 Artificial Neural Network Metrics after Hyperparameter tuning.....63
3.6 Results After Hyperparameter tunning for forecasting timeseries algorithms.....65
3.6.1 SARIMAX Architecture.....65
3.6.2 SARIMAX Metrics after AIC minimization.....67
3.6.3 NeuralProphet & Prophet.....67

3.2 Sensitivity Analysis.....68

3.3 Model Deployment.....70

CHAPTER 4 CONCLUSION AND RECOMMENDATIONS.....73

4.1 Conclusion.....73

4.2 Recommendations.....73

REFERENCES.....75

APPENDIX.....77

BIOGRAPHY.....86

الملخص 87

List of Tables

Table 2.1: Simple-Imputer Metrics..... 19
Table 2.2: Interpolation technique Metrics..... 19
Table 2.3: Iterative-Imputer Metrics.....19
Table 2.4: KNN-Imputer Metrics with eight nearest neighbors.....22
Table 2.5: PCA Components with the metrological conditions.....31
Table 2.6: Segments of K-means with PCA.....45
Table 2.7: KNN Hyperparameters after RandomizedSearchCV..... 49
Table 2.8: KNN metrics after RandomizedSearchCV..... 51
Table 2.9: Random Forest Hyperparameters after exhaustive GridSearchCV..... 51
Table 2.10: Metrics of Random forest after exhaustive GridSearchCV..... 52
Table 2.11: XGBoost Hyperparameters after RandomizedSearchCV..... 53
Table 2.12: Metrics of XGBoost after exhaustive RandomizedSearchCV..... 54
Table 2.13: CatBoost Hyperparameters after RandomizedSearchCV..... 54
Table 2.14: Metrics of CatBoost after exhaustive RandomizedSearchCV..... 56
Table 2.15: Artificial Neural Network Hyperparameters after Keras Tuner..... 57
Table 2.16: Metrics of Artificial Neural Network after Keras Tuner..... 58
Table 2.17: SARIMAX Hyperparameters after AIC minimization..... 60
Table 2.18: Metrics of SARIMAX after Model Evaluation..... 62

List of Figures

Figure 2.1:The comparison results for Evaluation Metrics.....20
Figure 2.2: PCA explained variance by components.....22
Figure 2.3: K-means with PCA Clustering.....23
Figure 2.4: Component 1 vs Component 2 clustered by PCA components.....24
Figure 2.5: Component 1 vs Component 3 clustered by PCA components.....24
Figure 2.6: Correlation Heatmap.....26
Figure 2.7: Feature engineering selection using F_regression.....28
Figure 2.8: KEPA Standards.....29
Figure 2.9:NO_2 vs O_3 trend in 2014.....31
Figure 2.10:NO_2 episodes vs O_3 values in 2014.....32
Figure 2.11: wind direction and episodes concentrations of NO2.....33
Figure 2.12: NO2 vs O3 relationship..... 33
Figure 2.13: O3 Polar plot 33
Figure 2.14: NO2 Polar plot 33
Figure 2.15: NO2 Polar frequency..... 35
Figure 2.16: O3 Polar frequency 35
Figure 2.17: percentile rose for NO2 daylight and nighttime frequency 36
Figure 2.18: percentile rose O3 daylight and nighttime frequency..... 36
Figure 2.19: Calendar plot with wind direction for NO2 37
Figure 2.20: Calendar plot with wind direction for O3 37
Figure 2.21: Calendar plot classes for NO2..... 38
Figure 2.22: Calendar plot with classes for O3..... 38
Figure 2.23: Example of KNN Algorithm concept..... 44
Figure 2.24: Bagging and Aggregation Block Diagram in Random Forest..... 45
Figure 2.25: Boosting Technique Block Diagram in XGBoost 46
Figure 2.26: Neural Network Configuration..... 47
Figure 2.27: Simple RNNs Architecture..... 49
Figure 2.28: Train Validate Test Split..... 50
Figure 2.29: Values of KNN after RandomizedSearchCV..... 52

Figure 2.30: Prediction Model for KNN on test set.....52
Figure 2.31: KNN Predictions Performance on test set.....53
Figure 2.32: Random Forest Regressor Predictions Performance on Test set 54
Figure 2.33: Prediction Model for Random Forest on test set 54
Figure 2.34: Prediction Model for XGBoost on test set 55
Figure 2.35: XGBoost Predictions Performance on Test set 55
Figure 2.36: Learning curve in XGBoost 56
Figure 2.37: CatBoost Predictions Performance on Test set 57
Figure 2.38: Prediction Model for CatBoost on test set 57
Figure 2.39: CatBoost Feature importance after fitting the data 57
Figure 2.40: Learning curve for CatBoost 58
Figure 2.41: SHAP value for CatBoost algorithm 58
Figure 2.42: Results of Hyperparameter tunning from Keras Tuner 59
Figure 2.43: Neural network learning curve with early stopping applied 59
Figure 2.44: Neural Network Predictions Performance on Test set 60
Figure 2.45: Prediction Model for Neural Network on test set 60
Figure 2.46: Neural Network Architecture is computationally expensive with high metrics 61
Figure 2.47: Neural network with 60 neurons achieves an r2 of 0.6 and an RMSE of 0.0126 ... 61
Figure 2.48: AIC minimization..... 62
Figute: 2.49: ACF and PACF plots 62
Figure 2.50: 80% split from the trainset 63
Figure 2.51: Forecasted Daily O3 on the daily test set 63
Figure 2.52: Residuals Diagnostics Plot 63
Figure 2.53: Hourly forecasted O3 on test set after 90% split 64
Figure 2.54: Metrics from Modeling Timeseries NeuralProphet and Prophet 64
Figure 2.55: Comparison of observed vs. NeuralProphet and Prophet forecasts 64
Figure 2.56: Residuals Diagnostics Plot from NeuralProphet..... 65
Figure 2.57: Comparison between machine learning algorithms using pycaret automl 66
Figure 2.58: Residuals for catboost model in pycaret automl 67
Figure 2.59: Feature importance in pycaret automl 67

Acknowledgment

I want to express my sincere gratitude to those who have been pivotal in my academic journey. First and foremost, I am immensely thankful to my thesis supervisor, Dr. Fahed Alfadhli, and my thesis co-supervisor, Dr. Nawaf Alhajeri. Their unwavering support, expert guidance, and valuable feedback have been instrumental in the completion of my thesis. Their mentorship has been invaluable, and I am deeply appreciative of his contributions. I also owe a special debt of gratitude to my father, who has not only been my parent but also my ideal role model. His dedication, integrity, and unwavering belief in me have shaped my academic pursuits and character. I would like to extend my thanks to the broader academic community and my peers for their collective influence on my growth and learning. To all who have been part of my academic journey, your support, encouragement, and belief in me have left a lasting impact. I am genuinely appreciative of your presence in my life and the role you have played in my academic achievements.

Chapter 1
Introduction & Literature Review

A significant concern for many people is air quality. Air quality is one of the top environmental concerns in many parts of the world. However, it is a fact that the air we breathe contains many different contaminants, including ozone, carbon dioxide, carbon monoxide, sulfur dioxide, particulate matter, and other chemicals that are harmful to our health. In many cases, the presence of these pollutants can be controlled through regulations imposed by the government to limit the release of toxic gases into the atmosphere. But many of the pollutants are not easily regulated, and it is, therefore, essential to develop better methods for identifying the presence of harmful pollutants in the environment. Air pollution is a health hazard and an environmental pollutant. The main components of air pollution are ground-level O_3, NO_2, CO, and particulate matter (PM). Recently, researchers have developed a new method to assess the levels of air pollutants in different environments, such as urban areas. The new method is based on the use of advanced computer models that can simulate various conditions and produce predictions based on the data that has been collected. The methods can be used for developing better strategies for dealing with the problems caused by pollution and can be used to inform decisions about when to take action to improve the level of air quality in a particular area. The study of air quality is concerned with monitoring air quality and modeling the impact of emissions on the environment. This thesis aims to analyze the environmental conditions of the urban areas using real-time data obtained from the air quality sensors deployed across the north of Kuwait particularly in Al Jahra, and Saad Al Abdullah. Air quality modeling is an essential technique for understanding the impact of pollution on humans and the environment. Air quality assessment is an integral part of maintaining good environmental standards as it can help identify sources of pollution and provide essential information for designing effective control strategies to prevent or reduce pollution levels in the environment. One of the main challenges faced by governments around the world is the amount of air pollution that is present in some regions. Every year millions of people worldwide suffer from health problems caused by inhaling toxic substances in the air, which can cause a wide range of diseases, including respiratory diseases and cancer. And in order to ensure that the air around us is safe to breathe, governments must put strict controls on emissions from industries and other sources of pollution, such as power plants and vehicle exhausts.

One method commonly used for measuring the concentrations of pollutants in the air is known as continuous monitoring. This method involves placing stationary sensors at various locations in the environment and taking measurements at regular intervals over a period of time. The data produced by the sensors is then analyzed using special computer programs to assess the air quality in the area and identify possible sources of pollution. However, this method is costly, and it can be challenging to identify the precise source of the pollution in the areas that have already been investigated. Another disadvantage of this method is that it can only be used to measure the levels of certain types of pollutants, and it can, therefore, not be used to monitor the composition of different gases in the atmosphere.

Air Quality is one of the most intriguing subjects to be studied, as it is continual research that will be developed now and then by different tools and techniques to conclude helpful information. Most industrial operations produce emissions of harmful gases and particles which harmfully impact the environment, such as NOx, SOx, CO_2, and other gases. Air quality plays a critical pioneer role in environmental challenges like global warming, Ozone depletion, and extinction of animals or plant species. Around a century ago, specialists and scientists started raising this crucial concern, and worldwide organizations that are solely dedicated to saving the environment by setting the necessitated regulations and standards have been established. The most widely known environmental organizations are the American EPA (Environmental Protection Agency), the European Network of the Head of Environmental Protection Agencies, Kuwait Environment Public Authority (KEPA). The standards and regulations set by these organizations are followed and adopted globally in all industries that involve air quality tracking methods and standards. All regulations necessitate tracking concentrations of different pollutants in the air to ensure their levels are within acceptable ranges as per standards, thereby maintaining healthy air quality. Some of the chemicals released are known or suspected cancer-causing agents responsible for developmental and reproductive problems. The ultimate goal of this project is to develop a computational model to predict the levels of air pollutants on a real-time basis using the data collected from the air quality sensors and historical data from previous years. The results obtained from the model will help manage and control air pollution in the north of Kuwait. In this research, a set of environmental parameters will be measured and analyzed to determine the factors affecting the level of air pollution on the north of Kuwait. The data obtained from the measurements will be

used to develop air quality models that can predict the pollutants' levels in real-time. These models can be used by the administration and local authorities to monitor and control the level of air pollution in the north of Kuwait.

Air Pollution is deemed one of the most vital worldwide environmental concerns, which can yield atmospheric problems, resulting in the deterioration of the ambient Air Quality, causing severe health effects, especially regarding respiratory system chronic diseases and cancers. Crucial factors are affecting the Air Quality in Kuwait. Motor vehicles, refineries, petrochemical plants, fossil fuel power plants, and other industries generate pollutant emissions in Kuwait. Moreover, meteorological conditions are substantial and can affect people and human activities. Air pollution is a major cause of concern in many developed and developing countries. According to the World Health Organization (WHO), about 7 million people die yearly from outdoor air pollution. Exposure to air pollution is associated with health problems such as respiratory diseases, heart disease, and cancer. This is one of the main reasons governments worldwide have made efforts to reduce the level of air pollution. Due to changing climatic conditions, countries have been experiencing an increase in the number of people suffering from various respiratory diseases. It is noteworthy to declare that this thesis combines primary and secondary pollutants. They differ from each other. The primary air pollutants, such as Particulate matter with aerodynamic diameters less than or equal to 2.5 μm ($PM_{2.5}$), are formed and emitted directly from particular sources like cars, vehicles, and natural gas power plants and the environment. On the other hand, secondary air pollutants like O_3, NO_2, and SO_2 are formed by chemical reactions. Still, they are harder to control because of the different forms and syntheses they appear to have as they originate naturally in the environment causing catastrophic problems like photochemical smog.

For the time being, we live in the era of technology and quantum computers, so developing predictive models of our physical worlds could facilitate many optimizing aspects of our lives. A predictive Emissions Monitoring System (PEMS) was created as a replacement for the Continuous Emissions Monitoring System (CEMS), which has many disadvantages, including high initial capital investment, high operational costs, repairs, and operator preparation. To create a model that can forecast emissions, a PEMS uses first-principles, mathematical, or Artificial Intelligence (AI) approaches to use the operating parameters of combustion facilities. A model is created using

historical paired emissions and selected process data (e.g., wind speed, wind direction, pollutants composition, solar intensity, relative humidity, temperature data, and environmental conditions) to estimate the plant's emissions and airborne concentrations. A case study of model construction for tracking air pollutants using different machine learning algorithms by sci-kit-learn, a free machine learning library for python that supports most of the ML algorithms, both supervised and unsupervised, like, linear and logistic regression, support vector machine (SVM), Naive Bayes classifier, gradient boosting, k-means clustering, KNN. Besides deep learning, we will use TensorFlow, an open-source deep learning library that will be carried out for Kuwait's power plants. TensorFlow is a programming interface for high-level neural networks (API). The study will provide the plant operators with critical insight into the dispersion of air pollutants around the plant and alert them in case of high concentrations to safeguard the workers' health.

Kuwait is one of the largest producers of oil in the world, with many refineries all over the country. Since Kuwait is only 17818 m^2 in area, it is at an even higher risk of air pollution from refineries. It is for those reasons that this topic is a common area of interest in research in Kuwait. There were several studies reported discussing the Air Pollution status in the state of Kuwait. Abdul- Wahab and Bouhamra, 1999 studied the air pollution of the primary and secondary pollutants in Kuwait's residential area (Mansouriya). They concluded that Kuwait is affected not only by pollutants produced by the petroleum refineries and petrochemical industries but also by road traffic and other sources. Bader Al Azmi, V. Nassehi, and R. Khan, 2008 were also able to confirm that the traffic volume in a residential area (Al Rabia) significantly affects the pollutants level. Raslan Alenezi and Aamir Ashfaque, 2008 devoted their study to Kuwait's North, particularly the oldest and busiest cities in Kuwait (Al-Jahra). The study focused on daily averages and maximum hourly concentrations for each season. It could also confirm that road traffic was a significant source of air pollution in the Al-Jahra area. Raslan Alenezi, Bader Al-Anzi, and Aamir Ashfaque, 2010 reassessed Air Quality's seasonal influence, and their study centered on the seasonal impact on the Air Quality in Al Jahra. Alenezi and Ashfaque reported that hourly averages for CO_2, O_3, and PM_{10} are higher in winter than in summer due to prevalent meteorological conditions. After validation, a computer dispersion model (CALINE4) was also used in their study to predict the CO concentrations as a function of the number of cars in the vicinity of the school. Results showed high levels of NM-HC almost all of the times above the Kuwaiti EPA limit, while

CO and Nitrogen Dioxide (NO_2) concentrations were found to be below the KU-EPA limits. Moreover, Ettouney 2009 has concluded insightful results between the meteorological conditions and pollutants. Ettouney finds out that the daily averages of O_3 increase directly from the higher solar intensity. And from (2001-2004) the NO_2 pattern showed a continuous increase due to the population, industrial growth, number of motor vehicles, and power generation. Al Adwani assessment of air quality in Al-Jahra centered on presenting the daily averages and maximum hourly concentrations for each season of the year 2008 and elaborating on the impacts of the pollutants with the diurnal variation and overall air quality in the urban area. And discussing the significant sources of Air pollution in the Al-Jahra area, such as road traffic. Al Adwani used a mobile APML to monitor the effect of fuel change from lead to unleaded gasoline containing an additive Methyl Tertiary Butyl Ether (MTBE) on air quality in a specific heavy-traffic residential area of Kuwait. They have used a mathematical model to simulate this influence on ambient air quality. They reported that MTBE–gasoline enhances the degree of combustion and hence lowers CO and hydrocarbons (HC) emissions. However, it increases the emissions of NOx and particulate matter. Al Adwani's research confirmed that the primary sources of pollutants are the northern oil fields, gravel quarries, freeways, Power desalination plants, and wastewater plants. Thus, an important reason for selecting Al-Jahra as a study area is a fact that being the largest, ancient, and historical residential suburban site. The correlation shows a decrease in ozone concentration with an increase in Nitrogen oxide (NO_2) and non-methane hydrocarbons (NM&HC). Bader Al-Azmi made a comparative assessment of ambient air quality in Rabia Area from 2001 to 2004, he decided to investigate the level of pollution of SO_2, NO_2, and O_3, and after the comparison, the study showed that the concentration of all pollutants for the year of 2001 was higher than 2004. The pollutants in autumn and winter were higher than in summer and spring in Rabia Area. Al-Harbi expanded the study and comprehensively compared two urban localities, Fahaheel and al Rabia. Data analysis was made to compare the two localities from the pollutants value. In 2009, Ramadan measured the average concentrations of NO, NO_2, SO_2, H_2S, NH_3 in the southern part of Kuwait. He then produced spatial distribution maps and compared the findings with the applicable air quality standards promulgated by Kuwait Environment Public Authority (KEPA).

Furthermore, the Chemical Mass Balance (CMB) model was developed to quantify the contribution of the primary emission sources. Results show that the main emission sources in

Fahaheel were petroleum downstream facilities and highway traffic, which accounted for 69% and 17%, respectively. In Al-Rabia City, highway traffic and the area's commercial activity accounted for 79% and 13%, respectively. The average daily profiles and seasonal variations were constructed for each pollutant and their descriptive statistics using Statistica 6 software. Concentration roses of air pollutants and the Chemical Mass Balance (CMB) model were performed using advanced Microsoft Excel add-ins programs. Al-Hidar's study evaluated the outdoor air quality in Al-Mansoriah residential area. Data collected over five years (2000-2004) was stored and manipulated with EnviDas data acquisition software. Four pollutants were chosen to be studied: SO_2, CO, NO_2, and O_3, over five years and divided into six categories from Good to Hazardous. The four pollutants studied resulted in a "good" and "moderate" ambient air quality. This was expected as it is only a residential area surrounded by a few sources of pollution. Ettouney assessed air pollution in Umm-Alhyman. Data analysis provided insights into the status and levels of air pollution and how it compares to international limits. The ISC-AERMOD software package made predictions of the measured data. By using the ISCST3 model option to predict pollutant dispersion and compare it against the pollutants database for 2003 in Al- Jahra, ANN was performed to forecast ozone concentration and to prove that with high wind speed, the ozone concentration will be high. It was concluded that all data are within the air quality standards except for PM_{10} due to the short rain season. Also, it was observed that the NOx concentrations increase because of increasing industrial activities and vehicles.

However, this study will significantly enhance what was previously done by representing the meteorological conditions in better visualizations. The study will not only stop at the graphical representation but also by mentioning the high episodes of wind direction causality and the seasons that were affected by high pollution, which can be considered a formidable change from the former studies after using robust software programs for data analysis. Previously published research was exciting and insightful, clearly elaborating data analysis and conveying meaningful results to the reader. This thesis will take Air Quality to a higher level of data analysis and modeling. Most of the former papers use Microsoft Excel add-ins programs, Statistica 6, ISC-AERMOD software, etc. However, these tools are efficient and powerful. Still, regarding scalability and automation, it is always better to use a programming language to facilitate the work and automate many functions efficiently to save time. Data analysis will be performed using R-studio, and R. R is a free software

environment highly used in academia for statistical computing and graphics. It is an open-source language heavily tailored for handling data, from data wrangling through data visualization to machine learning. R's functionality is enormous, and kind of like our universe, R's universe is continually expanding. The main package that will be used for air quality analysis is called openair. Openair is an R library that was developed for analyzing air quality data. Openair package can show deep details that other software can't handle, especially when the data contains many time variations. This library is extensively used in academia and worldwide research with remarkable results. For the modeling part, python is the suitable choice. It is highly preferred over R due to its performance in data manipulation and model development. After developing the model in python, it will be deployed for production. For every efficient machine learning model, it should be used by other users by placing the model into an environment where it can complete the work it was designed to. It might be integrated with mobile applications or any software program. In this thesis, the model will be deployed using the Streamlit library in python. Streamlit is a powerful python library that can quickly build applications framework. It has been built to create interactive web applications around the given data for projects in data science and machine learning. After deployment in the Streamlit cloud, the model will be hosted, and a customized shareable URL can be shared to test your model anytime. It has many templates and prebuilt widgets that will undoubtedly reduce the time spent building the UI. In addition, many available features help the user a lot with the UX.

It is paramount to monitor and study the primary and secondary pollutants' behavior to demonstrate a better comprehension of their trends and impact on the surrounding environment. There are limited studies conducted on the relations between the pollutants, how they are correlated to each other, and how they are affected by the meteorological conditions. There are even scarcely developed models that can predict the primary and secondary pollutants, which can ease the forecasting to the authorities to perform the precautions and commit the awareness, if any. This study will focus on assessing and analyzing the level of NO_2, O_3, SO_2, $PM_{2.5}$, and Total Hydrocarbons (THC) in the North of Kuwait from two monitoring stations in Kuwait (Al Jahra, Saad Al – Abdullah), for a vast data set starting from 2014 to 2019, to conclude helpful information and recommendations, considering the number of violations against KEPA and the frequency of the episodes, and the correlations between the pollutants besides the effect of the meteorological

conditions over these pollutants using advanced statistical analysis by different software programs which weren't used before for analysis in Kuwait. The thesis will focus on the relation between NO2 and O3 where the rationale for choosing to analyze the relationship between ozone (O3) and nitrogen dioxide (NO2) in the thesis is grounded in several key factors:

1. **Chemical Interplay between O3 and NO2**: O3 and NO2 are closely related in atmospheric chemistry. NO2 is a significant precursor to ground-level ozone. Under the presence of sunlight, NO2 undergoes photolysis, leading to the formation of O3. This relationship is pivotal in understanding urban air pollution dynamics, especially in areas with high vehicular and industrial emissions.
2. **Indicator of Urban Air Quality**: Both O3 and NO2 are critical indicators of urban air quality. NO2 is primarily emitted from fossil fuel combustion, especially from vehicles and industrial processes. O3, while not directly emitted, is formed through chemical reactions involving NO2 and volatile organic compounds (VOCs) in the presence of sunlight. Analyzing their relationship provides insights into the sources and dynamics of urban air pollution.
3. **Public Health Concerns**: Exposure to high levels of NO2 and O3 is associated with various adverse health effects, including respiratory problems, cardiovascular diseases, and aggravated asthma. Understanding their interaction is crucial for public health risk assessment and for formulating strategies to mitigate these risks.
4. **Environmental and Climatic Implications**: The interaction between NO2 and O3 has broader environmental and climatic implications. O3 is a greenhouse gas and contributes to the warming of the atmosphere. Additionally, the presence of NO2 can influence the oxidative capacity of the atmosphere, affecting the lifespan of other pollutants and greenhouse gases.
5. **Policy and Regulatory Frameworks**: Understanding the NO2-O3 relationship is essential for developing effective air quality management policies and regulations. By analyzing this relationship, policymakers can identify targeted strategies to control the emissions of NO2 and other precursors to reduce O3 formation.
6. **Advancements in Air Quality Modeling**: Analyzing the NO2-O3 relationship using advanced statistical methods and machine learning offers an opportunity to enhance

predictive models of air quality. These models can be crucial for forecasting pollution episodes and implementing timely preventive measures.

7. **Region-Specific Relevance**: In Kuwait, where the thesis is focused, the climatic conditions (high temperatures and abundant sunlight) can significantly influence the NO2-O3 chemistry. Understanding this relationship in the context of Kuwait's specific environmental and industrial landscape is vital for local air quality management.

The decision to focus on the O3 and NO2 relationship is underpinned by the central role these pollutants play in urban air chemistry, their health and environmental impacts, their relevance to policymaking, and the specific context of Kuwait's environmental conditions. This thesis will also deploy the concept of Machine Learning and Deep Learning to build up valuable models that can predict the future. It is noteworthy that from 2014 to 2018 will consist of training and validation data, and 2019 will be the test data. The main objective of this research is to understand the factors that determine the amount of air pollution in many different parts of the world.

Aims:

1. **Evaluate Air Quality in Urban Areas of Kuwait**: To conduct a detailed assessment of air quality in specific urban localities in Kuwait, focusing primarily on Al Jahra and Saad Al Abdullah.
2. **Understand Seasonal Patterns of Pollutants**: To identify and analyze the seasonal variations in the concentrations of key air pollutants, specifically nitrogen dioxide (NO2) and ozone (O3).
3. **Advanced Data Analysis Implementation**: To apply sophisticated statistical methods and machine learning models for a more in-depth analysis of air quality data.

Objectives:

1. **Hourly Pollution Level Assessment**: To determine the hourly levels of NO2 and O3 pollutants, providing a granular view of air quality over time.
2. **Correlation Analysis**: To establish correlations between NO2 and O3 concentrations, investigating their interplay and how they affect each other under varying conditions.
3. **Exceedance Analysis**: To highlight the number of times pollutant levels exceed the standards set by the Kuwait Environmental Public Authority (KEPA), indicating the severity of air pollution.

4. **Modeling and Predictive Analysis**: To conduct diagnostic analysis for evaluating the performance of various machine learning models, identifying the most effective models for air quality prediction in Kuwait.
5. **Comparative Methodology Analysis**: To compare the effectiveness of different machine learning and deep learning models, such as boosting, bagging, and neural networks, in analyzing air quality data.
6. **Stakeholder Impact Assessment**: To understand how the findings of the thesis are relevant and beneficial to various stakeholders in Kuwait, including policymakers, public health authorities, environmental agencies, and the public.

In essence, The Thesis aims to provide a comprehensive and advanced analytical understanding of air quality patterns in Kuwait, leveraging modern data science techniques to offer actionable insights for effective air quality management and policymaking. From a technical perspective, we will develop a model that can be used to predict the levels of air pollution in real-time. This model will help reduce the harmful effects of air pollution on our health and the environment. A mathematical model is developed to predict air pollution in real-time. The model contains a series of equations that are used to relate different levels of environmental parameters to the number of air pollutants in the atmosphere. These equations are based on the relationship between environmental factors and the number of air pollutants in the atmosphere. The input parameters of the model are the meteorological conditions consisting of the ambient temperature, wind speed, wind direction, solar intensity, and the relative humidity of the environment. The output parameters are the concentrations of ozone and nitrogen oxides in the air. In the realm of environmental research, understanding air quality dynamics is crucial, particularly in regions with rapid industrialization and urbanization like Kuwait. This article delves into a comparative analysis of several key studies on air quality in Kuwait, culminating in a discussion of The Thesis which is a recent study that pushes the boundaries of air quality research in the region. We'll explore how The Thesis not only aligns with previous findings but also introduces advanced methodologies and insights, adding substantial value to the existing body of knowledge. Let’s have an overview between The Thesis research and previous studies in Kuwait and to expand it the comparison between worldwide studies.

a) Comparison with Previous Studies:

1. **Seasonal Variation**:
    - **Previous Studies**: Found higher concentrations of NO2 in winter and lower in summer, attributing this to meteorological conditions and human activities, such as reduced fossil fuel consumption in summer due to vacations and higher dispersion due to wind.
    - **Thesis Research** Also observed higher NO2 concentrations in winter compared to summer, aligning with previous findings. However, the thesis found a robust negative correlation between NO2 and O3, which wasn't explicitly mentioned in previous studies.
2. **Methodological Advancements**:
    - **Previous Studies**: Used traditional monitoring and statistical analysis techniques to assess air quality. Some included machine learning approaches but not in depth.
    - **Thesis Research**: Applied advanced statistical methods and machine learning models, offering a more sophisticated analysis. The use of the CatBoost algorithm for predicting pollutant levels is a notable advancement.
3. **Pollutant Concentrations and Exceedances**:
    - **Previous Studies**: Reported exceedances in pollution levels, particularly for NO2 and PM10, and highlighted the impact of traffic and industrial activities.
    - **Thesis Research**: Also highlighted exceedances against KEPA standards, but with a more refined analysis due to advanced modeling techniques.

b) Synthesize Existing Knowledge:

The key findings and common themes from the literature include:

- **Seasonal Variations**: All studies agree on the significant impact of seasons on NO2 and O3 concentrations, with higher NO2 levels in winter and higher O3 levels in summer.
- **Influence of Human Activities and Meteorology**: Traffic, industrial activities, and meteorological conditions (like wind speed and temperature) are crucial factors influencing air quality.
- **Methodological Evolution**: Earlier studies used more traditional monitoring and analysis methods, while The Thesis research incorporate advanced statistical and machine learning techniques for more nuanced insights.

c) Addressing the Weakness in Kuwaiti Studies:

The literature review does indeed focus heavily on studies conducted in Kuwait. This limitation is primarily due to the specific regional focus of the research, aiming to understand and address local environmental and climatic conditions unique to Kuwait. These studies were prioritized because they provide relevant data and insights that are directly applicable to the Kuwaiti context, which is essential for formulating effective local air quality management strategies.

d) Contribution of the Thesis Research:

The Thesis Research contributes significantly in several ways:

1. **Advanced Analytical Techniques**: By employing sophisticated machine learning models, our study provides more accurate and granular insights into air pollution patterns in Kuwait. This is crucial for developing targeted mitigation strategies.
2. **Predictive Capabilities**: The use of predictive models like CatBoost offers stakeholders the ability to forecast pollution levels more accurately, facilitating proactive rather than reactive measures.
3. **Comprehensive Understanding**: The Thesis research bridges the gap between traditional air quality monitoring and cutting-edge data analysis, providing a more holistic view of air quality trends.
4. **Policy Implications**: The detailed and predictive nature of the thesis findings can better inform policymakers and environmental agencies, leading to more effective policy decisions and public health advisories.
5. **Future Research Foundation**: The Thesis Research sets a precedent for future air quality studies in Kuwait and similar regions, offering a methodology that can be replicated or expanded upon for further research.

In essence, the thesis enhances the understanding of air quality dynamics in Kuwait using advanced methodologies, offering actionable insights for stakeholders, and paving the way for further research in this critical area.

Comparative Analysis:

1. Seasonal Variation and Pollutant Concentration:

- **Previous Studies**: Consistently observed higher NO2 levels in winter, attributed to reduced dispersion and human activities, such as lower fuel consumption in summer.

- **The Thesis:** Echoes these findings but introduces a novel insight - a robust negative correlation between NO2 and O3, highlighting intricate interplays between these pollutants across seasons.

2. Methodological Evolution:

- **Previous Studies**: Relied on traditional monitoring methods and basic statistical analysis, providing foundational knowledge on air quality patterns.
- **The Thesis:** Advances this approach by employing sophisticated machine learning models, like the CatBoost algorithm, offering a more nuanced and accurate analysis of air quality data.

3. Exceedance and Environmental Impact:

- **Previous Studies**: Reported pollution level exceedances, particularly for NO2, and emphasized the impact of traffic and industrial activities on air quality.
- **The Thesis:** Reaffirms these exceedances but with a refined analytical lens, enhancing the understanding of pollution sources and patterns.

Adding Value: The Thesis in the Context of Kuwaiti Air Quality Research

The Thesis marks a significant leap in air quality research in Kuwait, offering several key contributions:

- **Incorporation of Advanced Data Analysis**: The utilization of advanced statistical methods and machine learning models sets The Thesis apart. This approach allows for a more detailed and precise understanding of air pollution trends, crucial for effective policy formulation and environmental management.
- **Enhanced Predictive Capabilities**: The predictive modeling in The Thesis provides stakeholders with tools to forecast pollution trends more accurately. This predictive power is invaluable for planning proactive environmental interventions.
- **Deeper Insights into Pollutant Dynamics**: The study's novel findings, like the negative correlation between NO2 and O3, offer a deeper understanding of pollutant interactions, which is vital for developing targeted pollution reduction strategies.
- **Policy and Public Health Implications**: The detailed insights from The Thesis can better guide policymakers and environmental agencies in Kuwait, leading to more informed decisions and effective public health advisories.

- **Foundation for Future Research**: By pioneering the use of advanced methodologies in Kuwaiti air quality research, The Thesis paves the way for future studies, potentially expanding to other regions with similar environmental challenges.

Number	Study Title	Focus of the Research	Best ML Model	Results/Findings	Recommendation	Software Used
1	Al-Azmi-2008 - Comparative Assessment of Ambient Air Quality in Rabia Area for Years 2001 and 2004	Assessing pollution levels of SO2, NO2, and O3 in the Rabia area of Kuwait for the years 2001 and 2004	Not specified	Pollution concentration for the year 2001 was greater than 2004, indicating growing environmental awareness and changes like the use of cleaner fuel treatment	To assess air quality in Rabia area and address major sources of SO2 emission in Kuwait	Not specified
2	Alenezi-2011 - Assessment of Ambient Air Quality in Al Jahra Governorate for 2008	Analyzing air pollution data covering major pollutants such as CO, methane, PM10, and NO2 in Al-Jahra, Kuwait	Not specified	Detailed analysis of air pollution and meteorological data for 2008, showing variations in pollutant concentrations in different seasons	Not specified	Not specified
3	Al-Haider-2008 - Outdoor air quality data analysis of Al-Mansoriah residential area (Kuwait)	Analyzing outdoor air quality data for five years (2000-2004) in Al-Mansoriah, Kuwait, using AQI based on SO2, CO, O3, and NO2	Not specified	General decrease in AQI values, indicating 'good' and 'moderate' air quality over the study period	Stricter regulations should be applied for monitored areas in Kuwait	Not specified
4	Al-Harbi-2014 - Assessment of Air Quality in two Different Urban Localities	Assessing air quality in two urban localities, Fahaheel and Al-Rabia City, with respect to NO2, O3, CO, NMHCs, and SO2【76†source】	Not specified	Comparison of diurnal and seasonal variations of pollutants in Fahaheel and Al-Rabia, including exceedances of KUEPA air quality thresholds and sources of pollution【78†source】	Not specified	Not specified
5	Ettouney-2010 - Daily and Seasonal Changes of Air Pollution	Assessment of air pollution in Umm-Alhyman, Kuwait, focusing on various air pollutants and meteorological parameters over a period of four days in different years【82†source】	ISC-AERMOD software package	Increase in measured concentrations of SO2, NOX, and CO between 2001 and 2008 due to population growth and industrial activities, with comparisons to international standards【84†source】	Not specified	ISC-AERMOD
6	Al-Salem-2010 - Monitoring and Modelling the Trends of Primary and Secondary Air Pollution Precursors: The Case of the State of Kuwait	Reviewing recent activities and trends of primary and secondary air pollutants in Kuwait, focusing on areas sensitive to precursors like NOx and VOCs, and areas adjacent to industrial activities【88†source】	Not specified	Recommendations for better monitoring of air pollutants in Kuwait, including better placement of monitoring stations and modern prediction techniques, with a focus on NO2 emissions and the impact of traffic and industrial sources【91†source】	Improve monitoring stations placement, gather coherent metrological data, record a wider range of pollutants, and revise outdoor air quality rules for healthier living conditions【91†source】	Not specified
7	Thesis Research Air Pollution Assessment and Computational Air Quality Modeling in North Kuwait	Evaluating hourly data of air quality in urban localities of Al Jahra and Saad Al Abdullah, focusing on NO2 and O3 pollutants【110†source】	CatBoost algorithm	A robust negative correlation between NO2 and O3. CatBoost algorithm exhibited the highest accuracy for predicting pollutant levels【110†source】	Develop a real-time predictive model for air pollution management and policy-making【115†source】	Python, R programming language, TensorFlow
8	Seasonal Influence on the Ambient Air Quality in Al Jahra City for Year 2010	Monitoring air quality in Al Jahra city, Kuwait, focusing on CO, CH4, PM10, NM-HC, SO2, NO2, and meteorological conditions in 2010【121†source】	Not specified	Higher concentrations of pollutants in winter except for CO2, O3, and PM10. NO2 concentrations and temperature are major factors influencing O3 levels【124†source】	Not specified	Not specified
9	An assessment of the air pollution data from two monitoring stations in Kuwait	Assessing air pollution data from two monitoring stations in Kuwait, covering major pollutants and meteorological parameters from 2001 to 2004【128†source】	Not specified	Most pollutants below international standards except for particulate matter. Increase in nitrogen oxides due to vehicle and power expansion【132†source】	Not specified	Not specified
10	Abdul-Wahab-2008 - Analysis of air pollution at Shuaiba Industrial Area	Study of diurnal and monthly variations of major primary and secondary air pollutants in Shuaiba Industrial Area, Kuwait, relative to meteorological parameters【149†source】	Not specified	Significant diurnal and seasonal variations in pollutant concentrations, with higher levels of nitrogen oxides, sulfur dioxide, and suspended dust in summer. Ozone shows marked seasonal variation, peaking in spring and late summer【152†source】【153†source】【154†source】	Not specified	Not specified
11	Al-Salem-2008 - Comparative assessment of ambient air quality in Fahaheel and Al-Riqa	Investigation of air pollution in Fahaheel and Al-Riqa, Kuwait, focusing on O3, NO, NOx, CO, H2S, and NH3, including analysis of exceedances of KUEPA air quality limits, primary air pollution sources, diurnal patterns, and the 'weekend effect' on O3 levels【158†source】【160†source】	Not specified	High levels of ozone above human health threshold in both urban areas, with CO, NOx, and NO levels higher in Fahaheel. Identification of primary pollution sources and analysis of diurnal patterns and weekday/weekend variations in pollutant levels【158†source】【162†source】【163†source】【164†source】【165†source】【166†source】	Not specified	Not specified
12	Al-Salem-2009 - Ambient air quality assessment of Al-Mansoriah Residential Area	Analysis of air quality in Al-Mansoriah, Kuwait, covering O3, NMHC, CH4, NO, NO2, CO, H2S, and SO2, focusing on exceedances of KUEPA limits, diurnal patterns, sources of pollutants, and 'weekend effect' on O3 levels【177†source】	Not specified	High levels of ozone, exceedances of permissible levels of NMHC, acute and chronic levels of SO2. NO showed strong diurnal peaks, especially in cooler months. Clear 'weekend effect' observed, suggesting Al-Mansoriah is a NOx sensitive region【177†source】【178†source】【179†source】【180†source】	Not specified	Not specified

As a recap, The Thesis not only aligns with the broader trends observed in previous Kuwaiti air quality studies but significantly builds upon them by introducing more advanced analytical techniques and deeper insights. This study represents a pivotal addition to Kuwaiti environmental research, offering a more comprehensive, accurate, and actionable understanding of air quality dynamics. Its implications extend beyond academia, promising to inform better policy-making and environmental management practices in Kuwait and similar regions worldwide. Internationally, Our Thesis stands out in its specific focus on the relationship between NO2 and O3 using advanced machine learning techniques, particularly XGBoost. While international studies also leverage ML and advanced statistical methods, they vary in pollutant focus, with many emphasizing PM2.5 and a broader range of methodologies. The unique aspect of Our Thesis lies in its nuanced exploration of the NO2-O3 dynamic, contributing a specific and detailed understanding to the field of air quality research.

Number	Study Title	Focus	Best ML Model	Results/Findings	Similarities
1	Aerosol and Air Quality Assessment of ML Algorithms	Short-term PM10 and PM2.5 forecasting in urban Polish areas.	XGBoost	XGBoost outperformed other models in PMx forecasting	Use of XGBoost and ML for air quality
2	Forecasting Air Pollution Particulate Matter (PM2.5) Using Machine Learning Regression Models	PM2.5 forecasting in Taiwan.	Gradient Boosting Regression	Gradient boosting regression showed better performance	Gradient boosting technique for air pollution prediction
3	Machine Learning Approaches for Outdoor Air Quality	Various ML methods for air quality modeling.	Varied, including Random Forest and Ensemble Learning	Increasing trend in ML studies for air quality; varied effectiveness of models	Employment of advanced ML models
4	Enhancing Air Quality Forecasting in China Using a Modified XGBoost Model	PM2.5 forecasting in China using ML models.	Modified XGBoost	Enhanced forecasting of PM2.5 with modified XGBoost model	XGBoost model for enhanced prediction
5	Application of Open Air Model for Urban Air Quality Data Analysis in Indonesia	Urban air quality data analysis using openair model in R.	Openair (Trend Analysis)	Effective analysis of long-term air quality data	Statistical analysis of air quality
6	Openair – Data Analysis Tools for the Air Quality Community	Development of openair software for air quality data analysis.	Openair (Comprehensive Analysis)	Future development of openair for comprehensive air quality analysis	Data analysis and visualization tools
7	2479-Thesis_Dissertation Main Document-46769-1-2-20231129.pdf (Our Thesis)	Relationship between NO2 and O3 in Kuwait using ML techniques.	CatBoost	Strong negative correlation between NO2 and O3; seasonal variations	Advanced ML techniques for air quality assessment
8	A Hybrid CNN-LSTM Model for Forecasting	Forecasting air quality using a hybrid CNN-LSTM model.	CNN-LSTM	CNN-LSTM model shows low error and short training time	Use of advanced ML models for forecasting
9	Urban Air Pollution Mapping Using Fleet Vehicles as Mobile	Mapping urban air pollution using fleet vehicles.	Random Forest	Random Forest performs best on all metrics	Use of Random Forest for air quality mapping
10	Machine Learning Algorithms in Air Quality Modeling	General air quality modeling using machine learning.	Random Forest	Random Forest – a tree-based ensemble learning approach	Use of Random Forest for air quality modeling
11	Machine Learning Algorithms for Predicting Air Pollutants	Predicting air pollutants using machine learning.	Not specified	Implemented 4 classification models	Use of ML for predicting air pollutants
12	Application of Density Plots and Time Series Modelling to the	Density plots and time series modeling for air quality.	ARIMAX	ARIMAX demonstrated strong performance	Time series modeling for air quality
13	Stochastic Gradient Boosting	Stochastic gradient boosting for air quality.	Stochastic Gradient Boosting	Accuracy of gradient boosting can be substantially improved	Use of boosting techniques for air quality

Chapter 2
Methodology

2.1 Data collection

The inhabitants of Kuwait's cities, towns, and villages are exposed to different levels of air pollution, which also vary throughout the day. The information regarding clean and poor air quality enables us to mitigate the risks and identify the most and least desirable hours in terms of the level of air pollution. This information can be grasped and extracted from the data. Data always has a better idea. When the data is collected and refined, it can help cities design intelligent measures and take future-thinking actions to improve air quality across various sectors. In this thesis, the concern is centered on studying Air Quality in the North of Kuwait. The designated monitoring stations were Al Jahra Monitor and Saad Al Abdullah Monitor. More than seven years of hourly data points were collected with the help of KEPA, starting from 2013 to 2020. But the probability of collecting anomalous data has increased nowadays due to its considerable size and origin for heterogeneous sources. And since the study is focused on hourly records, it is expected to find days with no records due to different reasons like maintenance, shutting down the monitor, lack of labor, and climate reasons. High-quality data can lead to better model performance and better predictions. Thus, the next section will discuss how to apply data preprocessing and its uses in this thesis. Data preprocessing is fundamental in data science, machine learning, and artificial intelligence. While gathering and recording the data, three main factors could contribute to the quality of the data:

1. **Accuracy:** Sometimes, the values are incorrect and deviate from the expected values, and that is due to various reasons, which include:

- Human errors in data entry.
- Deliberately submitting incorrect values for disguising.
- Input fields may have some incorrect formats.

2. **Completeness:** Lacking feature values, and missing complete columns, the dataset might face incompleteness due to the following:

- Unavailability of the data.

- Cancellation of inconsistent data.
- Deletion of irrelevant data.

3. **Consistency:** The aggregation of the data is inconsistent.

While merging the information, some features are probably affected when the data is incomplete. This yields to make the structure of the data indecipherable to the stakeholders.

Hence, to ensure high-quality data, applying to preprocess is crucial because it will make the data easy to explore, well interpretable, and less complex in applying other processes in the data.

Data Preprocessing

Data processing is an important aspect of machine learning and data science. It involves cleaning, transforming, and organizing the data to make it suitable for building predictive models. This can include filtering out irrelevant or missing data, transforming the data into a specific format, and scaling or normalizing the data to ensure that it can be effectively used in a machine-learning algorithm. Data processing is a crucial step in the machine learning pipeline, ensuring that the data is ready for further analysis and modeling. Data preprocessing is a vital step in data science as it is essential to user workflow. It is deemed a suitable data mining technique heavily used to transform the raw data in a practical, readable, and efficient format that a computer can work with. There are several steps involved in data preprocessing.

2.2.1 Data Cleaning

Data cleaning is an essential step in the machine learning and data science process. It involves identifying and correcting errors or inconsistencies in the data and filling in missing or incomplete values. This is necessary because real-world data is often messy and imperfect, and these issues can affect the accuracy and performance of a machine-learning model. Data cleaning can be time-consuming and labor-intensive, but it is critical to ensure that the data is usable and trustworthy. Some standard techniques used in data cleaning include handling missing values, dealing with outliers, and removing duplicates. Data cleaning is fixing, detecting, and correcting irrelevant or false records from the database. When combining multiple data sources, there is a high chance for the data to be duplicated or mislabeled. Thus, based on a justified mathematical

assumption, the data will confront irrelevant information and missing parts that might be replaced or deleted. Data cleaning involves handling missing records, noisy data, outliers, peculiarities, etc. The combination between the metrological conditions and the pollutant concentrations was merged hourly in one table format.

This made the dataset very ginormous and led to many nulls and outliers in different years. Once dealing with missing data, multiple approaches can be used, and they are pointed out below:

1. **Removing the training example:** The user can drop the missing value directly, which is usually discouraged. It leads to the loss of data that may add value to the dataset.
2. **Filling manually:** This approach is not accurate in massive datasets, and it is time-consuming.
3. **Using a standard mathematical value:** The missing value can be replaced by the central tendency like (mean, median, mode).
4. **Interpolation & Regression:** This is the most probable method to fill missing values using algorithms like regression and decision trees to predict and replace the missing value.

The missing values of unrecorded pollutants concentration or the values of the metrological conditions from the monitoring station have been compensated with justified mathematical assumptions, and to ensure high quality and extreme performance of an algorithm, a simple comparison was made to compare between the model metrics at different strategies of handling the missing values like applying an interpolation between the former and the next value, forward-filling, and back-filling, Simple-Imputer, KNN-Imputer, Iterative-Imputer. After applying the classical machine learning algorithms, the results show that the interpolation technique was more convenient than other techniques with better model metrics.

- Interpolation: Filling the missing values in a series by using only linear interpolation.
- Simple-Imputer: Filling the missing values in a series by attributing them to a constant statistical value (mean, mode, median).
- KNN-Imputer: Filling out or predict the missing values in a series by using the approach of the KNN algorithm rather than other naïve approaches of filling the values by mean or the median.

- Iterative-Imputer: Filling out the missing values by using each feature to be modeled as a function of the other features. Thus, it will be considered a regression problem where the missing values in the dataset can be predicted.

This thesis will compare different strategies for handling the missing values and fit them into several machine-learning models to ensure the best evaluation metrics and model robustness. This approach gives better estimation when dealing with noisy data. It is more rational to be applied in the air quality data for the metrological conditions and the pollutant concentrations.

Table 2.1: Simple-Imputer Metrics.

Model	MAE	MAPE %	MSE	RMSE	R^2 (train)	R^2 (validation)	R^2 (test)
Linear Reg	0.008698	0.8698	0.000143	0.01197	0.58	0.57	0.56
KNN	0.006925	0.6925	0.00009885	0.009942	0.74	0.73	0.72
RandomForest	0.006894	0.6894	0.00009253	0.009619	0.77	0.75	0.73
XGBoost	0.006646	0.6646	0.00008891	0.009429	0.8	0.77	0.74

Table 2.2: Interpolation technique Metrics

Model	MAE	MAPE %	MSE	RMSE	R^2 (train)	R^2 (validation)	R^2 (test)
Linear Reg	0.008511	0.8511	0.00013947	0.01180	0.59	0.58	0.57
KNN	0.006729	0.6729	0.00009454	0.009723	0.752	0.74	0.73
RandomForest	0.006602	0.6602	0.00008952	0.009461	0.8	0.77	0.74
XGBoost	0.006506	0.6506	0.00008567	0.009256	0.81	0.78	0.75

Table 2.3: Iterative-Imputer Metrics

Model	MAE	MAPE %	MSE	RMSE	R^2 (train)	R^2 (validation)	R^2 (test)
Linear Reg	0.008436	0.8436	0.00013874	0.01177	0.6	0.59	0.58
KNN	0.006666	0.6666	0.00009392	0.009691	0.758	0.745	0.73
RandomForest	0.006621	0.6621	0.00008952	0.009403	0.81	0.78	0.75
XGBoost	0.006466	0.6466	0.00008680	0.009317	0.82	0.79	0.76

Table 2.4: KNN-Imputer Metrics with eight nearest neighbors

Model	MAE	MAPE %	MSE	RMSE	R^2 (train)	R^2 (validation)	R^2 (test)
Linear Reg	0.008303	0.8303	0.00013275	0.01152	0.6	0.595	0.58
KNN	0.006619	0.6619	0.00009280	0.009633	0.76	0.75	0.74
RandomForest	0.006557	0.6557	0.00008666	0.009309	0.82	0.79	0.76

XGBoost	0.006393	0.6393	0.00008378	0.009153	0.83	0.8	0.78

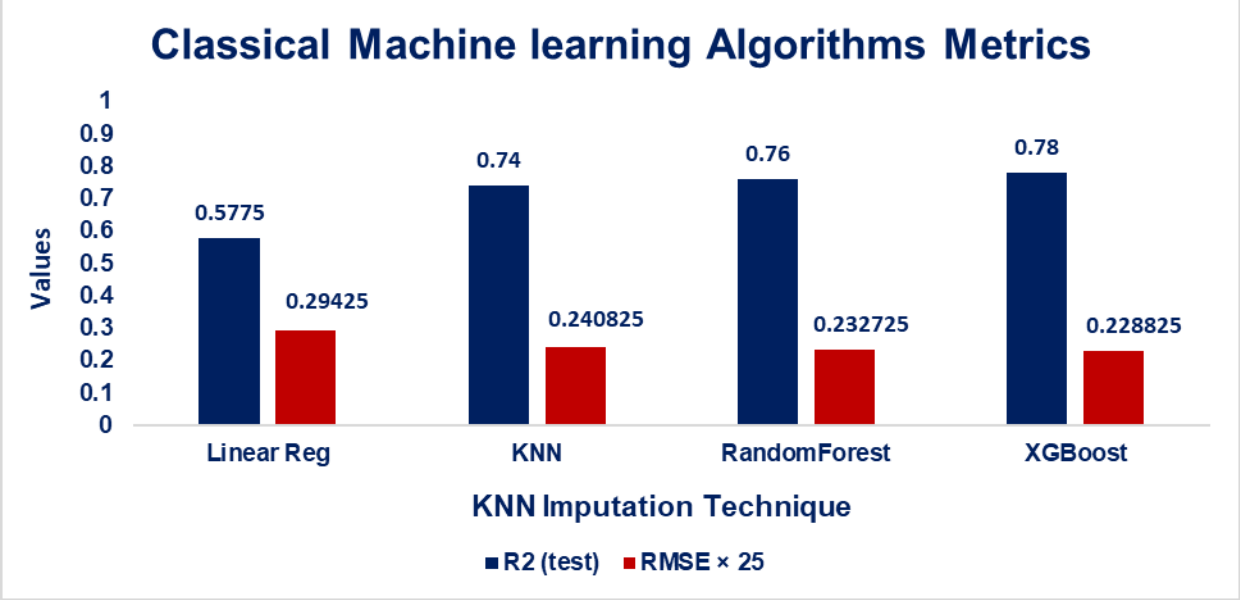


Figure 2.1:The comparison results for Evaluation Metrics

From the above tables and figure (1), we can conclude that after imputing the missing values using the KNN algorithm technique, KNN can compensate for the missing data better than other techniques. Hence, we will continue following this mathematical strategy to develop advanced, sophisticated Machine learning and Deep learning algorithms with hyperparameters tunning to each model. KNN can compensate for the missing data better than other techniques based on a comparative analysis of different methods for handling missing data. This conclusion was drawn after implementing various techniques for imputing missing values, including the K-Nearest Neighbors (KNN) algorithm, and comparing their performance.

Explanation of how this conclusion was derived:

1. **Comparative Analysis of Missing Data Imputation Techniques:**
    - Various methods were employed to handle missing data in the dataset. This included techniques like interpolation, Simple-Imputer, Iterative-Imputer, and the KNN-Imputer.
    - Each of these methods was applied to the dataset, and their performance was evaluated using machine learning models.
2. **Performance Evaluation of Imputation Techniques:**

- The effectiveness of each missing data imputation technique was assessed based on how well they improved the performance of subsequent machine learning models.
- Key performance metrics (such as MAE, MAPE, MSE, RMSE, and R2 for train, validation, and test sets) were used to evaluate each method's impact on the models.

3. **Superior Performance of KNN-Imputer:**
    - Upon analysis, it was found that the KNN-Imputer technique outperformed other methods in terms of the mentioned metrics.
    - This led to the conclusion that the KNN algorithm was more effective at compensating for missing data compared to the other techniques tested.
4. **Decision to Continue with KNN for Advanced Modeling:**
    - Given the superior performance of the KNN-Imputer in handling missing data, the decision was made to continue using this technique in the development of more sophisticated machine learning and deep learning models with hyperparameter tuning.

The Grubbs' test has been applied in excel using the software package xlstat to test the outliers. Grubbs' test is more sophisticated than other tests. Detecting outliers and peculiarities in a dataset is preferable to testing for one outlier at a time. Any outlier detected in the dataset will be removed with reasons consideration. Then, the test is replicated until no outliers are detected. The test results can be known from the p-value at the level of 95% significance. If the p-value is below 0.05, the null hypothesis could be rejected, and the alternative hypothesis is significant. Grubbs' test complies with the concept of probability distribution function as it attempts to generate normally distributed data. This is an essential plus because it provides an excellent opportunity to use the elegant statistics of the normal distribution to make successful forecasts.

An unsupervised machine learning approach was applied using cluster analysis to ensure the complete detection of the outliers. Cluster analysis is a multivariate statistical technique that groups observations based on the features or variables described. The main aim of clustering is to maximize the similarity of observations within a cluster and the dissimilarity between clusters. After applying the clustering technique, the data could be divided into four clusters only, labeled as southeast at average speed, south at low speed, south at high speed, and southwest at average speed. This good result indicates that most of the data wind direction is oriented in the south region of Saad Al Abdullah at three-speed levels. The principal component analysis is performed to

reduce the dimensionality while preserving correlations and relationships inherent in the original structure of the dataset so that different algorithms can learn and be deployed to make accurate predictions.

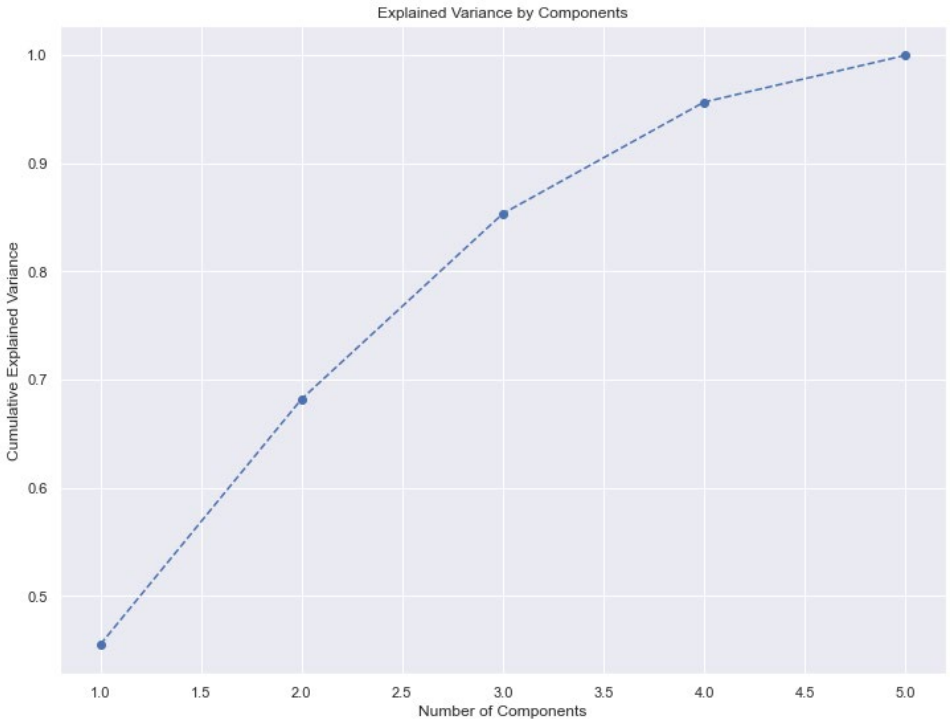


Figure 2.2: PCA explained variance by components

Figure (2) shows that after applying the principal component analysis using K-means, the features can be explained by only 3 components which cover around 85% of the dataset.

Table 2.5: PCA Components with the metrological conditions

Components	Wind direction	Wind speed	Temperature	Solar Radiation	Relative Humidity
Component 1	-0.026315	0.418613	0.564875	0.460766	-0.541
Component 2	-0.847137	0.108313	-0.103955	0.379806	0.339952
Component 3	0.393743	0.709566	-0.395494	0.262689	0.340677

The first principal component positively correlates with temperature, solar radiation, and wind speed. If one increases, then the remaining ones tend to increase as well. Thereby this component primarily measures and indicates a high level of ozone. The second component has negative associations with wind direction and temperature and positive associations with relative humidity, which may indicate a moderate level of SO_2. After applying the principal component analysis, and interpreting the results with the metrological conditions, applying K-means with PCA Clustering will provide a clear perception and insightful meaning about the distribution of the dataset.

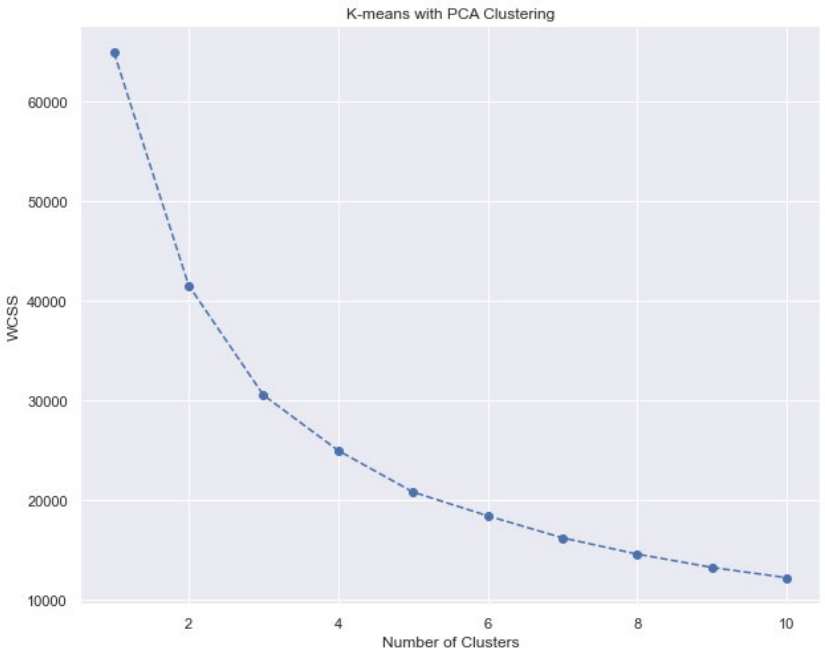


Figure 2.3: K-means with PCA Clustering

Figure (2.3) graphs the eigenvalue against the component number, which shows that the dataset can be segregated into 4 clusters at the "elbow" [0,1,2,3], each segment will represent a different cluster based on the similarities between each record in the dataset. And based on the below table results below, the number of observations and number of proportions, they were renamed to [south-east at average speed, south at low speed, south at high speed, and south-west at average speed].

Table 2.6: Segments of K-means with PCA

Segment K-means PCA	wd	ws	TH	SR	RH	SO2	NO2	O3	THC	Component 1	Component 2	Component 3	N Obs	Prop Obs
West & Low speed	267.167945	1.866283	31.944121	55.253965	22.184665	0.008432	0.034583	0.022351	2.66697	0.077198	-1.018414	-0.467344	4695	0.30876
South West & low speed	225.38828	1.746417	18.2705	63.658305	64.362879	0.005482	0.036142	0.015965	2.751144	-1.665094	0.260023	0.384855	4744	0.311982
South West & High speed	249.625548	4.233077	37.300676	558.475163	17.258406	0.008448	0.013206	0.041593	2.499949	2.020832	-0.059522	0.790242	2736	0.179929
South East & Avg Speed	136.002309	2.368608	32.499291	379.58548	30.049489	0.015363	0.029348	0.036947	2.643523	0.66241	1.224268	-0.591776	3031	0.199329



Figure 2.4: Component 1 vs Component 2 clustered by PCA components



Figure 2.5: Component 1 vs Component 3 clustered by PCA components

The formidable amount of this data has granted a good chance to apply several tools and software packages such as OpenAir in Rstudio for advanced Air Quality analysis and the well-known TensorFlow 2.0 in Python for machine learning and predictive modeling. In addition to constructing a complete descriptive study and understanding several trends, behaviors, and correlations between pollutants and metrological conditions.

2.2.2 Data Transformation

Data transformation is identified as converting data from one format or structure to another to apply computational modeling. Data transformation can be simple or complex based on the input feed and the output data. After applying data cleaning and smoothing to the data, the final step before the modeling part is data transformation to turn the data into an appropriate format for the computer to learn from. Below are the techniques for data transformation:

1. **Normalization:** This process helps to scale the data within a range of (-1.0 to 1.0 or 0.0 to 1.0) to avoid building erroneous machine learning models while training.
2. **Feature engineering selection:** In a vast dataset, data includes many features that will increase classifier operation time. In addition, the prediction accuracy might decrease, especially if there were a weak correlation between the features and the target variable.
3. **Hierarchy Generation concept:** The attributes are converted from level to higher hierarchy to classify the problem.

The normalization technique has been applied in python using the advanced package Scikit-learn to generate a robust model and facilitate the training process. From the same library SK-learn, three approaches have been considered when applying the feature engineering selection.

a. **Dropping constant variance:** Zero variance indicates that the total values are identical.
b. **Feature selection by correlation:** This method is used to drop the highly correlated features that might affect the training process considering the threshold value is 0.7, which is frequently used in statistics and data science.

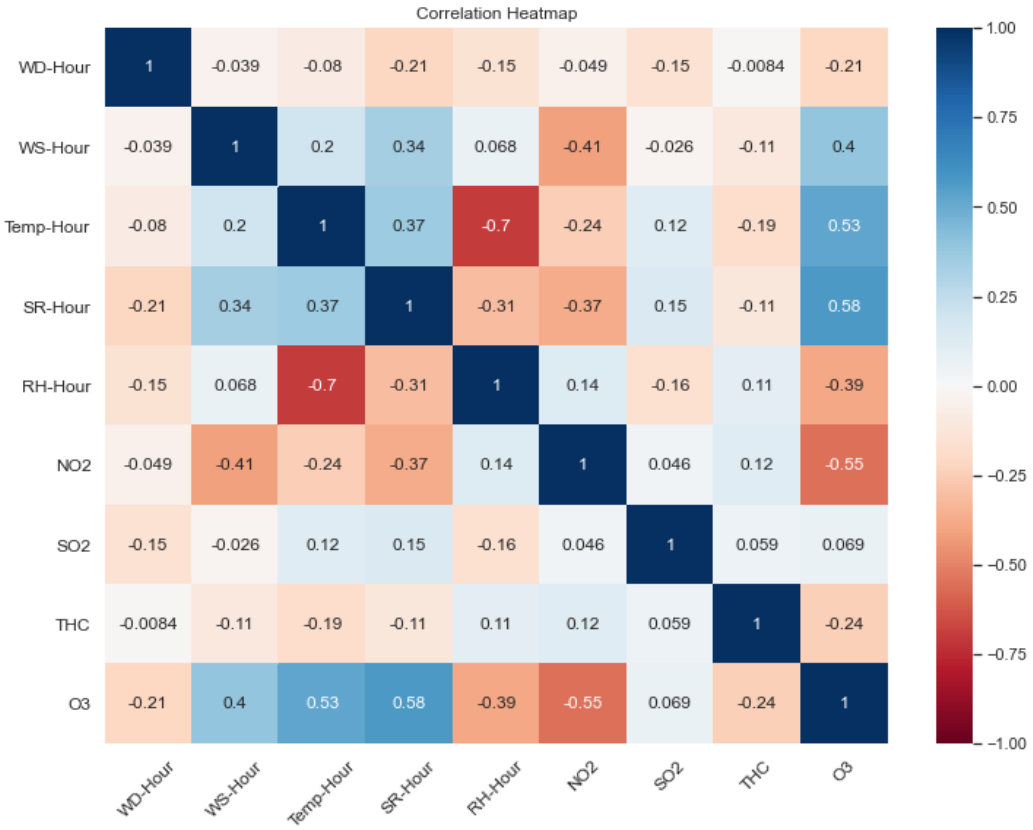


Figure 2.6: Correlation Heatmap

The heatmap correlation shows that the temperature and relative humidity are highly correlated. Thus, it is better to positively study the feature effect by adding one feature, testing the model performance, and repeating the same technique. The value of 0.7 is often considered a benchmark for indicating a strong correlation when interpreting correlation coefficients. In statistics, correlation coefficients range from -1 to 1, with values closer to 1 or -1 indicating a stronger linear relationship between variables. A threshold of 0.7 is commonly used to denote a substantial or strong correlation, though this is more of a guideline than a strict rule. One widely recognized reference that discusses the interpretation of correlation coefficients, including the use of a 0.7 threshold as indicative of a strong correlation, is the book "Psychological Methods of Research" by Coolican, H. (2014). Coolican suggests that correlation coefficients (r) are often interpreted as follows:

a) **0.1 to 0.3 (or -0.1 to -0.3)**: Weak correlation

b) **0.3 to 0.7 (or -0.3 to -0.7)**: Moderate correlation

c) **Above 0.7 (or -0.7)**: Strong correlation

This framework is a general guideline and is widely used in various fields, including psychology, education, and social sciences, to interpret the strength of linear relationships between variables. Hence, we can conclude a strong negative correlation between the hourly temperature and the humidity reaches -0.7, and a moderate negative correlation between NO2 and O3. Also we can notice a moderate positive correlation between O3 and wind speed and a moderate negative correlation between NO2 and the wind speed.

c. **Feature Selection by Information gain:** This method is widely used which can share the mutual information between the inputs and the output to calculate the weight of the feature and to determine if the feature is significant or not.

This occurs by applying the F_Regression from Scikit-learn to identify a subset of input variables almost relevant to the target variable. Then by adding the function SelectKBest, it simply retains the features of independent variables with high scores.

As a result of the three applied tests of the feature engineering selection, and after understanding the input's importance step by step, the five significant features that are included in the model were the Nitrogen dioxide, wind speed, Solar Intensity Radiation, relative humidity and temperature which are shown in the below figure. These techniques were performed in python using the innovative library SK-learn, a very efficient tool for predictive data analytics. These libraries can save time and effort to build a complete comprehension and outcome with marvelous results. In the analysis conducted for my thesis, the selection of the five significant features – Nitrogen dioxide (NO2), wind speed, Solar Intensity Radiation, relative humidity, and temperature – was primarily data-driven, utilizing machine learning techniques through the SK-learn library in Python. However, it's important to understand the physical implications and relevance of these features in the context of air quality and environmental studies.

1. **Nitrogen Dioxide (NO2):** NO2 is a primary pollutant mainly emitted from fuel combustion processes such as traffic, industrial activities, and power plants. High concentrations of NO2 are associated with adverse health effects and also contribute to the formation of ground-level ozone and particulate matter.
2. **Wind Speed:** Wind speed plays a crucial role in the dispersion and dilution of air pollutants. Higher wind speeds generally facilitate the dispersion of pollutants, reducing

their concentration in a specific area, while low wind speeds can lead to pollutant accumulation.

3. **Solar Intensity Radiation:** Solar radiation affects the photochemical reactions in the atmosphere, particularly the formation of ozone. Strong solar radiation can increase the rate of photochemical smog formation, which includes pollutants like ozone and secondary particulate matter.
4. **Relative Humidity:** Humidity can influence the concentration and chemical composition of airborne pollutants. It affects the formation of secondary aerosols and can also lead to the hygroscopic growth of particulate matter, impacting air quality and visibility.
5. **Temperature:** Temperature can affect both the chemical reactions in the atmosphere and the behavior of pollutants. For example, higher temperatures can enhance the formation of ground-level ozone and alter the chemical transformation of pollutants.

These features were not arbitrarily chosen; rather, they are known to have direct or indirect impacts on air quality. The data analysis approach provided a quantitative assessment of their significance in the context of the studied environment. However, the physical interpretation aligns with established environmental science principles, indicating that these variables are critical in understanding and predicting air quality patterns.

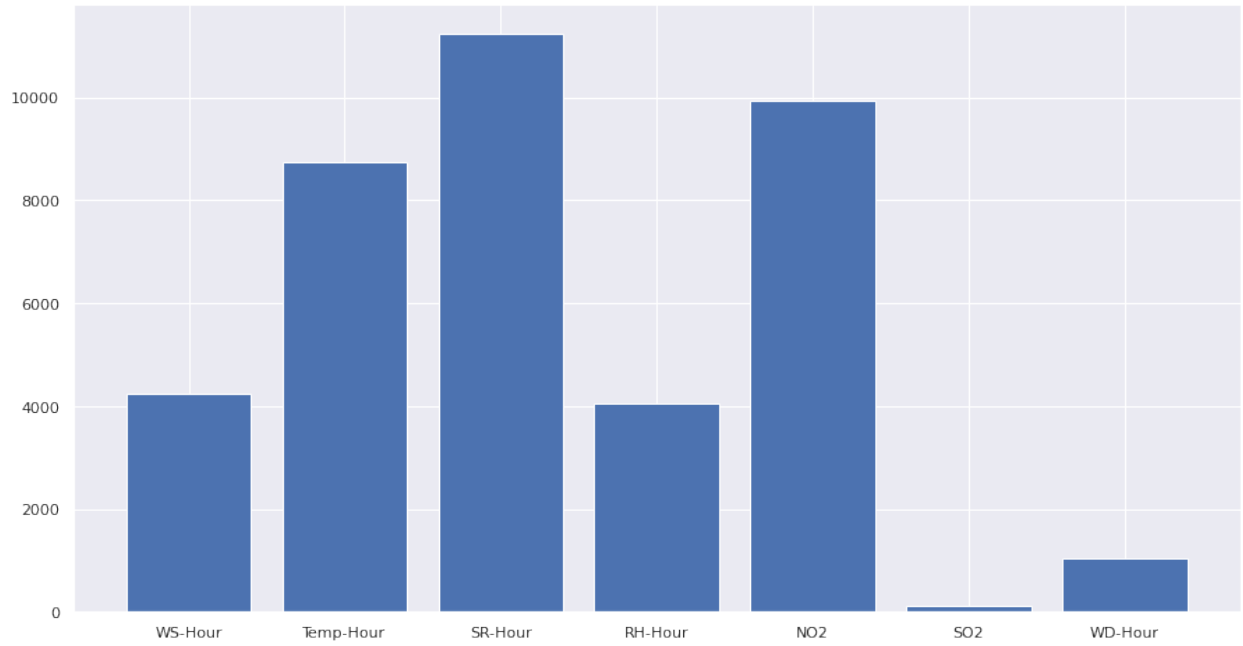


Figure 2.7: Feature engineering selection using F_regression

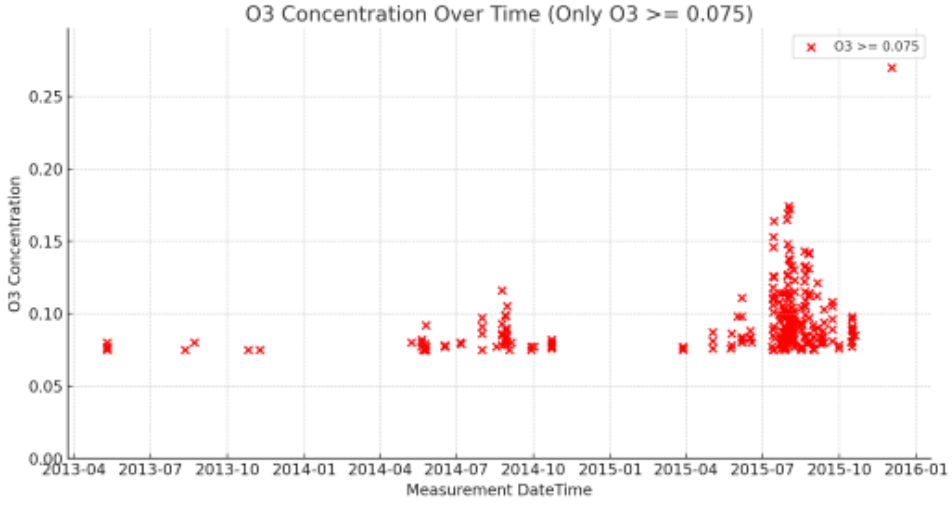
On the other hand, there are different methods that can be used in feature selection that is more efficient and is widely used in real-life problems but time-consuming such as:

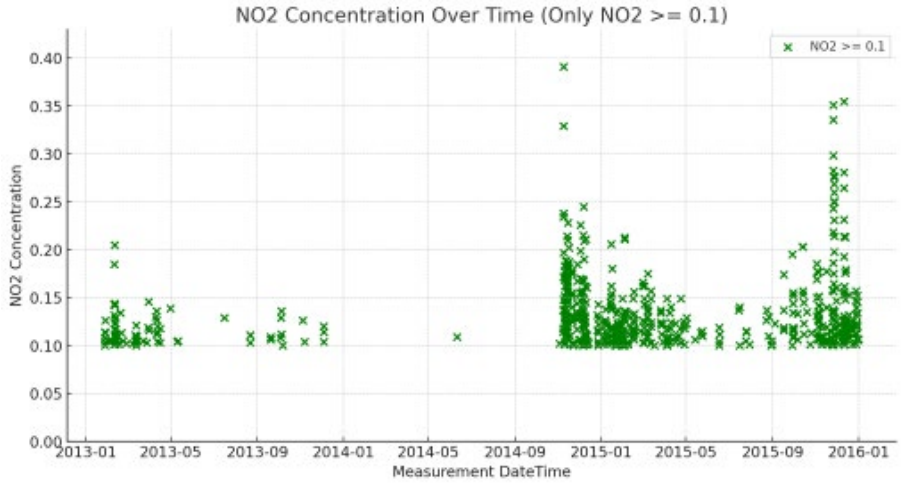
- **Forward Selection:** It is an iterative technique in which the user starts with having no feature in the model and keeps adding the feature which best improves the model till the addition of a new feature does not improve the performance.
- **Backward Elimination:** It is the opposite of the forward selection. The user starts will all the features and removes the least significant feature at each iteration which improves the model performance. The user keeps repeating this process until no improvement is observed on the removal of the features.

National Ambient Air Quality Standards in the State of Kuwait*

Pollutant Name	Symbol	Duration of Exposure	National Limit	Unit of Concentration
Sulphur Dioxide	SO_2	1 Hour	0.075	Parts per million (ppm)
		24 Hours	0.019	Parts per million (ppm)
Nitrogen Dioxide	NO_2	1 Hour	0.100	Parts per million (ppm)
		Annual	0.021	Parts per million (ppm)
Carbon Monoxide	CO	1 Hour	35.00	Parts per million (ppm)
Ozone	O_3	8 Hour	0.075	Parts per million (ppm)
Particulate Matter 10 Micron	PM_{10}	24 Hours	350	Micrograms per cubic meter ($\mu g/m^3$)
Particulate Matter 2.5 Micron	$PM_{2.5}$	24 Hours	75	Micrograms per cubic meter ($\mu g/m^3$)

Figure 2.8: KEPA Standards





Year	NO2 Exceedance Count	O3 Exceedance Count	NO2 % of Total	O3 % of Total
2013	73	8	1.33%	0.15%
2014	153	49	1.78%	0.57%
2015	365	203	4.23%	2.41%

Data Analysis

Data analysis is the process of examining and evaluating data to extract insights and information. It is a crucial step in the data science workflow, as it allows you to explore and understand the data, identify patterns and trends, and make informed decisions based on the data. Data analysis involves various techniques and methods, including visualizing the data, applying statistical tests, and using machine learning algorithms to build predictive models. It is a multi-disciplinary field that combines elements of computer science, statistics, and domain expertise to extract insights from data. Data analysis has many applications, including predicting customer behavior, identifying trends in financial markets, detecting fraudulent activity, and optimizing business processes. It is an essential tool for businesses and organizations that want to make data-driven decisions and improve their operations. Data analysis is a crucial component of data science and is vital in extracting value from data. It allows businesses and organizations to gain a deeper understanding of their data and make more informed decisions based on that knowledge. As mentioned before, for air quality analysis, openair package will apply in Rstudio programming language. The openair package is an R package for analyzing and visualizing air quality data. It was developed by a team of researchers and software developers at the University of York, UK, with the goal of providing a comprehensive toolkit for analyzing and interpreting air pollution data. The openair package was designed specifically for use in academic research and has become a popular tool among

researchers studying air quality and its impacts on human health and the environment. The package is particularly useful for handling large and complex air quality datasets, and provides a range of functions for data import, cleaning, aggregation, and visualization. One of the key features of the openair package is its ability to handle data from a wide range of air quality monitoring networks and sensors, including data from fixed monitoring stations, mobile monitoring units, and personal exposure monitors. The package also includes functions for performing common statistical analyses and modeling techniques, such as linear regression, generalized linear models, and mixed-effects models. In addition to its analytical capabilities, the openair package also includes several visualization functions for creating maps, plots, and other graphical representations of air quality data. These functions allow users to easily explore and communicate the patterns and trends present in their data and can be used to create publication-quality figures for academic papers and reports. Overall, the openair package is a valuable resource for researchers studying air quality and its impacts, and provides a wide range of tools for analyzing, visualizing, and interpreting air pollution data. This thesis will focus on the behavior of the NO_2 and O_3 and what kind of relations and mutual patterns they might have. For instance, in 2014 and 2015, NO_2 concentrations for winter season were consistently higher than summer values and it was noticed that the maximum value recorded was 0.153 ppm on 08-12-2014 at 20:00 hour evening, in contrast the O3 value has recorded the minimum value for 0.001 ppm.

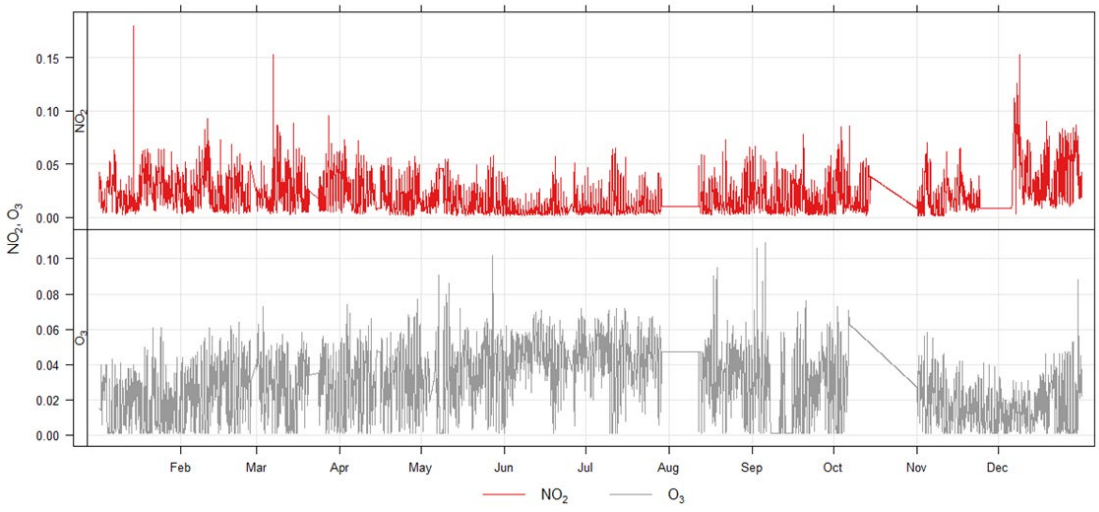


Figure 2.9:NO_2 vs O_3 trend in 2014

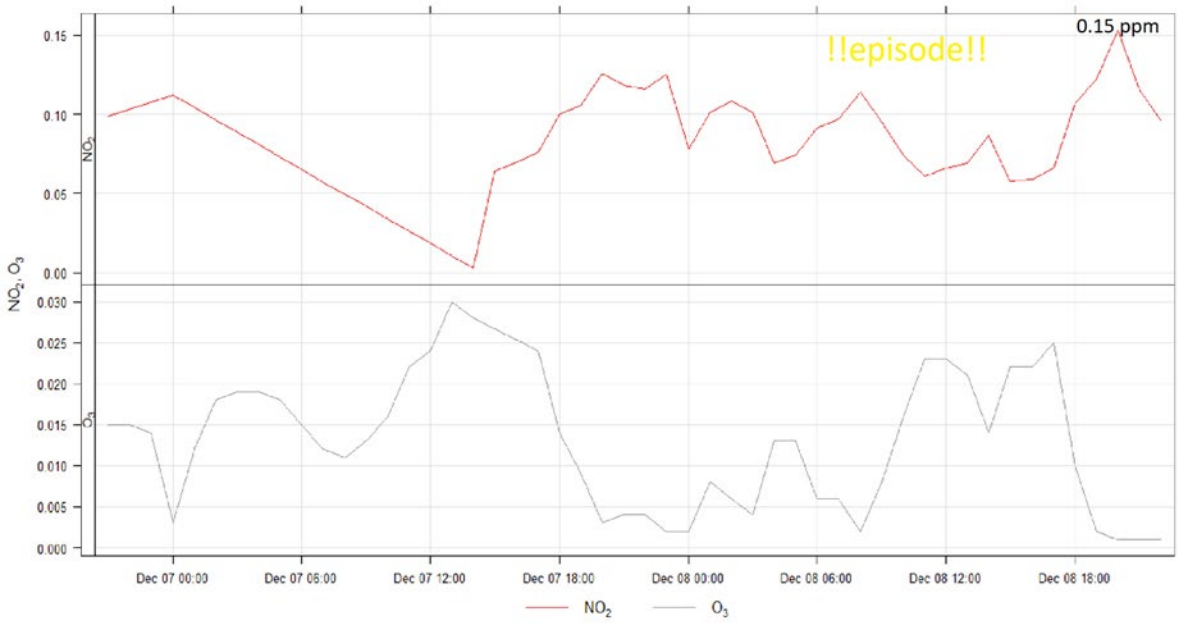


Figure 2.10:NO_2 episodes vs O_3 values in 2014

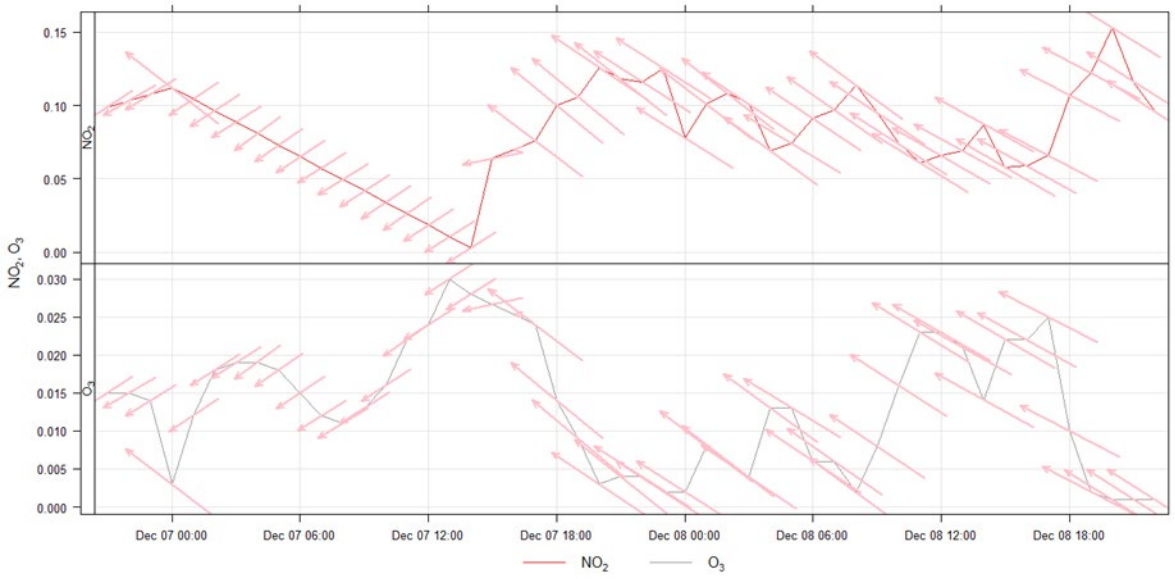


Figure 2.10:wind direction during episodes of NO_2

From figure 2.10 we can notice a considerable effect from the wind-flow source from the Southeast and east to that it may cause episodes of NO_2 pollutant exceeding the KEPA limitations.

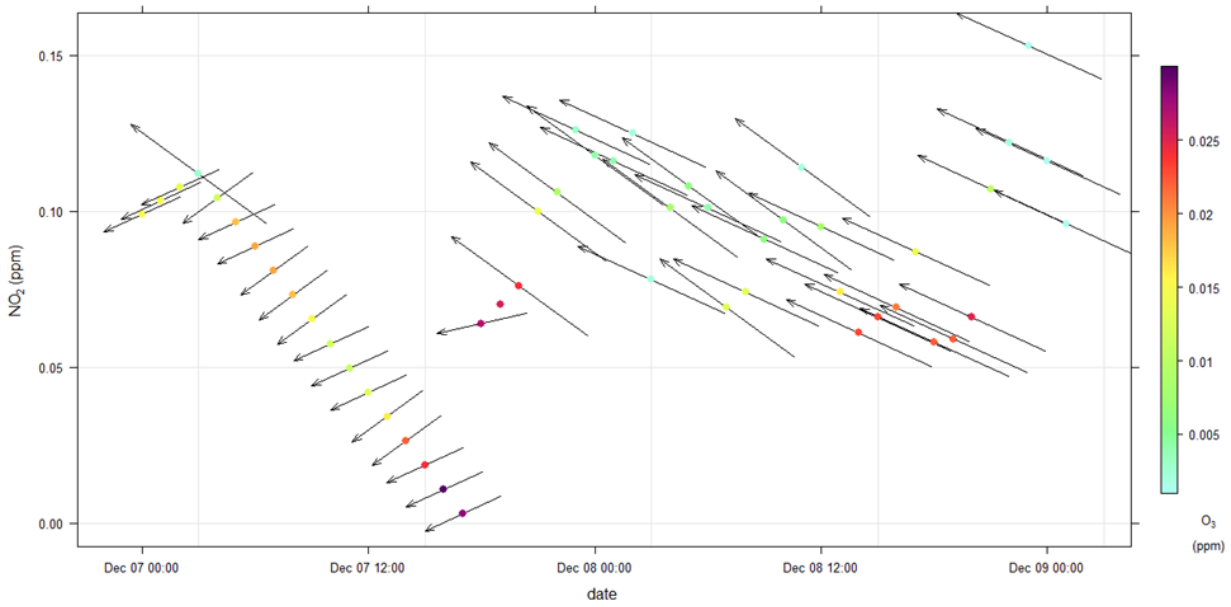


Figure 2.11: wind direction and episodes concentrations of NO_2

We can clearly notice from figure 2.11 that at low concentration of NO_2, the O_3 is increased.

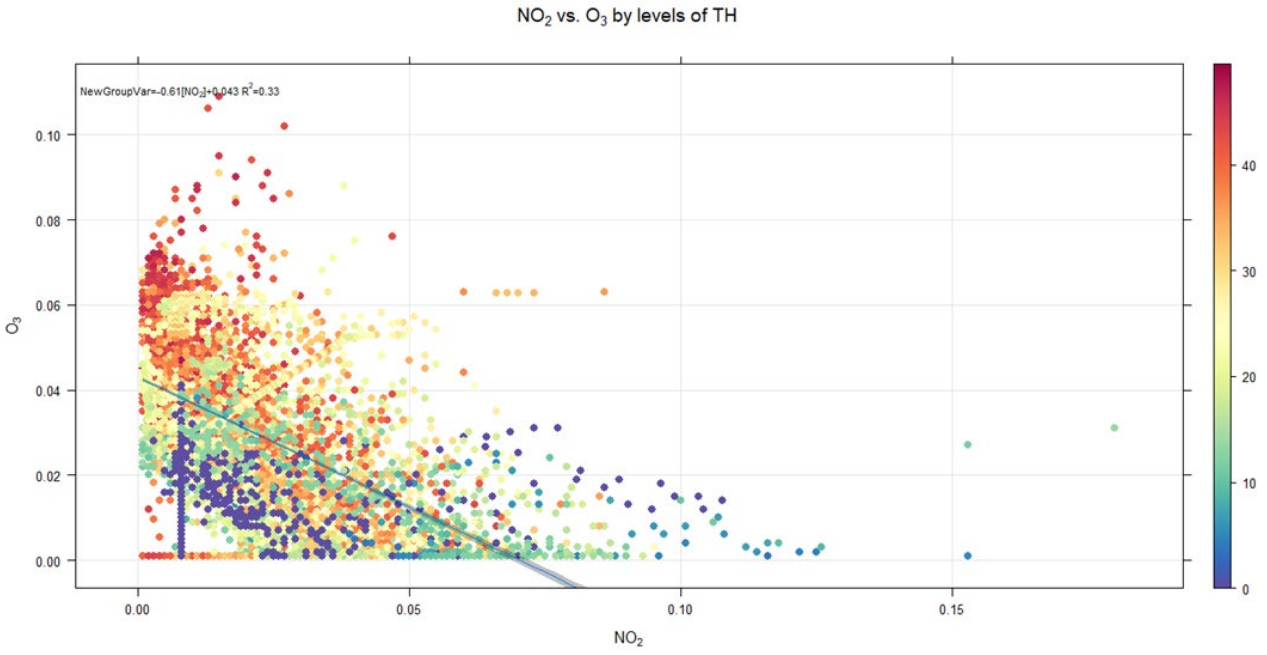


Figure 2.12: NO_2 vs O_3 relationship

From figure 2.12 there is a strong negative relationship between NO_2 and O_3. Higher values of O_3 are caused with higher values of temperature, in contrast lower values of temperature caused with low values of NO_2.

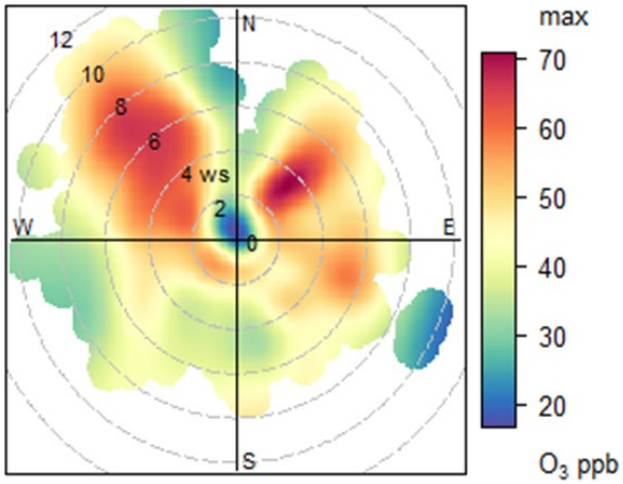


Figure 2.13: O_3 Polar plot

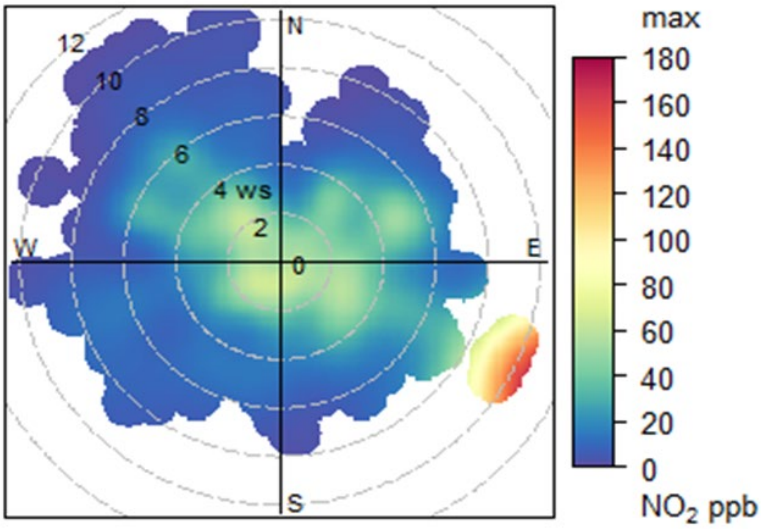


Figure 2.14: NO_2 Polar plot

A polar plot is a type of graph that is used to display data in a circular layout, with the angles of the circular layout representing the different categories or variables of the data and the radius of the plot representing the magnitude or value of the data. One common use of a polar plot is to compare the levels of two air pollutants, such as ozone (O_3) and nitrogen dioxide (NO_2), at different locations or over time. By plotting the levels of O_3 and NO_2 on a polar plot, it is possible to visualize the relationship between the two pollutants and identify any patterns or trends in their concentrations. From figure 2.13 and 2.14, the polar plot for all data shows that the highest O_3

concentrations tend to occur for high wind speed conditions from almost every direction. Lower concentrations are observed for low wind speeds because concentrations of NO_2 are higher, resulting in O_3 destruction. Another tool in open air that can dig deeper is the polar frequency method. A polar frequency plot is a type of graph that is used to visualize the distribution or frequency of data within a circular layout. Like a regular polar plot, a polar frequency plot uses the angles of the circular layout to represent different categories or variables, and the radius of the plot to represent the magnitude or value of the data. One common use of a polar frequency plot is to compare the frequency or occurrence of two air pollutants, such as ozone (O3) and nitrogen dioxide (NO2), at different locations or over time. By plotting the frequency of O3 and NO2 on a polar frequency plot, it is possible to visualize the relative prevalence of the two pollutants and identify any patterns or trends in their occurrence.

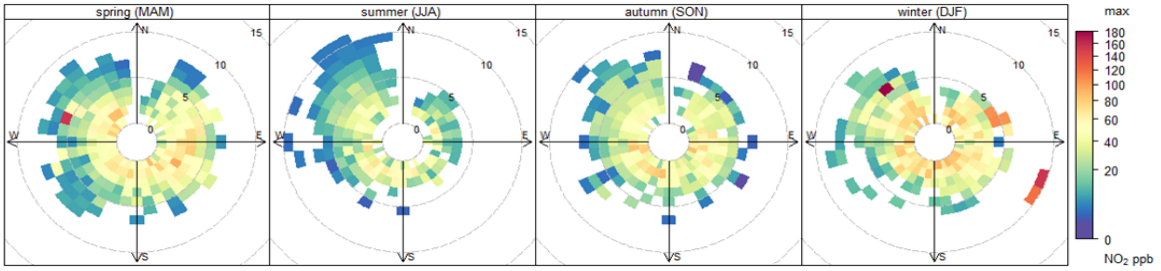


Figure 2.15: NO_2 Polar frequency

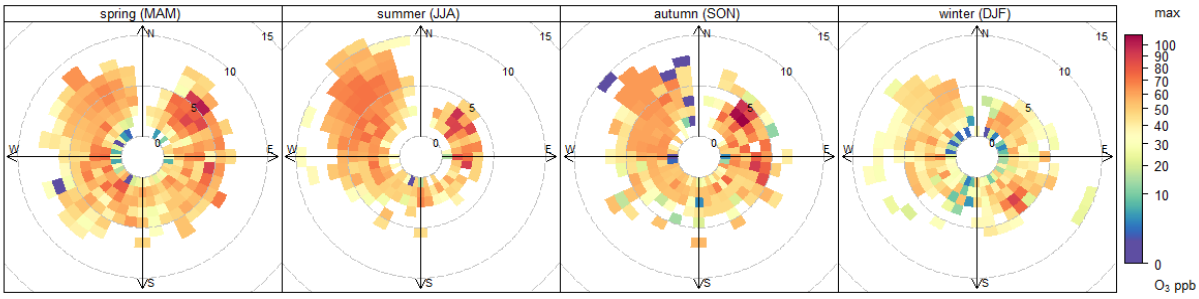


Figure 2.16: O_3 Polar frequency

The polar frequency plot of O_3 and NO_2 shows that the frequency of O_3 tends to be higher in the summer months and lower in the winter months, while the frequency of NO_2 tends to be more stable throughout the year. Also, we can notice that at high frequency and episode of NO_2, the O_3 concentrations went to minimum values. A good comparison also might be added is to compare the time of episode not only the seasons and months but also the daylight and nighttime so that we can extract and learn the highest episodes time. For this task we might use percentile-

rose, A percentile rose plot is a type of graph that is used to visualize the distribution or frequency of data within a circular layout, similar to a polar frequency plot. However, instead of showing the raw frequency or occurrence of a variable, a percentile rose plot shows the percentile distribution of the data. In the context of air quality, a percentile rose plot can be used to compare the distribution of two pollutants, such as ozone (O_3) and nitrogen dioxide (NO_2), at different locations or over time. By plotting the percentile distribution of O_3 and NO_2 on a percentile rose plot, it is possible to visualize the relative prevalence of the two pollutants and identify any patterns or trends in their distribution.

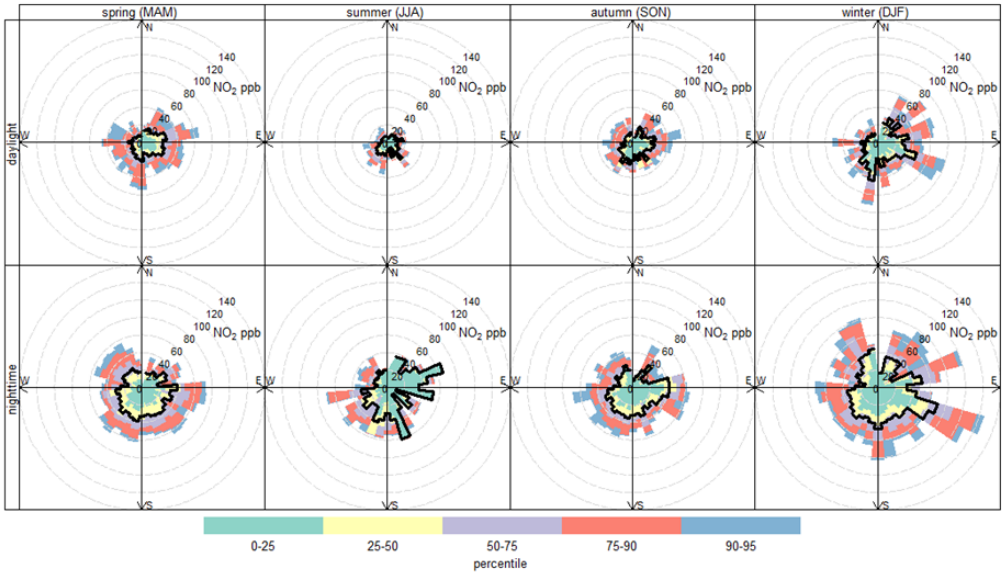


Figure 2.17: percentile rose for NO_2 daylight and nighttime frequency

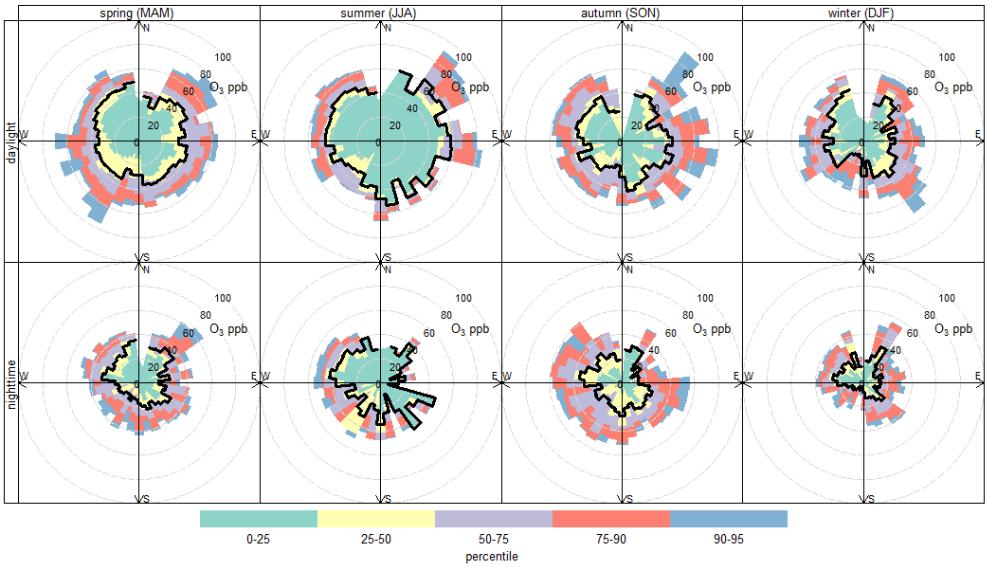


Figure 2.18: percentile rose O_3 daylight and nighttime frequency

From figure 2.18 and 2.19, O_3 is highly active in the daylight as we expected due to the positive correlation with Temperature. (exceeding 70 ppb wasn't found at night). NO_2 is highly active at nighttime as we expected due to the negative correlation with Temperature and positive correlation with Relative humidity. To create a percentile rose plot using the openair package in R, you can use the percentileRose function, which takes as input a data frame containing the pollutants of interest and any relevant metadata (e.g. location, date). The percentileRose function will then generate a percentile rose plot based on the data provided. One of the most beautiful figures in openair is that can generate a calendar plot. The calendarPlot function is a feature of the openair package in R that allows users to create a graphical representation of the distribution of a variable over time, using a calendar layout. The calendarPlot function is particularly useful for visualizing data that is collected on a daily or weekly basis, as it allows users to easily see the patterns and trends in the data over the course of a year or other time period.

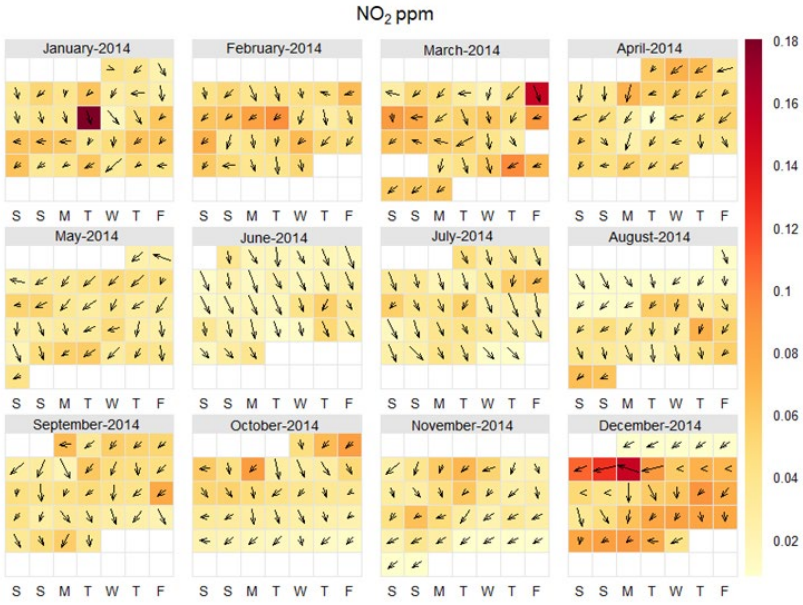


Figure 2.19: Calendar plot with wind direction for NO_2

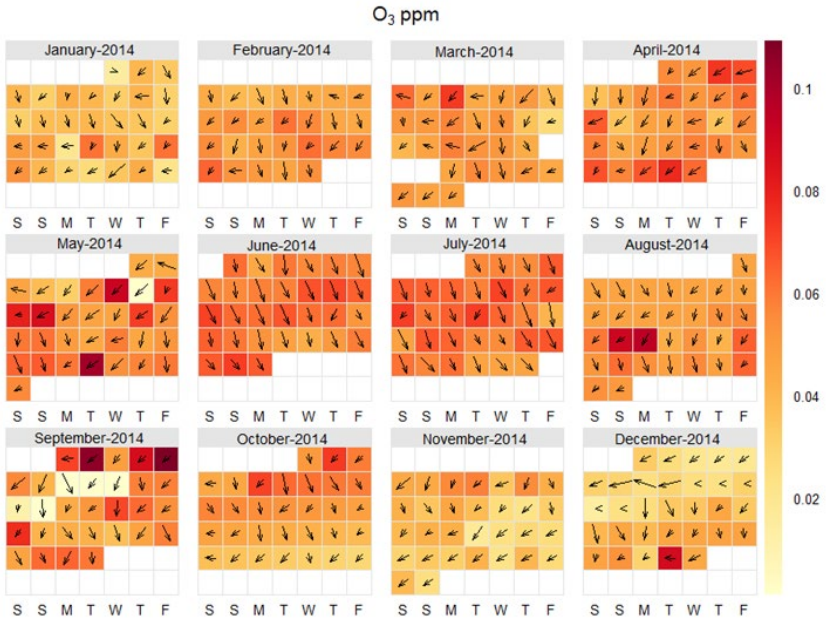


Figure 2.20: Calendar plot with wind direction for O_3

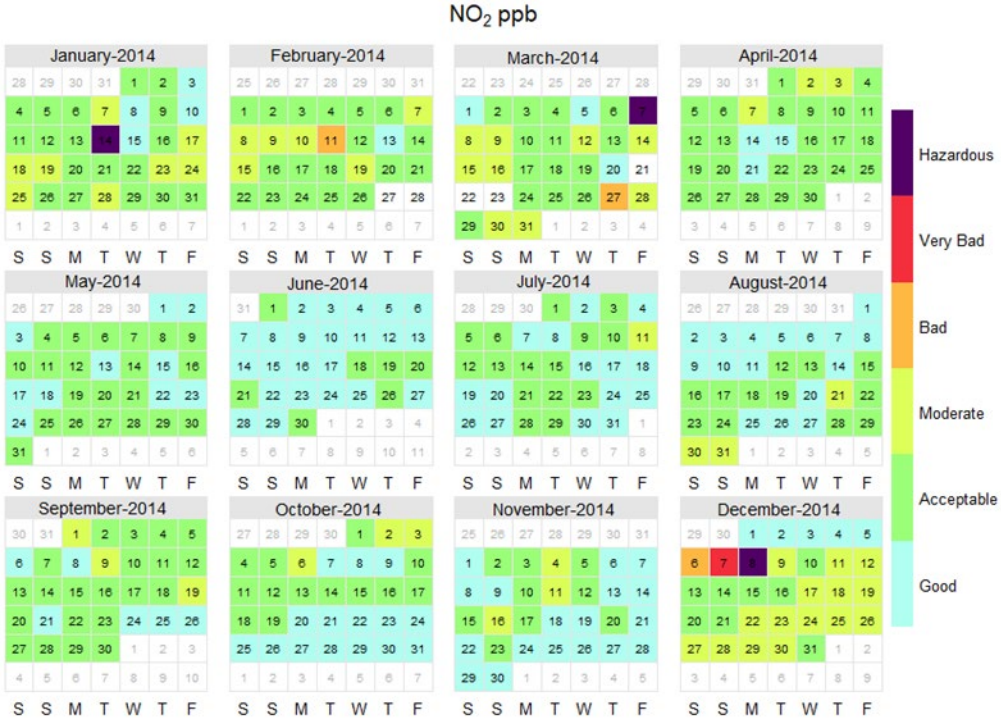


Figure 2.21: Calendar plot classes for NO_2

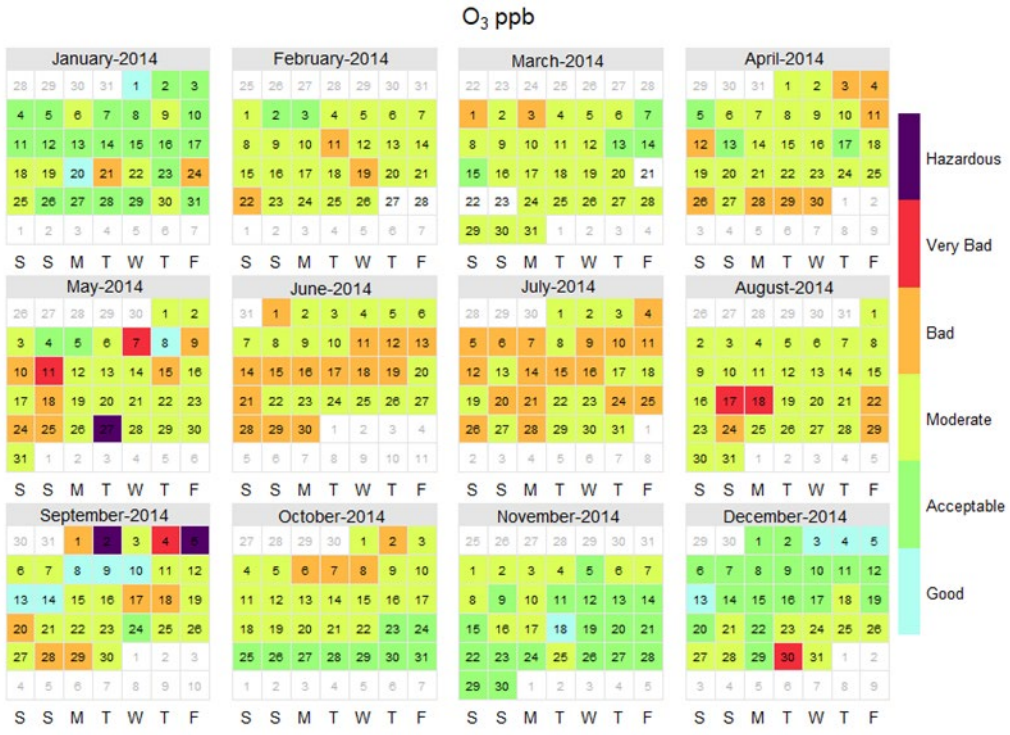


Figure 2.22: Calendar plot with classes for O_3

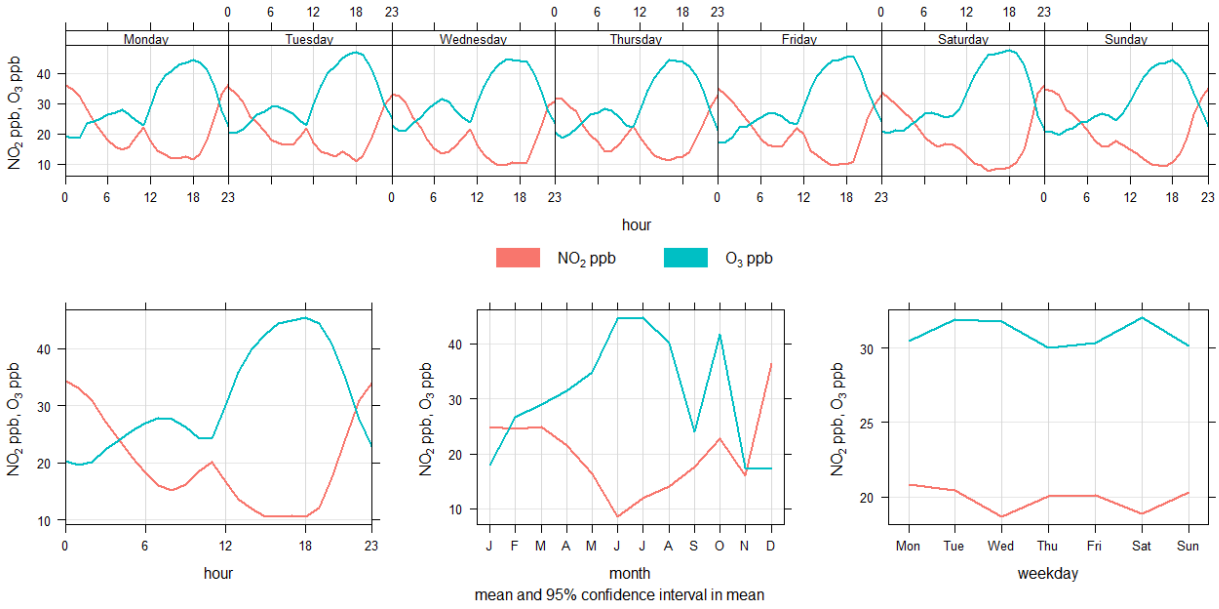
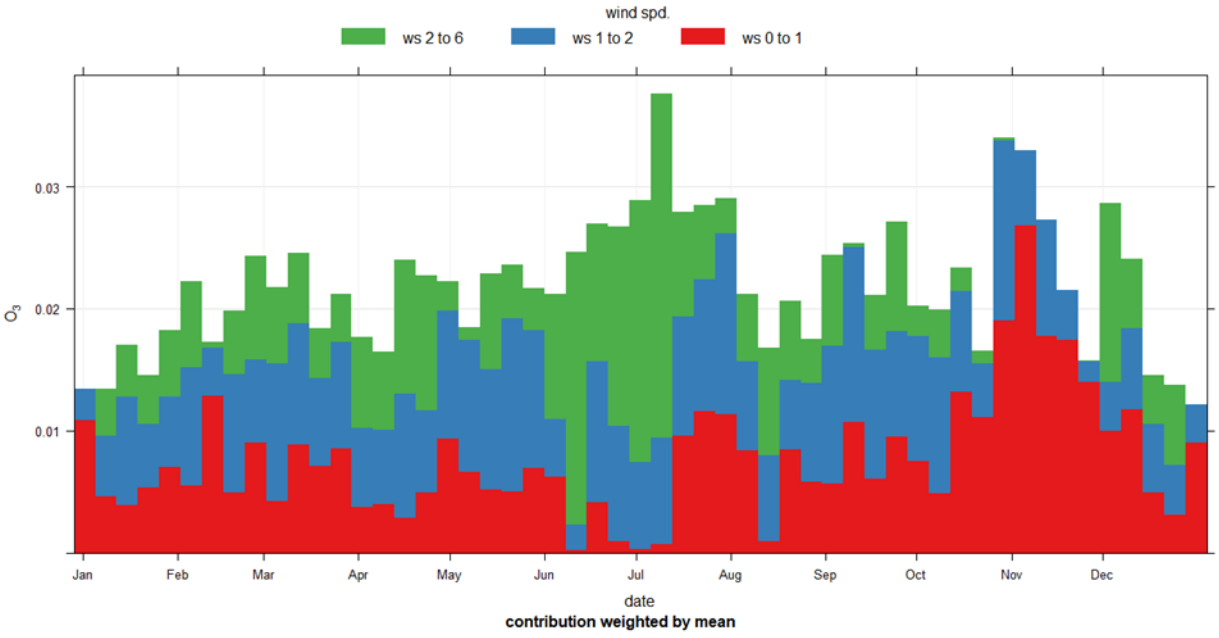
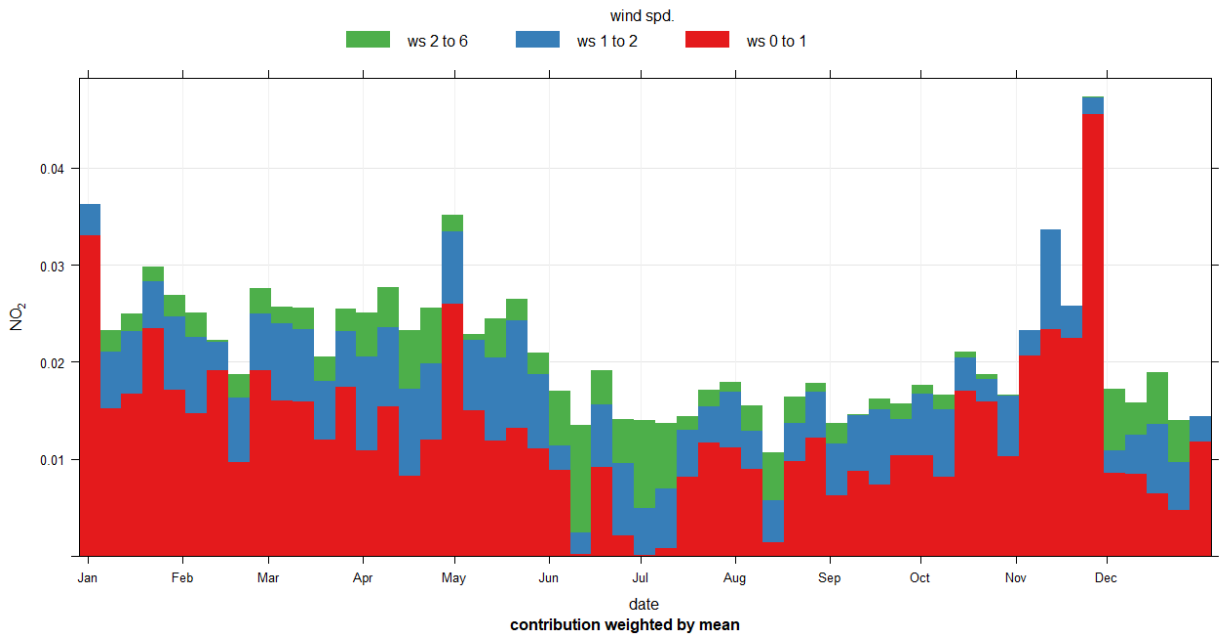


Figure 2.22: Time variation for No2 and O3 analysis

The timeVariation function in the Openair library is a powerful tool for analyzing temporal variations in air quality data. This function allows users to explore how air pollutant concentrations change over time, aiding in the identification of patterns, trends, and potential correlations. When

examining nitrogen dioxide (NO2) and ozone (O3) concentrations, a negative correlation suggests an inverse relationship between the two pollutants. In other words, as NO2 levels increase, O3 levels tend to decrease, and vice versa. This negative correlation could be indicative of competing chemical reactions in the atmosphere, as NO2 and O3 are often involved in complex photochemical processes. The timeVariation function can be instrumental in visually representing these correlations through figures, providing a comprehensive overview of the temporal dynamics and interactions between different air pollutants. Analyzing such correlations is crucial for understanding the complex interplay of atmospheric constituents and informing strategies for air quality management. Also studying the relationship between wind speed and the NO2 and O3 is a good practice to determine better insights. As mentioned earlier that the correlation between NO2 and wind speed is moderately negative and the correlation between O3 and wind speed is moderately positive, in the below figure we can assure that most of the avg high speed values occurs at the high level of O3 and vice versa. On the other hand, for NO2, we can notice that at the region of low-speed values the concentration of NO2 tends to increase and vice verca.





1. **Correlation between NO2 and Wind Speed (-0.4):**
    - *Physical Meaning:* A correlation of -0.4 between nitrogen dioxide (NO2) and wind speed suggests a moderate negative relationship. In practical terms, as wind speed increases, there is a tendency for NO2 concentrations to decrease, and vice versa. This negative correlation could be explained by the fact that higher wind speeds enhance the dispersion and dilution of pollutants, leading to lower concentrations at a specific location. It implies that atmospheric mixing or transport processes associated with higher wind speeds contribute to a reduction in local NO2 levels.
2. **Correlation between O3 and Wind Speed (0.4):**
    - *Physical Meaning:* A correlation of 0.4 between ozone (O3) and wind speed indicates a moderate positive relationship. As wind speed increases, there is a tendency for O3 concentrations to increase, and vice versa. This positive correlation might be influenced by factors such as increased atmospheric mixing, which can lead to higher O3 levels due to enhanced photochemical reactions. It's important to note that the interpretation of O3 and wind speed correlations can be complex and context-dependent, as various factors, including precursor emissions and atmospheric conditions, can influence ozone concentrations.

Choosing a model

This part is the most important one in our workflow. Many models have been created over the years by developers and researchers. The selection of an applicable model depends on the type of data consisting of image, text, sound, and categorical or tabular data. In this thesis, the data is comprised of massive historical data with a measurement date-time, metrological conditions, and the concentrations of the pollutants. This represents tabular numeric data that can be treated as a regression problem. The purpose is to minimize the errors between the actual observed data and the predicted data for different models. The substantial target of this thesis is to apply the sensitivity analysis between several models to discover the robust model that can minimize the loss functions, gives the best evaluation metrics, and performs well in any out-of-sample data. It is also noteworthy that the regression problem can be transformed into a classification problem. The reason behind that is to identify whether pollution and episodes will occur.

2.4.1 Regression Loss function

1. Mean Absolute Error (MAE): Defined as the sum of the absolute differences between the target and the predicted variables outcome from the model.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_i{}^p|}{n}$$

2. Mean Absolute Percentage Error (MAPE): Defined as the sum of the absolute percentage errors between the target and the predicted variables outcome from the model.

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - y_i{}^p}{y_i} \right| \times 100$$

3. Mean Squared Error (MSE): Defined as the sum of the squared distances between the target and the predicted variables outcome from the model.

$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_i{}^p)^2}{n}$$

4. Root Mean Squared Error (RMSE): Is just an extension of MSE and defined as the square root of the Mean of the Square of Errors.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - y_i{}^p)^2}{n}}$$

5. R^2 score : Defined as a statistical measure that represents the goodness of the fit and the variability of data explained by the model in a regression problem ranged from zero to one by evaluating the proportion of the variance.

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

SSR: $Sum\ of\ the\ differences\ between\ the\ predicted\ value\ and\ the\ mean\ of\ the\ dependent\ variable$

SST: $Squared\ differences\ between\ the$ observed dependent variable and its mean.

SSE: $Squared\ error\ differences$ between the observed value and the predicted value.

All these metrics are used to evaluate the model performance accurately. Thus, it is good to mention that these loss functions are determined using the L2-norm. The L2-norm or Frobenius norm of a vector (Euclidean length) is more suitable in predicting air pollution as it will always predict one stable solution. Besides, the primary rationale is that the L2-norm loss is the distance from the origin (0) So, the closer to the origin is the difference of the outputs and the targets, the lower the loss and the better the prediction.

2.4.2 Linear Regression Predictive Model

Linear regression is the most fundamental machine learning method. Linear regression is a linear approximation of a causal relationship between two or more variables. The linear regression model always consists of a predictor variable and an observed variable related linearly to each other.

Linear regression model: $Y = \beta_0 + \beta_1 X_1 + \varepsilon_1$

The predictor error is known as the difference between the observed values and the predicted value. The beta coefficients are regression weights, and they are the association between the predictor variable and the outcome. They represent the change in y related to a one-unit change in X_1 and X_2, respectively. The best fit line is obtained by varying the values of $\beta_0\ and\ \beta_1$. They are selected in such a way that it gives the minimum predictor error. It is important to note that a simple linear regression model is susceptible to outliers. Therefore, it should not be used in big-size data.

2.4.2.1 Multiple linear regression

It is an extension of linear regression; it enables analysts to determine the variation of the model and each independent variable's relative contribution.

And here is a multiple regression model with three predictor variables (x) predicting variable y

$$Y = \beta_0 + \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3$$

It is the same as the linear regression model with more than one independent variable.

2.4.2.2 Assumptions of Multiple Linear Regression

1. Relationship Between Dependent and Independent Variables

   The dependent variable relates linearly with each independent variable.

2. The Independent Variables Are Not Much Correlated

   The data should not display multicollinearity, which happens in case the independent variables are highly correlated to each other.

3. The Residual Variance is Constant

   Multiple linear regression assumes that the remaining variables' error is similar at each point of the linear model.

4. Observation Independence

   The observations should be of each other, and the residual values should be independent

5. Multivariate Normality

   Multivariate normality happens with normally distributed residuals.

2.4.2.3 Least square method

Although several techniques may solve the regression problem, the most-used method is the least square method. In the least-squares regression analysis, the b's are selected to minimize the sum of the squared residuals.

$$Sr = d1^2 + d2^2 + d3^2 + d4^2 + d5^2 + d6^2 + d7^2 + d8^2 + d9^2$$

Least Squares Criterion:

$$\min \sum (y_i - \hat{y}_i)^2$$

For simple linear regression $p = 1$

$$\hat{y}_i = b_0 + b_1 x_{1i}$$

To determine values for b_0 and b_1, the equation is differentiated with respect to each coefficient:

This yield:

$$\frac{\partial \mathrm{Sr}}{\partial \mathrm{b}_0} = -2 \sum (y_i - b_0 + b_1 x_{1i}) = 0$$

$$\frac{\partial \mathrm{Sr}}{\partial \mathrm{b}_1} = -2 \sum [(y_i - b_0 + b_1 x_i) x_{1i}] = 0$$

By rearranging will get the normal equations:

$$\begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix} \times \begin{pmatrix} b_o \\ b_1 \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix}$$

By solving simultaneously:

$$b_1 = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}$$

$$b_0 = \frac{\sum y_i - (b_1 \cdot \sum x_i)}{n}$$

For any regression: $p = N$

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_N x_{Ni}$$

To determine values for b, the equation is differentiated with respect to each coefficient:

This yield:

$$\frac{\partial \mathrm{Sr}}{\partial \mathrm{b}_N} = -2 \sum x_{Ni} (y_i - b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_N x_{Ni}) = 0$$

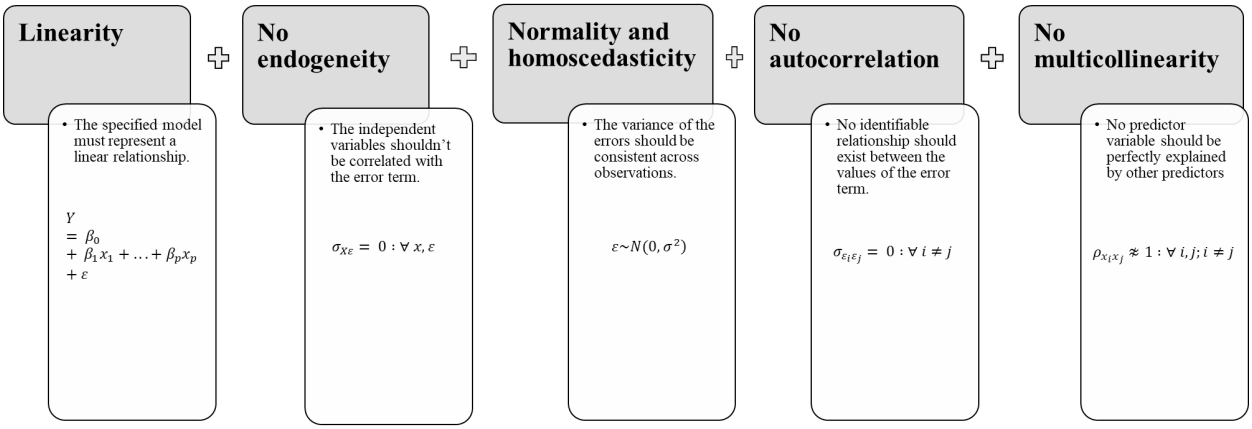
The normal equations:

$$\begin{pmatrix} N & \sum x_{1i} & \sum x_{2i} & \cdots & \sum x_{Ni} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}\,x_{2i} & \cdots & \sum x_{1i}\,x_{Ni} \\ \sum x_{2i} & \sum x_{1i}\,x_{2i} & \sum x_{2i}^2 & \cdots & \sum x_{2i}x_{Ni} \\ \vdots & \vdots & & \vdots & \\ \sum x_{Ni} & \sum x_{1i}\,x_{Ni} & \sum x_{2i}x_{Ni} & \cdots & \sum x_{Ni}^N \end{pmatrix} \times \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_{1i}\,y_i \\ \sum x_{2i}\,y_i \\ \vdots \\ \sum x_{Ni}\,y_i \end{pmatrix}$$

By solving simultaneously:

will get the value for each b_N

The following assumptions must be considered when using multiple regression



2.4.3 The K-nearest neighbors (KNN) algorithm

The K- Nearest Neighbor is one of the simplest supervised machine learning algorithms. It can be used to solve many regression and classification problems. It assumes that similarity exists between different points in a close neighborhood. Hence, it captures similarity, distances (Euclidean distance), or closeness with basic mathematics by calculating the distance between points on a graph. The most significant parameter that should be initialized is K to choose the number of neighbors. Selecting the optimum value of K is called hyperparameter tuning, which will be discussed in the next section. But it is crucial to mention that after running the KNN algorithm several times with different values of K., it may not be able to accurately make predictions because the algorithm might be unprovided with a correct value of K. Thus, predictions become less stable. However, several methods of calculating the distances between points, like

Euclidean distance and Manhattan Distance. Euclidean distance gives better predictions. Therefore, it will be the primary method in this study.

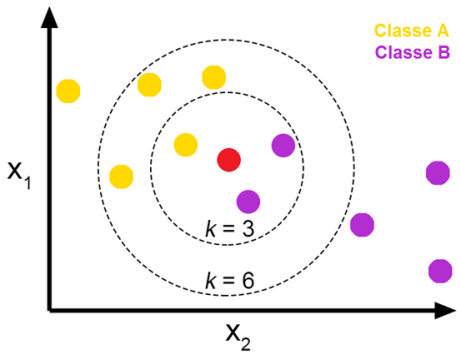


Figure 2.23: Example of KNN Algorithm concept

Euclidean Distance functions: $d(x_i,y_i) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_k - y_k)^2}$

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

2.4.4 Ensemble Techniques

The ensemble method or ensemble learning is a machine learning technique that combines multiple base models to solve a particular computational problem to produce one optimal predictive model. The combinations of predictions from several models built will improve the robustness of a single model. Ensemble techniques utilize the concept of Decision Trees to outline the problem's definition and provide a practical solution to the problem. Ensemble techniques can be split into two groups:

1. **Bagging:** In the bagging technique, the data is distributed with decision trees from the base learners parallel to several sub-learners, creating a net of bootstrapping. In bootstrapping, a sample is chosen from a set using the replacement method, making the selection completely random. After that, the results of each tree are aggregated to yield the robust and most accurate predictor. One critical model that relies on the bagging technique is the Random Forest regression. Random Forest is one of the most powerful and standard supervised machine learning algorithms. It adds additional randomness to the model by growing the trees for learning. It also works smartly by searching for the best feature among a random subset of features. This yield results of a wide diversity that generally determines a better model. Random Forest is most occasionally exposed to the overfitting phenomena because it results in a high variance after bagging and yields a high train error. Thus, it is always

essential to provide the algorithm with the optimal parameters to avoid overfitting by controlling the max depths of the trees and the number of the learner's estimators.

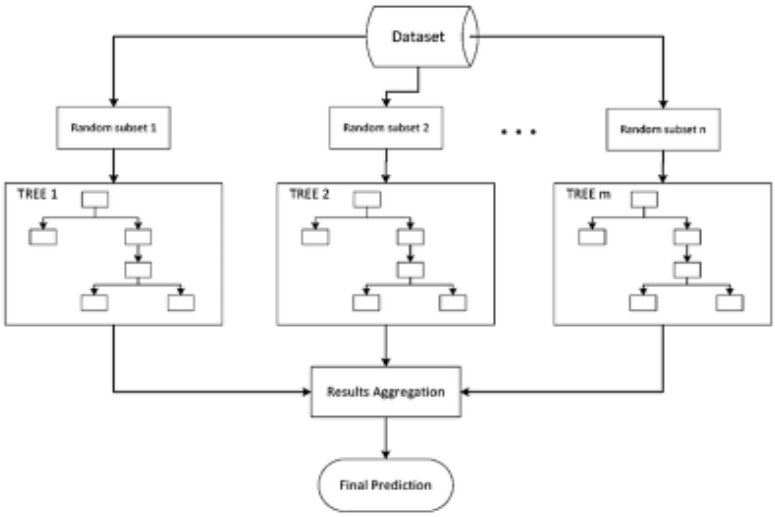


Figure 2.24: Bagging and Aggregation Block Diagram in Random Forest

2. Boosting: In boosting technique, the data is distributed in an ensemble sequential where the base learners are generated and combined sequentially. Boosting is a family of algorithms that can convert weak learners into solid learners. The basic concept of boosting or sequential method is to exploit and identify the dependence between the base learners. Instead of fitting the data randomly, it boosts to fit a sequence of weak learners, slightly better than the random guessing-like bagging technique to determine good predictions. After combining the predictions from sequential learners, predictions are combined through a weighted sum to produce the final prediction output. The widely used algorithms for boosting areaBoost, Gradient-Boost, LightGBM, CatBoost, and Extreme gradient boost (XGBoost). Most recent research shows that the use of XGBoost will surpass other boosting algorithms in performance, better predictions, and speed sometimes. And since most of this thesis is to determine close and better predictions, XGBoost is implemented. XGBoost is like any supervised machine learning algorithm with the parameter that must be initialized and tuned to find the optimum solution. XGBoost is also exposed to the overfitting phenomena since it combines various weak learners that may yield a high train error. XGBoost has vital parameters that must be tuned, like the learning rate, the trees' max depth, the number of learners' estimators, and the early stopping technique. Early stopping guides

by informing how many iterations are necessitated before the algorithms begin overfitting. This is a plus for the boosting techniques that ensures robustness.

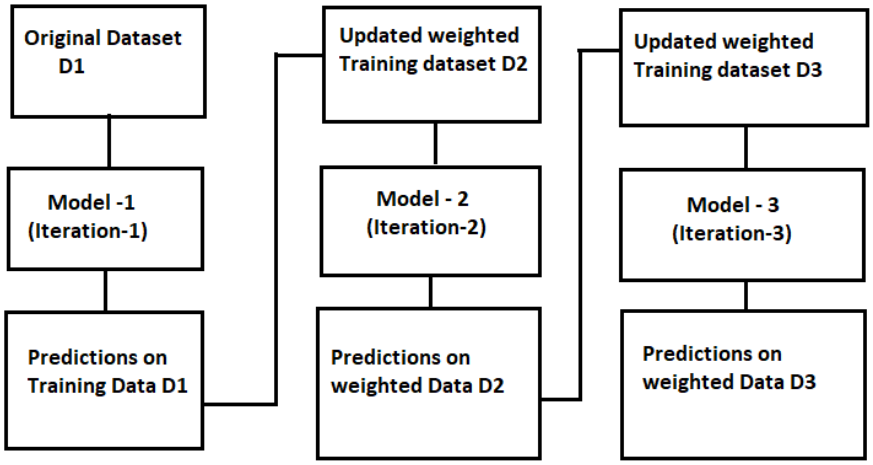


Figure 2.25: Boosting Technique Block Diagram in XGBoost

2.4.5 Artificial Neural Network with Deep Learning

Deep learning is a vast, intriguing, growing area that can also be considered a subset of machine learning. Deep learning is more competent when dealing with a vast amount of data where it can capture and mine even unstructured data like images and audio. Using artificial neural networks mimics the human brain's function using the neurons' benefits to learn from large amounts of data. Neural networks are mighty in solving complex problems in real-life situations. It can also capture the non-linearities in the data using different activation functions and optimizers. The main reason it is called Deep Learning is that when applying the artificial neural network, the net's depth and width must be initialized. The depth of the neural network is increased by adding more hidden layers to the network, whereas the width can be increased by adding additional neurons. These are the main hyperparameter of the neural network architecture, manually selected when creating the net.

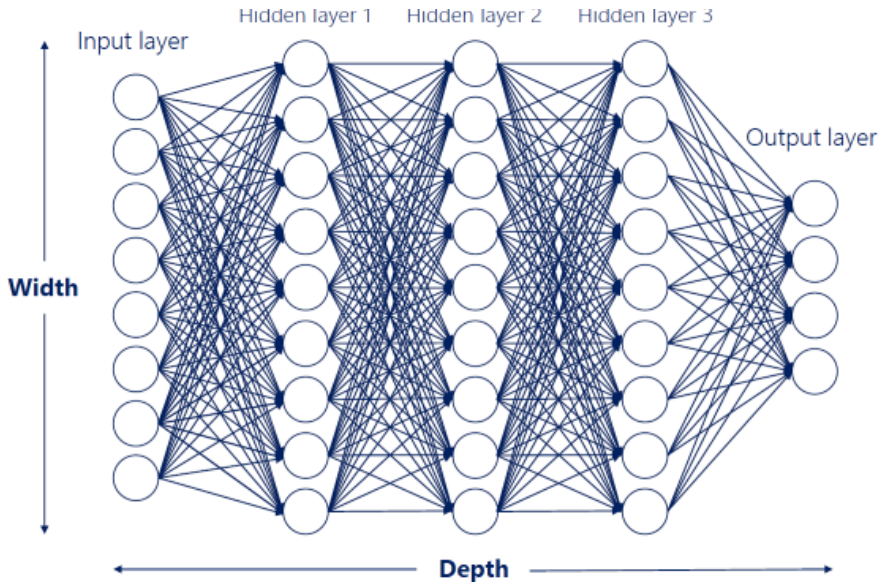


Figure 2.26: Neural Network Configuration

Neural networks work by stepping on linear combinations but adding a non-linearity. Mixing linear combinations and non-linearities allows us to model arbitrary functions. These non-linearities are required to stack layers. A neural network with non-linearities is just a linear combination. Hence, Activation functions (non-linearities) are needed to break the linearity and represent more complicated relationships. Activation functions transform inputs into outputs of a different kind.

2.4.6 Forecasting Time series algorithms

Time series algorithms are used to forecast what might happen in the future. Based on the historical data, developing a predictive time series algorithm differs from the classical machine learning algorithms. Traditional machine learning algorithms learn from historical data with no dependence on time since the data is shuffled to obtain precise predictions. In time series algorithms, the scope is different because the scope is to forecast what will happen in the future. Time series rely on keeping the chronological order of the values.

Thus, sometimes applying some machine learning methods to time-series data is not applicable because data cannot be shuffled since the chronological order is important. The objective of trying to forecast is that patterns observed in the past are expected to persist in the future. In this thesis, forecasting is applied using common time-series forecasting algorithms known with ARIMA models and the LSTM model.

2.4.6.1 SARIMAX Model

The SARIMAX model is a very advanced and sophisticated extension of the ARIMA models. It stands for The Seasonal Autoregressive Integrated Moving Average eXogenous Model. The SARIMAX is the seasonal equivalent of the ARIMAX model. Of course, there are seasonal versions of the other models (SARMA, SARIMA, SARMAX, etc.). Seasonal models help capture patterns that aren't ever-present but appear periodically. For example, the episodes of NO2 occurred in January compared to June. That is mainly due to the metrological conditions and the Increase in the number of motor vehicles. Therefore, SARIMAX is assigned to account for this expected influx of demand in January, and we can do so by monitoring the values in January of the previous year. The SARIMAX is among the most complicated models since it can incorporate seasonality, integration, and exogenous variables. However, it doesn't have to. The model can be simplified by setting the values of specific orders to 0 or not providing certain information. For instance, if considering not including exogenous variables and having no integration, the model automatically becomes equivalent to a SARMA.

Model for SARIMAX (1,0,2) (2,0,1,10):

$$X_t = C + \phi_1 X_{t-1} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \Phi_1 X_{t-10} + \phi_1 X_{t-11} + \Phi_2 X_{t-20} + \phi_1 X_{t-21} + \Theta_1(\epsilon_{t-10} + \theta_1 \epsilon_{t-11} + \theta_2 \epsilon_{t-12}) + \epsilon_t$$

2.4.6.2 Long short-term memory (LSTM)

The LSTM model is an artificial recurrent neural network (RNN) architecture. It is a complex area of Deep Learning specifically designed for sequential data. The specific feature of RNNs, over other networks like CNN, is that they have memory. RNN is a deep neural network. The net is not just deep but extremely deep; this is called unrolling the RNNs, which will end with a deep feed-forward neural network; thus, they are computationally expensive. The memory will be utilized using LSTM networks which perform well in time-series data and make accurate predictions because they can learn order dependence in the problems of sequence predictions. LSTM models are preferable in complex problem domains like speech recognition and machine translation. But they are indeed efficient when data spans over long sequences, which makes LSTM manipulate its memory state to solve the prediction problem.

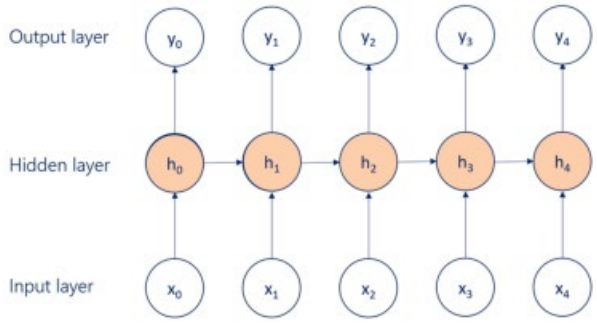


Figure 2.27: Simple RNNs Architecture

2.4.6.3 NeuralProphet

In our highly computerized world, the importance of time-series forecasting models has been heavily studied by Facebook (Metaverse) developers to introduce the NeuralProphet model. This promising model gained lots of popularity due to the easiness of application for the user, interpretable results, and excellent performance in terms of accuracy. It also provides hyperparameter selection and feature engineering automation, which is relatively straightforward and will save much time in collecting results. It works by starting with the autoregressive models that attempt to predict future values of the output based on its past observations. Generally, they are linear models that use the predicted variable's lagged values. But what makes the NeuralProphet very special and different from the Prophet models and other time-series algorithms is that the Author's kept in mind all the differences and improved its accuracy and scalability by introducing an improved and complex backend by PyTorch and using Autoregressive Network to combine the elegant scalability of Neural Networks with the interpretability of the AR models. The below points summarize the additional features of the NeuralProphet model. The below points to summarize the additional features of the NeuralProphet model.

A. The modeling process is much faster because it uses PyTorch's gradient descent for optimization.
B. Auto-Regressive Network is used to model Time-series autocorrelation.
C. Feed-Forward Neural Network is used to model lagged regressors.
D. The model has non-linear deep layers of the Feed-Forward NNs.
E. The model is easily tunable to specific forecasting horizons.
F. The model offers custom losses and metrics.

Training a model

Training a model is the heart of the machine-learning process. At this stage bulk of learning is done. The quality and quantity of any training dataset will remarkably impact the model's performance. After collecting, cleaning, preprocessing the data, and selecting several models, the user fits this data to teach the machine to differentiate between inputs and outputs or find a continuous value. While training a model, the machine tries to minimize the cost function by fitting the best weights and biases to a machine learning algorithm. To ensure excellent training that can produce a robust model, the data must be split into a validation set to avoid overfitting phenomena and provide an unbiased evaluation of the performance of several candidate models. The last part is the test set to evaluate the model's performance in unseen data to measure the generalization error before deploying the model.

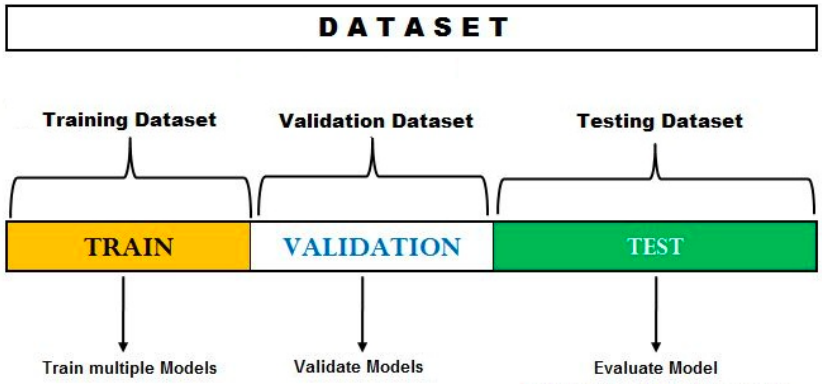


Figure 2.28: Train Validate Test Split

2.5. Hyperparameter tuning

Because the initial random values might often produce unsatisfactory results, as they are refined by selecting the best parameters to tune and retune the model to improve the training and predictions, hyperparameter tuning is a necessary step that the user can't neglect, it can be deemed as a trial-and-error technique that can identify the optimal parameters of a model to produce precise predictions, but since trial and error is tedious, time-consuming strategy, and may not give good results, there are built-in functions in the marvelous library sci-kit-learn that automates to search for the best Hyperparameters of the model that can give the optimum solutions and predictions. Two functions come in this package, GridSearchCv, and RandomizedSearchCV. They were accommodating in this study to speed up the selections of the parameters of different algorithms by adding only a list or dictionary of a wide range of different parameters.

2.5.1 Grid-Search-Cv

GridSearchCv can search multiple parameters simultaneously. GridSearchCv allows defining a set of parameters that could be tried with a given model and automatically runs cross-validation using each parameter to keep track of the resulting scores. It replaces the natural for loop and provides some additional functionalities. For instance, after creating a python list fork range that specifies the k values that we would like to search, then we create what is known as a parameter grid. It is simply a python dictionary in which the key is the parameter name, and the value is a list value that should be searched for that parameter. Finally, we will instantiate the grid by adding K-Folds cross-validation on a specified algorithm using the desired evaluation metrics.

2.5.2 Randomized-Search-Cv

The RandomizedSearchCv technique is a close cousin of a GridSearchCv. The problem that RandomizedSearchCv aims to solve is that the search can quickly become computationally infeasible when performing an exhaustive search of many different parameters at once. This technique was developed to reduce the computational expense of thousands of cross-validation trials, which will be adequate to a hundred thousand model fit. RandomizedSearchCv solves this problem by searching only a random subset of the provided parameters and allowing the user to explicitly control the number of attempted parameter combinations.

Chapter 3

Results

3.1 Hyperparameter tunning for Machine Learning & Deep learning algorithms

3.1.1 KNN Algorithm Architecture

Table 3.1.1: KNN Hyperparameters after RandomizedSearchCV

n_ neighbors	Leaf_ size	metric	weights	algorithm	p	Cross validation	Candidates	Elapsed time	Total fits
8	28	minkowski	uniform	auto	2	10	240	5 min	600

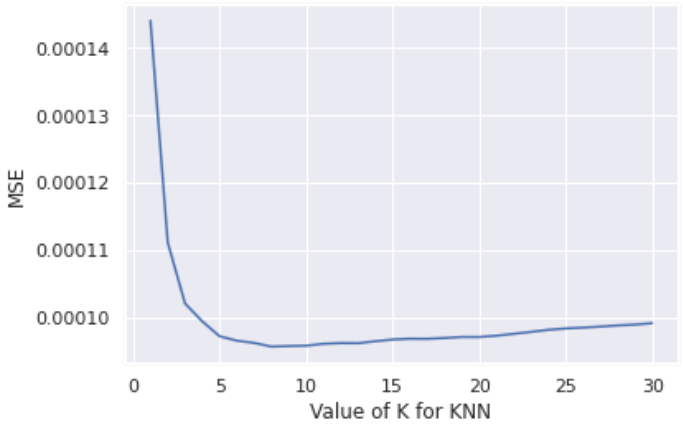


Figure 3.1: Values of KNN after RandomizedSearchCV

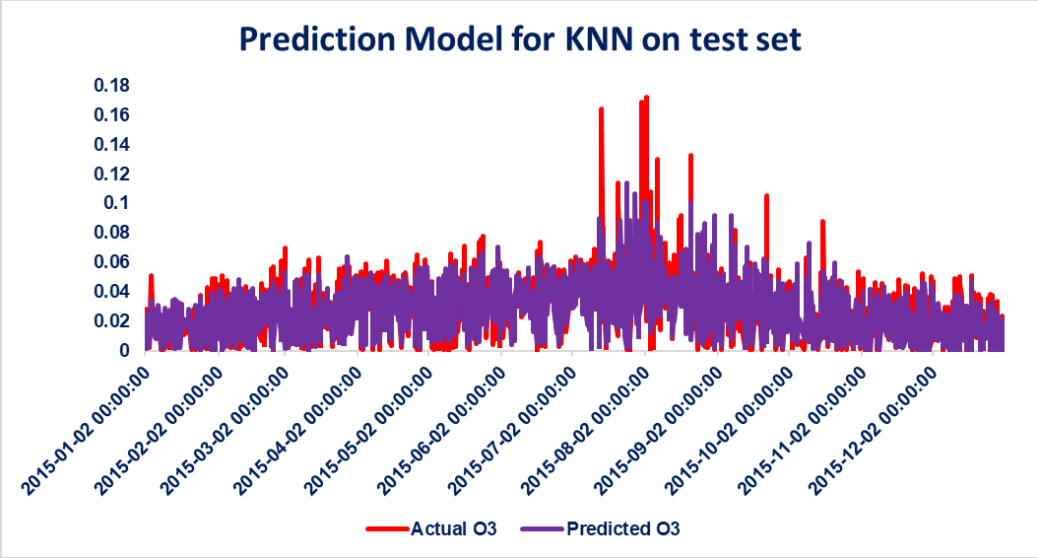


Figure 3.2: Prediction Model for KNN on test set

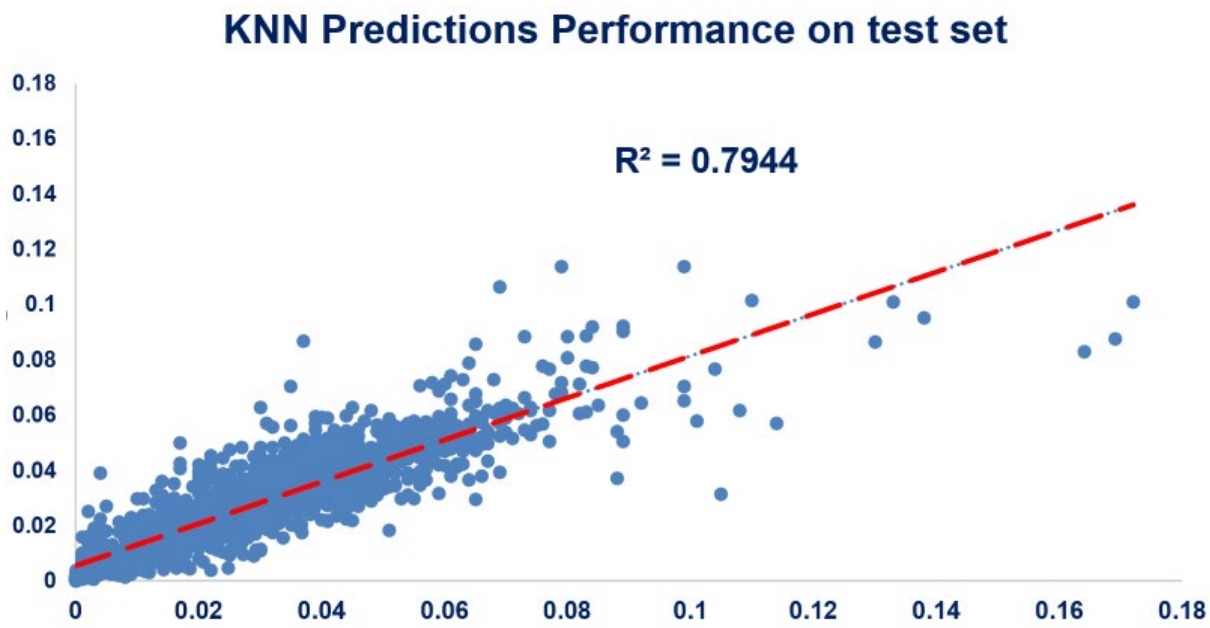


Figure 3.3: KNN Predictions Performance on test set

3.1.2 KNN Algorithm Metrics after Hyperparameter tuning.

Table 3.1.2: KNN metrics after RandomizedSearchCV

Metrics	Value
Mean Absolute Error (MAE)	0.006600441
Mean Squared Error (MSE)	0.000092354
Root Mean Squared Error (RMSE)	0.009610087
Mean Absolute Percentage Error (%)	14.965
R^2 (train)	0.81
R^2 (validation)	0.80
R^2 (test)	0.79

3.2.1 Random Forest Algorithm Architecture

Table 3.2.1: Random Forest Hyperparameters after exhaustive GridSearchCV

n_ estimators	Max depth	min samples split	Min Samples leaf	Cross validation	Elapsed time	Candidates	Total fits

1100	15	10	10	5	39 min	600	3000

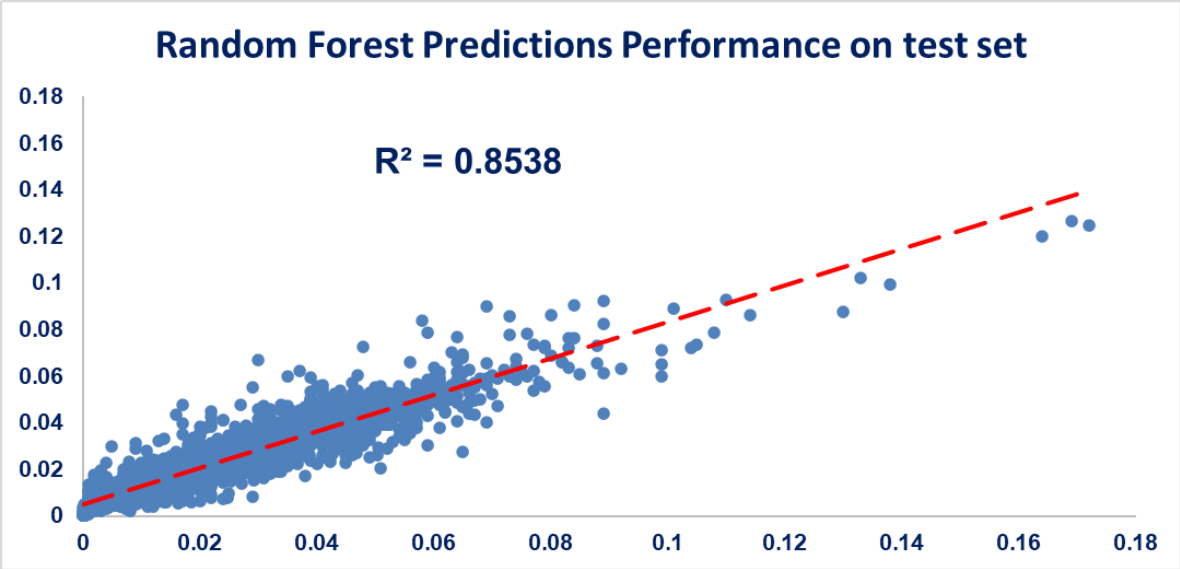


Figure 3.4: Random Forest Regressor Predictions Performance on Test set

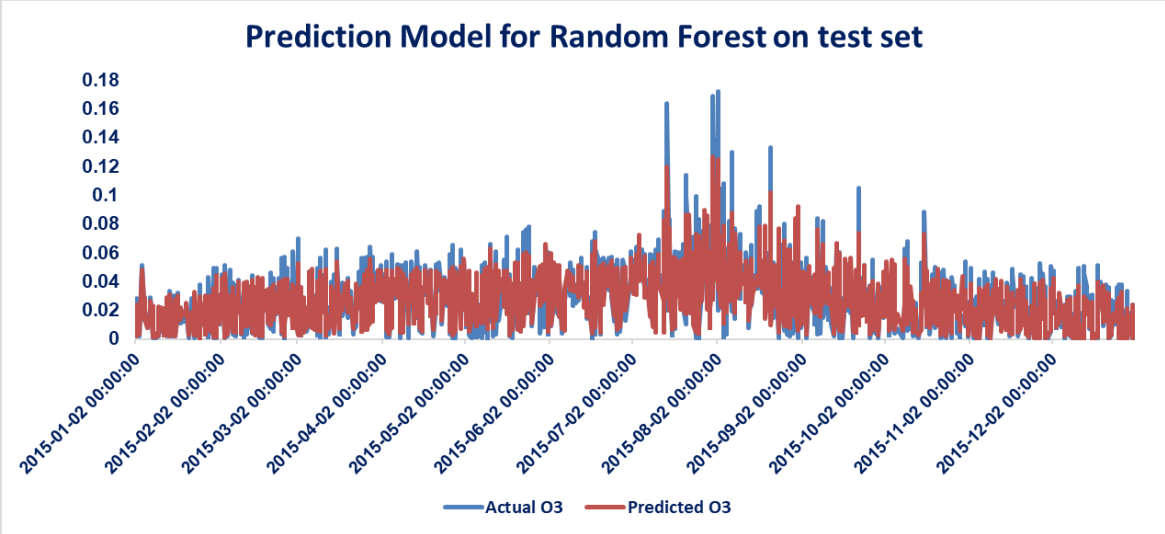


Figure 3.5: Prediction Model for Random Forest on test set

3.2.2 Random Forest Algorithm Metrics after Hyperparameter tuning

Table 3.2.2: Metrics of Random forest after exhaustive GridSearchCV

Metrics	Value
Mean Absolute Error (MAE)	0.0058729
Mean Squared Error (MSE)	0.00007292
Root Mean Squared Error (RMSE)	0.0085397
Mean Absolute Percentage Error (%)	11.444
R^2 (train)	0.89
R^2 (validation)	0.87

R^2 (test)	0.85

3.3.1 XGBoost Algorithm Architecture

Table 3.3.1: XGBoost Hyperparameters after RandomizedSearchCV

n_ estimators	Max depth	Base score	Booster	Learning rate	Min child weight	Cross validation	Elapsed time	Total fits
700	8	1	gbtree	0.05	4	5	45 min	100

3.3.2 XGBoost Forest Algorithm Metrics after Hyperparameter tuning

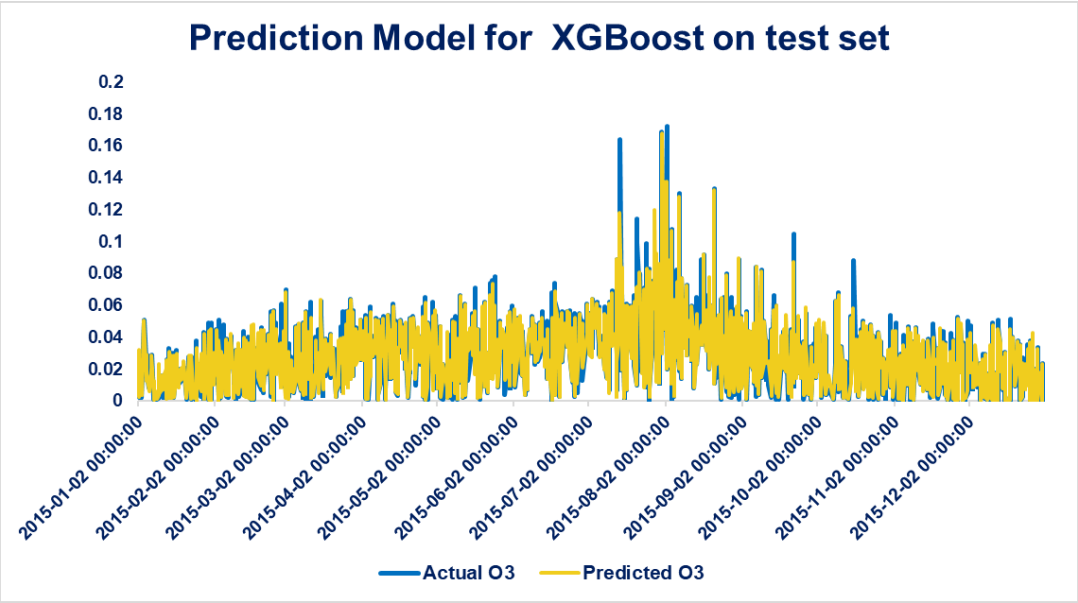


Figure 3.6: Prediction Model for XGBoost on test set

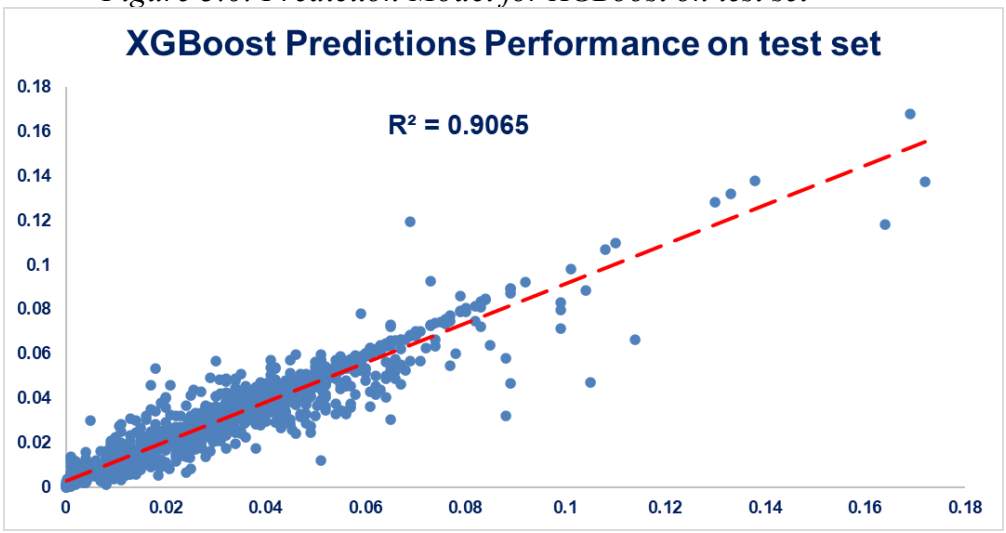


Figure 3.7: XGBoost Predictions Performance on Test set

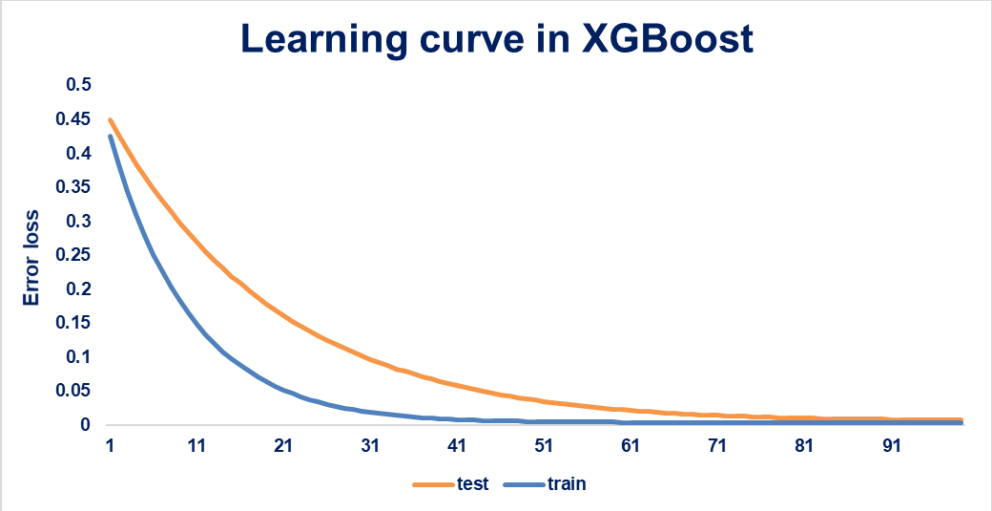


Figure 3.8: Learning curve in XGBoost

Table 3.3.2: Metrics of XGBoost after exhaustive RandomizedSearchCV

Metrics	Value
Mean Absolute Error (MAE)	0.003716055
Mean Squared Error (MSE)	0.000039167
Root Mean Squared Error (RMSE)	0.006258371
Mean Absolute Percentage Error (%)	9.4956
R^2 (train)	0.96
R^2 (validation)	0.93
R^2 (test)	0.905

3.4.1 CatBoost Algorithm Architecture

Table 3.4.1: CatBoost Hyperparameters after RandomizedSearchCV

n estimators	Max depth	Learning rate	l2_leaf_reg	Loss function	Cross validation	Elapsed time	Total fits
500	6	0.01	0.1	RMSE	5	35 min	100

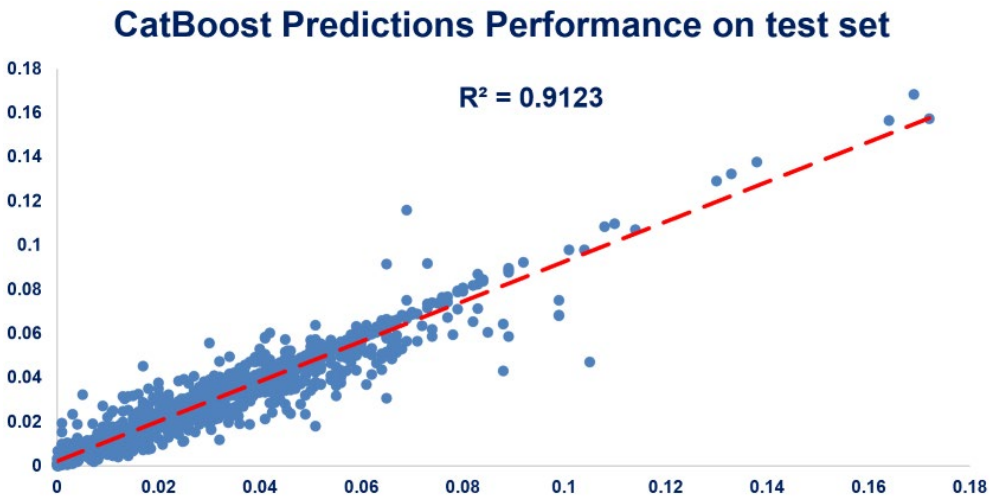


Figure 3.9: CatBoost Predictions Performance on Test set

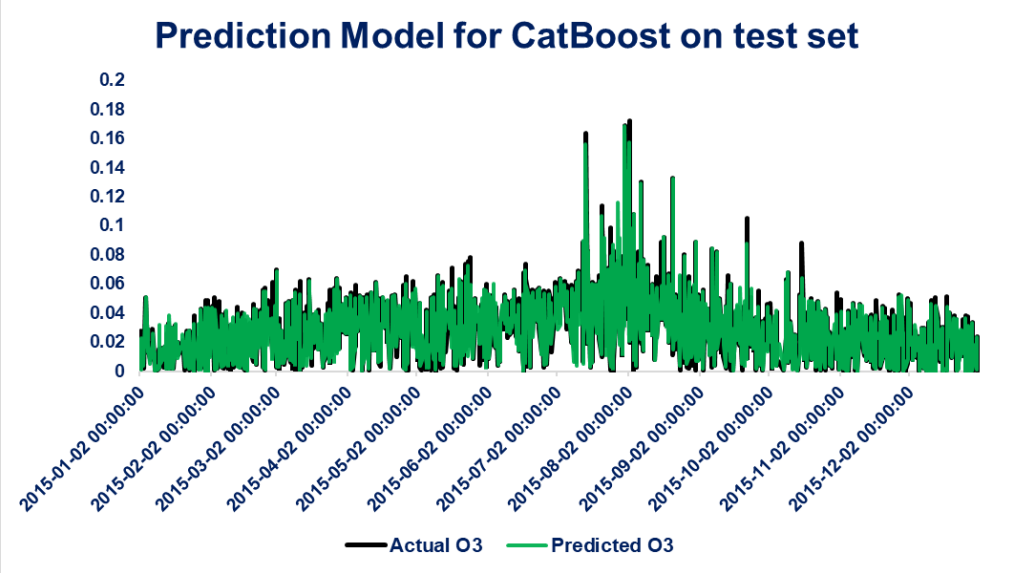


Figure 3.10: Prediction Model for CatBoost on test set

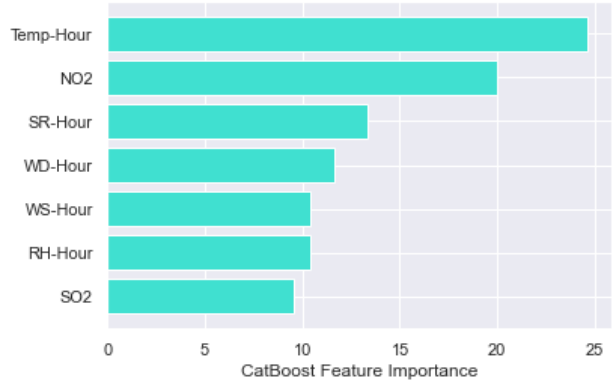


Figure 3.11: CatBoost Feature importance after fitting the data

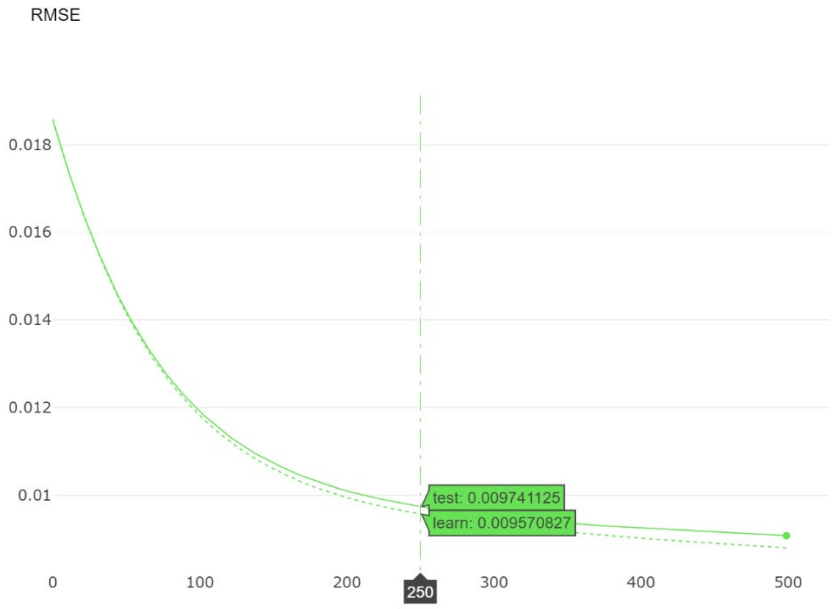


Figure 3.12: Learning curve for CatBoost



Figure 3.13: SHAP value for CatBoost algorithm

3.4.2 CatBoost Forest Algorithm Metrics after Hyperparameter tuning

Table 3.4.2: Metrics of CatBoost after exhaustive RandomizedSearchCV

Metrics	Value
Mean Absolute Error (MAE)	0.003662315
Mean Squared Error (MSE)	0.0000372414
Root Mean Squared Error (RMSE)	0.0060102577
Mean Absolute Percentage Error (%)	9.2
R^2 (train)	0.97
R^2 (validation)	0.935
R^2 (test)	0.91

3.5.1 Artificial Neural Network Architecture

Table 3.5.1: Artificial Neural Network Hyperparameters after Keras Tuner

Hidden layers	Total neurons	Learning rate	Batch size	Activation function	Optimizer	Dropout (L2 regularizer)	Optimum epochs
7	220	0.001	8	Relu	Adam	0	20

```
Trial 9 Complete [00h 01m 07s]
val_mean_squared_error: 8.855688065523282e-05

Best val_mean_squared_error So Far: 7.868838292779401e-05
Total elapsed time: 00h 14m 29s

Search: Running Trial #10

Hyperparameter    |Value            |Best Value So Far
num_layers        |5                |8
units_0           |16               |40
units_1           |32               |36
learning_rate     |0.001            |0.001
units_2           |44               |20
units_3           |40               |36
units_4           |24               |16
units_5           |12               |12
units_6           |32               |32
units_7           |16               |28

Trial 10 Complete [00h 01m 14s]
val_mean_squared_error: 8.465924474876374e-05

Best val_mean_squared_error So Far: 7.868838292779401e-05
Total elapsed time: 00h 15m 44s
INFO:tensorflow:Oracle triggered exit
```

Figure 3.14: Results of Hyperparameter tunning from Keras Tuner

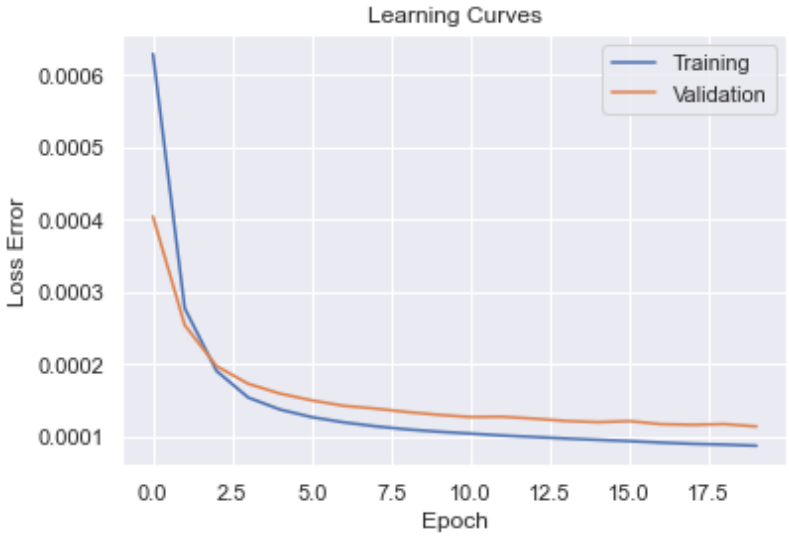


Figure 3.15: Neural network learning curve with early stopping applied

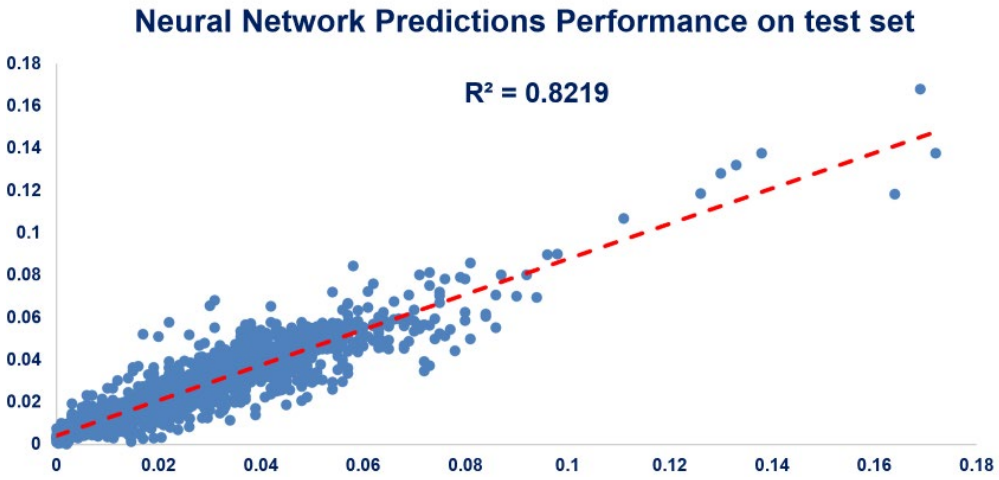


Figure 3.16: Neural Network Predictions Performance on Test set

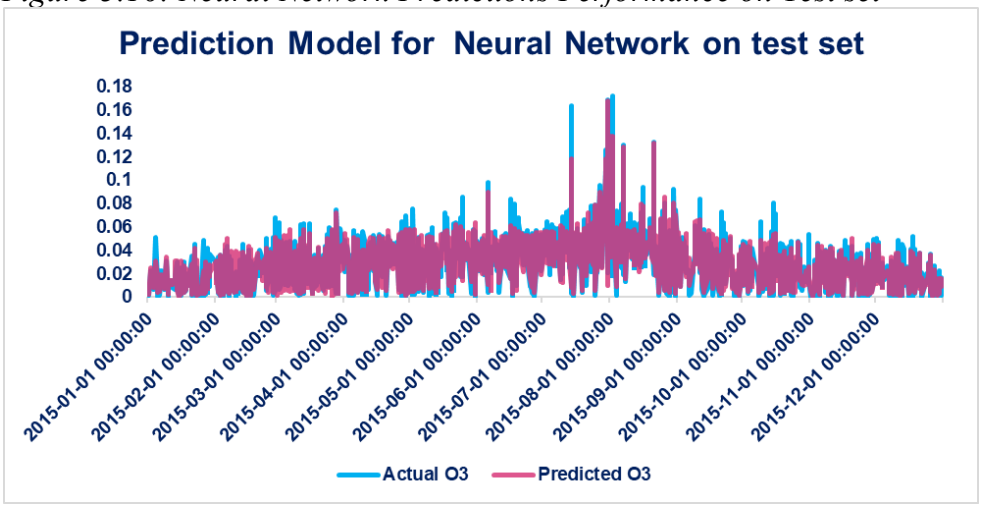


Figure 3.17: Prediction Model for Neural Network on test set

3.5.2 Artificial Neural Network Metrics after Hyperparameter tuning

Table 3.5.2: Metrics of Artificial Neural Network after Keras Tuner

Metrics	Value
Mean Absolute Error (MAE)	0.006348729
Mean Squared Error (MSE)	0.000085466
Root Mean Squared Error (RMSE)	0.009265346
Mean Absolute Percentage Error (%)	13.0107
R^2 (train)	0.84
R^2 (validation)	0.83
R^2 (test)	0.82

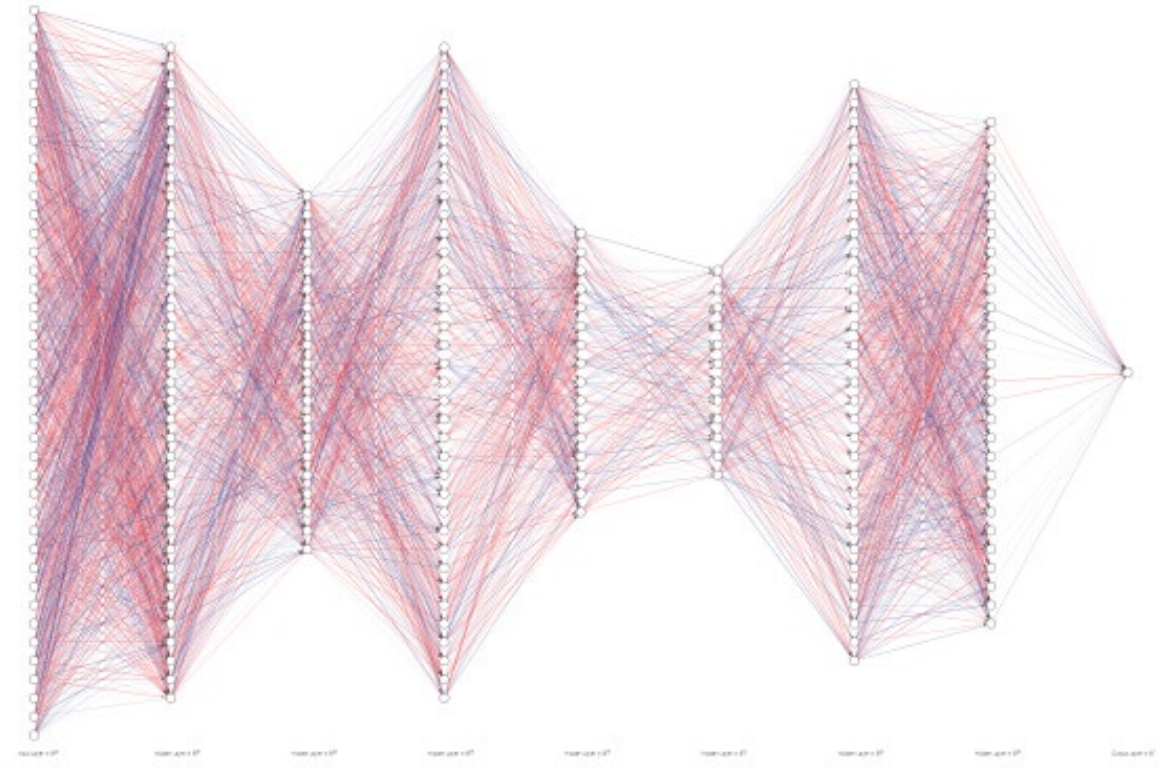


Figure 3.18: Neural Network Architecture is computationally expensive with high metrics

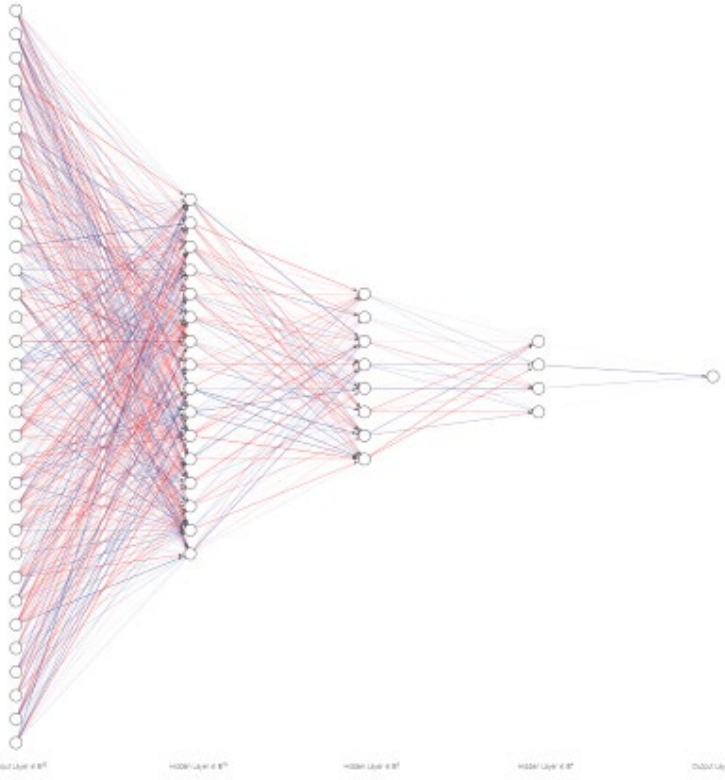


Figure 3.19: Neural network with 60 neurons achieves an r2 of 0.6 and an RMSE of 0.0126

3.6 Results After Hyperparameter tunning for forecasting timeseries algorithms

3.6.1 SARIMAX Architecture

Table 3.6.1: SARIMAX Hyperparameters after AIC minimization

p	d	q	Stationary	p-value	Seasonality	Lags
3	1	2	Yes	1.5×10^{-7}	0	10

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=-7108.770, Time=4.03 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-6787.555, Time=1.17 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-6939.358, Time=1.40 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-7032.454, Time=3.00 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-6787.492, Time=3.20 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=-7083.382, Time=3.47 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=-7077.358, Time=1.90 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=-7128.017, Time=2.37 sec
 ARIMA(3,1,1)(0,0,0)[0] intercept   : AIC=-7036.214, Time=1.99 sec
 ARIMA(4,1,2)(0,0,0)[0] intercept   : AIC=-7114.774, Time=5.03 sec
 ARIMA(3,1,3)(0,0,0)[0] intercept   : AIC=-7126.174, Time=2.46 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=-7117.459, Time=2.67 sec
 ARIMA(4,1,1)(0,0,0)[0] intercept   : AIC=-7069.232, Time=4.72 sec
 ARIMA(4,1,3)(0,0,0)[0] intercept   : AIC=-7123.229, Time=5.68 sec
 ARIMA(3,1,2)(0,0,0)[0]             : AIC=-7130.068, Time=2.37 sec
 ARIMA(2,1,2)(0,0,0)[0]             : AIC=-7110.773, Time=5.10 sec
 ARIMA(3,1,1)(0,0,0)[0]             : AIC=-7038.214, Time=3.27 sec
 ARIMA(4,1,2)(0,0,0)[0]             : AIC=-7116.781, Time=2.33 sec
 ARIMA(3,1,3)(0,0,0)[0]             : AIC=-7128.204, Time=1.68 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=-7079.388, Time=0.97 sec
 ARIMA(2,1,3)(0,0,0)[0]             : AIC=-7119.522, Time=2.48 sec
 ARIMA(4,1,1)(0,0,0)[0]             : AIC=-7071.231, Time=4.21 sec
 ARIMA(4,1,3)(0,0,0)[0]             : AIC=-7125.256, Time=2.76 sec

Best model:  ARIMA(3,1,2)(0,0,0)[0]
Total fit time: 68.303 seconds
```

Figure 3.20: AIC minimization

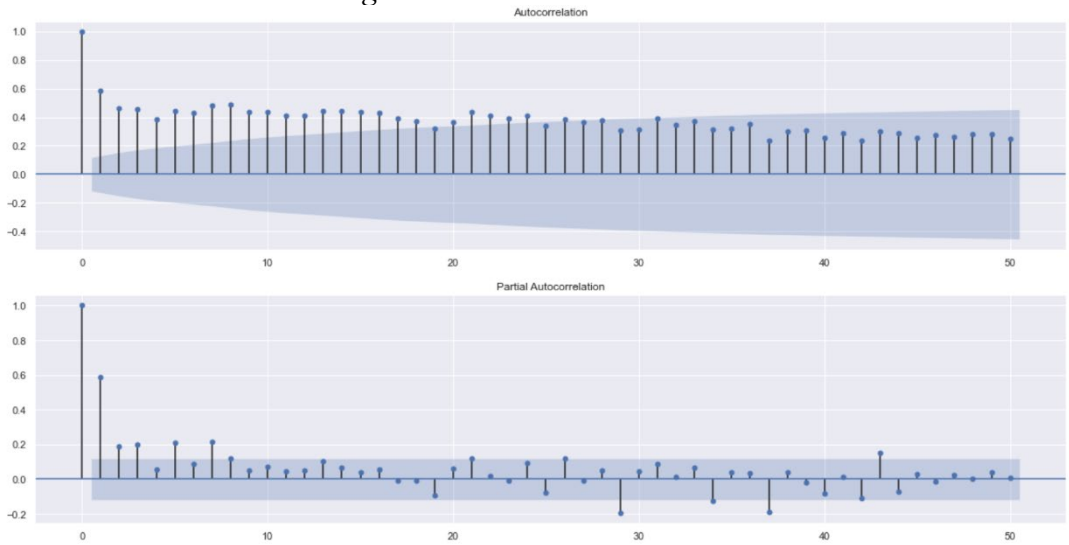


Figure 3.21: ACF and PACF plots

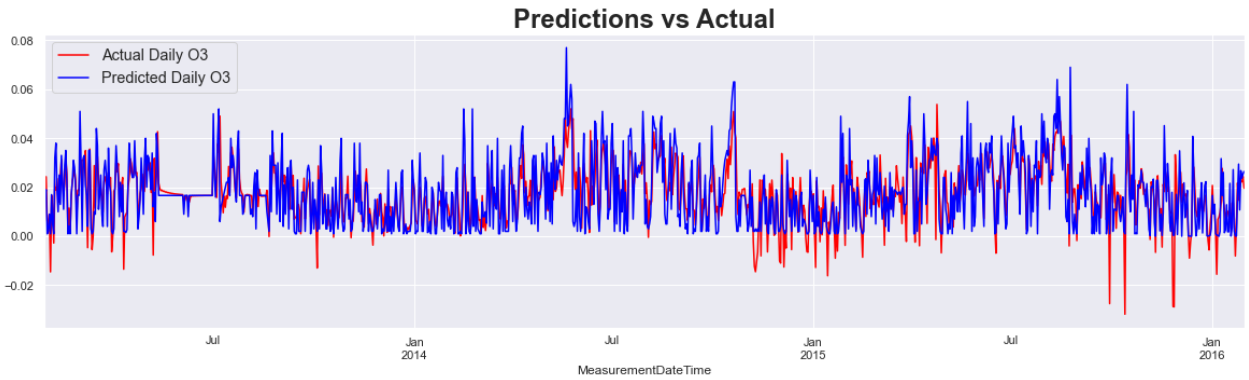


Figure 3.22: 80% split from the trainset

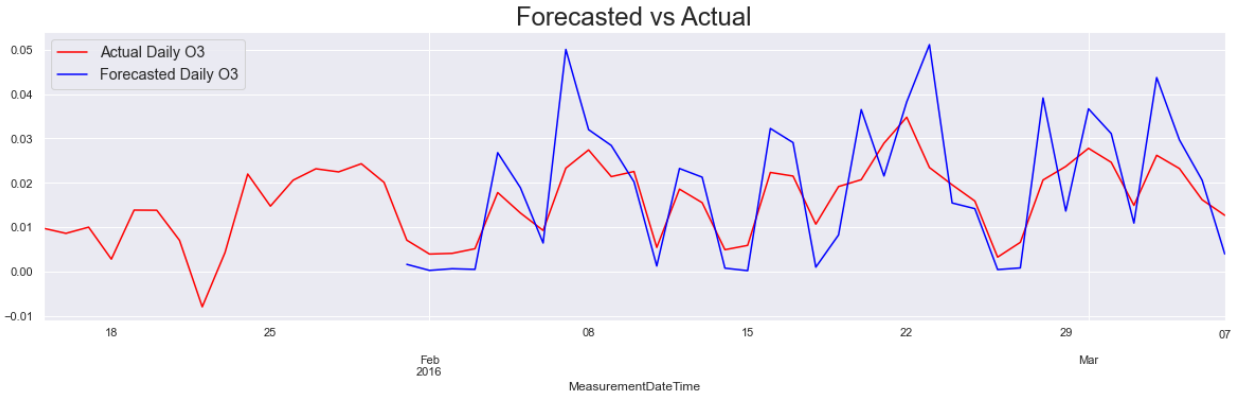


Figure 3.23: Forecasted Daily O3 on the daily test set

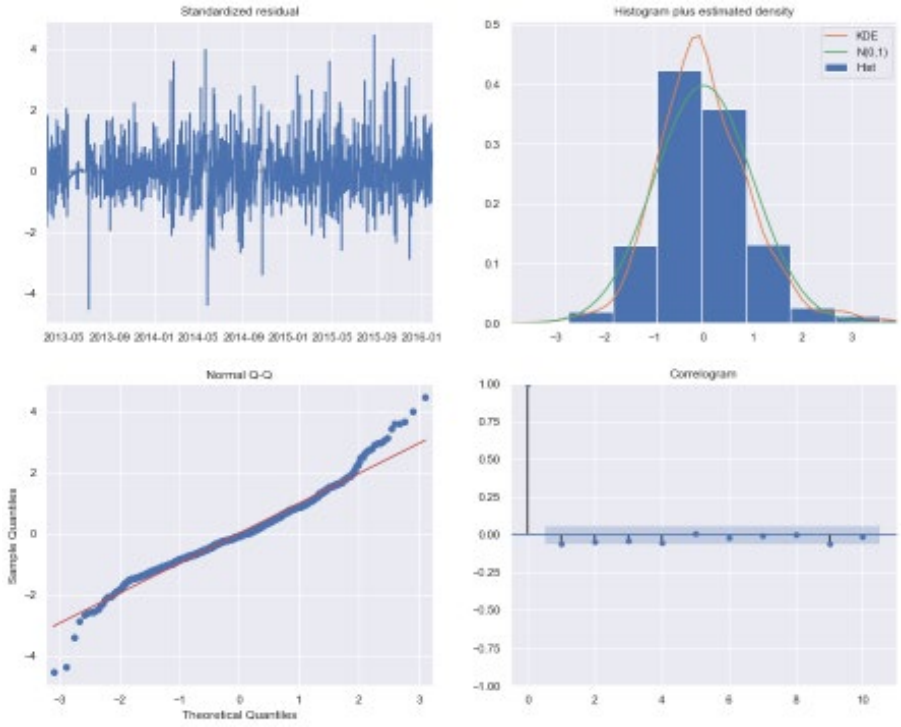


Figure 3.24: Residuals Diagnostics Plot

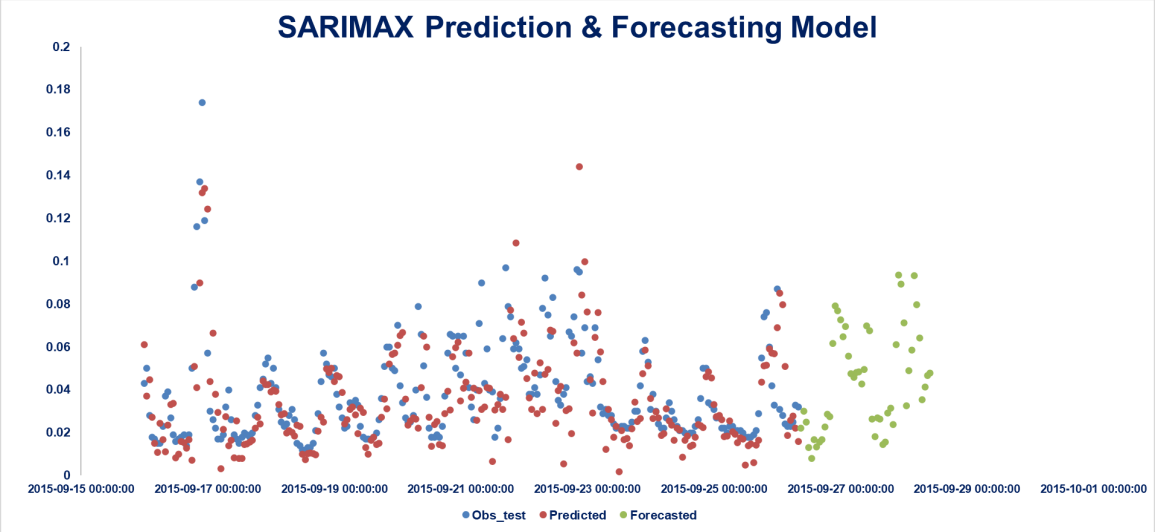


Figure 3.25: Hourly forecasted O3 on test set after 90% split

3.6.2 SARIMAX Metrics after AIC minimization

Table 3.6.2: Metrics of SARIMAX after Model Evaluation

Metrics	MAE	MSE	RMSE
Value	0.0198684	0.0005844	0.0241745

3.6.3 NeuralProphet & Prophet

```
MSE comparison ----
Prophet:        0.0002
NeuralProphet:  0.0001
```

Figure 3.26: Metrics from Modeling Timeseries NeuralProphet and Prophet

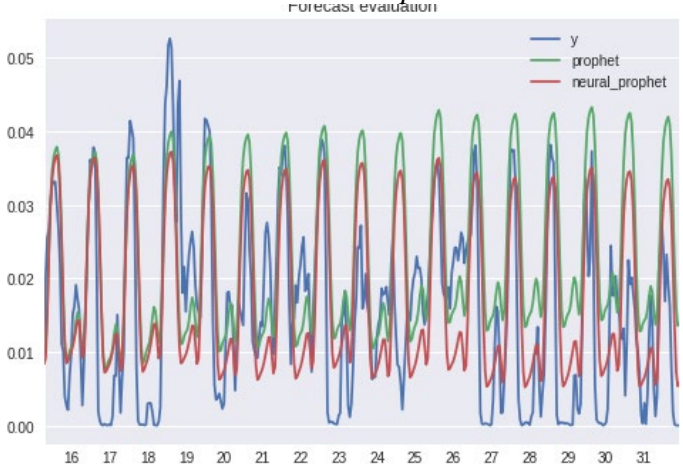


Figure 3.27: Comparison of observed vs. NeuralProphet and Prophet

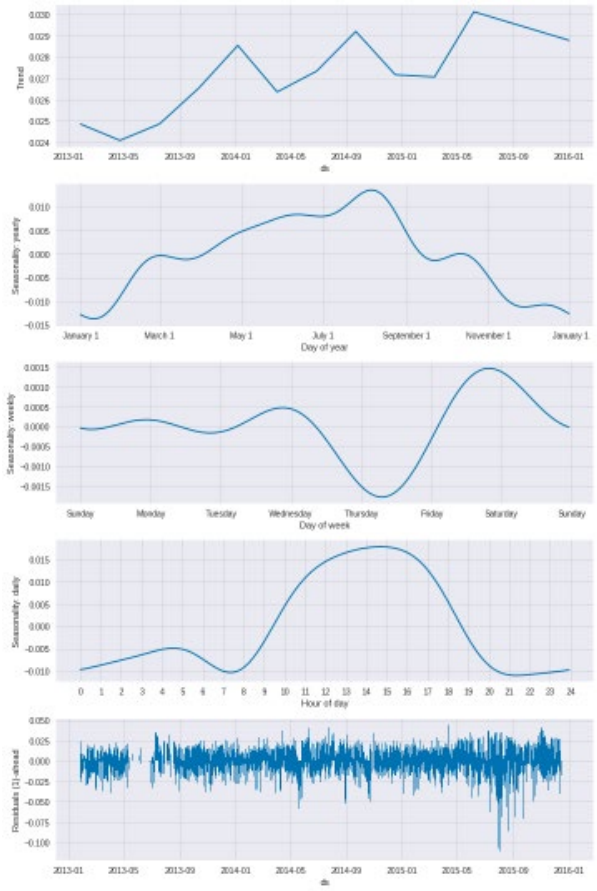


Figure 3.28: Residuals Diagnostics Plot from NeuralProphet

3.2 Sensitivity Analysis

Sensitivity analysis is a widely used technique in machine learning to optimize models. It is also a powerful technique for estimating the impact of changes to model parameters on performance. In this technique, we identify the parameters that significantly impact model performance and test different values for those parameters to see how they affect the model's performance. Sensitivity analysis is a technique used in machine learning to evaluate the performance of a model and identify the factors that impact its predictions the most. It is a model interpretability method that helps to understand how the model makes decisions and what factors are most important in predicting the target variable. Sensitivity analysis involves perturbing the input data in some way and observing how the model's predictions change in response. For example, you might add or remove a feature from the input data or change the value of a feature to see how the model's predictions are affected. By doing this, you can identify which features are

most important to the model and how changes in the input data affect the model's predictions. Sensitivity analysis is an essential tool for understanding and improving the performance of machine learning models. It allows you to identify potential sources of error or bias in the model, and it can help you to fine-tune the model to improve its performance on new data.

Additionally, it can provide valuable insights into the relationships between the input features and the target variable. It can be helpful in interpreting the model's predictions and making better business decisions. To do so, we will compare different supervised machine learning algorithms so that we can apply some optimization techniques to ensure that the model is working well in the production phase after the deployment. In the results section, the boosting technique outperforms the bagging technique, Artificial Neural Network, and time-series algorithms. Our results are compatible with the most recent research that confirms the preferability of boosting algorithms over other algorithms. Specialists started to use the boosting algorithms over the deep learning models in the tabular data since it is costly, time-consuming, and more applicable in unstructured data like images and text. After trying different machine learning algorithms, the best boosting technique in our result was the CatBoost algorithm. To confirm our results, we used the pycaret library in python, an auto-machine learning library that compares different machine learning algorithms in a shorter time. The results show that the CatBoost Regressor is the best algorithm for metrics like R^2, MAE, MSE, and MAPE for this particular data.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	0.0043	0.0000	0.0063	0.8809	0.0061	4.3849	2.2190
xgboost	Extreme Gradient Boosting	0.0043	0.0000	0.0064	0.8787	0.0061	6.0682	0.4360
et	Extra Trees Regressor	0.0041	0.0000	0.0064	0.8777	0.0061	3.4533	0.9800
lightgbm	Light Gradient Boosting Machine	0.0048	0.0000	0.0070	0.8536	0.0067	9.2379	0.2140
rf	Random Forest Regressor	0.0045	0.0001	0.0071	0.8495	0.0068	3.9257	1.6670
gbr	Gradient Boosting Regressor	0.0062	0.0001	0.0089	0.7629	0.0086	15.3717	0.5730
dt	Decision Tree Regressor	0.0062	0.0001	0.0104	0.6787	0.0100	1.7337	0.0440
lr	Linear Regression	0.0084	0.0001	0.0115	0.6051	0.0110	37.2053	0.6250
lar	Least Angle Regression	0.0084	0.0001	0.0115	0.6051	0.0110	37.5542	0.0140
br	Bayesian Ridge	0.0084	0.0001	0.0115	0.6051	0.0110	37.2033	0.0160
ridge	Ridge Regression	0.0084	0.0001	0.0115	0.6035	0.0110	37.8741	0.0130
knn	K Neighbors Regressor	0.0084	0.0001	0.0118	0.5852	0.0114	48.5364	0.0360
ada	AdaBoost Regressor	0.0110	0.0002	0.0131	0.4864	0.0127	143.3527	0.3080
omp	Orthogonal Matching Pursuit	0.0118	0.0002	0.0151	0.3208	0.0146	159.7325	0.0130
en	Elastic Net	0.0119	0.0002	0.0151	0.3201	0.0146	177.8048	0.0120
lasso	Lasso Regression	0.0121	0.0002	0.0154	0.2932	0.0149	188.8234	0.0130
huber	Huber Regressor	0.0116	0.0003	0.0169	0.0927	0.0159	31.1456	0.2570
llar	Lasso Least Angle Regression	0.0147	0.0003	0.0183	-0.0009	0.0177	233.4750	0.0150
dummy	Dummy Regressor	0.0147	0.0003	0.0183	-0.0009	0.0177	233.4750	0.0120
par	Passive Aggressive Regressor	0.0430	0.0021	0.0458	-5.2891	0.0439	586.9311	0.0140

Figure 3.29: Comparison between machine learning algorithms using pycaret automl

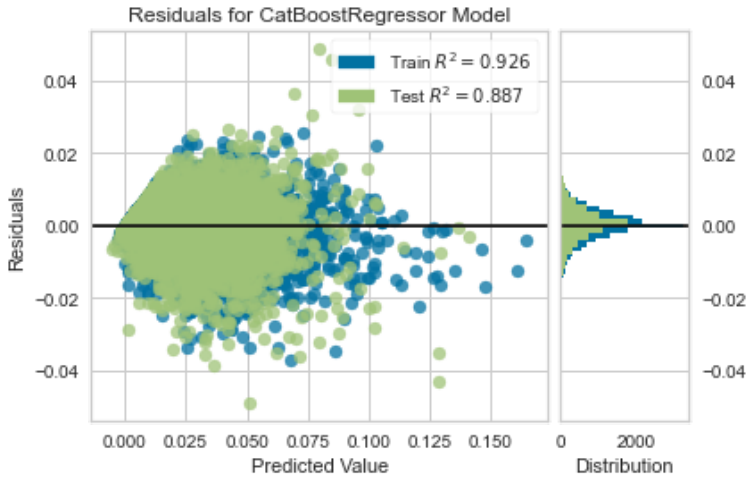


Figure 3.30: Residuals for catboost model in pycaret automl

From figure 30, we can observe a good R^2 in the train and test dataset, as well as a small relative errors since most of the points are close to zero and the shape represents a Gaussian distribution.

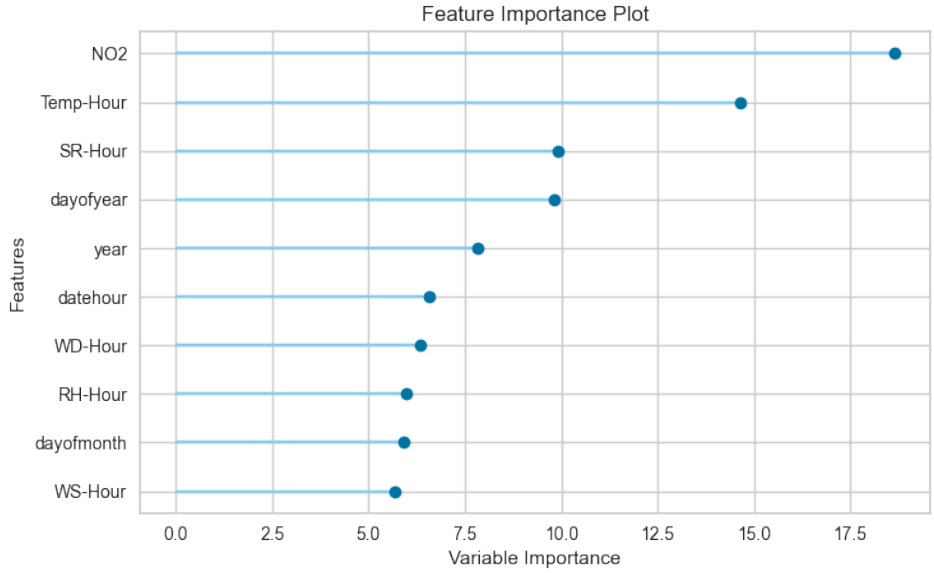


Figure 3.31: Feature importance in pycaret automl

3.3 Model Deployment

Model deployment makes a trained machine-learning model available in a production environment. This typically involves packaging the trained model, along with any necessary dependencies and pre-processing steps, into a deployable format that can be easily integrated with other systems and accessed by users or applications. Model deployment is an essential step in the machine learning workflow, as it allows the model to make predictions or inferences on new data

in a real-world setting. This can be useful for various applications, such as automating decision-making processes, providing personalized recommendations, or detecting anomalies in real-time data streams. Several challenges are associated with deploying machine learning models, such as ensuring that the model is efficient and scalable and can be integrated seamlessly with other systems and applications. To address these challenges, several tools and frameworks are available that facilitate the deployment of machine learning models, such as TensorFlow Serving, Kubeflow, and AWS SageMaker. Overall, model deployment is an essential step in the machine learning workflow that enables trained models to be used in real-world applications and helps to unlock the value of machine learning for businesses and organizations. Managing data is one of the biggest challenges when deploying machine learning models. In a traditional data processing environment, we would have one or more servers housing our models and the datasets used to train them. We would then have to decide how to scale out our new model or set of models to the rest of our application infrastructure. This can be complicated and expensive. Instead, we can deploy machine learning models on DataStream's platform and have them run in containers in your production environment or different cloud platforms to choose from, including AWS, Azure, and GCP. This approach allows us to add new functionality to our applications without updating or maintaining any of the underlying infrastructures. In this thesis, we will use a very different approach using the streamlit library in python. Streamlit is an open-source Python library for building and deploying interactive, web-based applications for data science and machine learning. It was developed to make it easy for data scientists and machine learning engineers to create beautiful, data-rich applications that can be shared with others simply and intuitively. Streamlit provides a high-level API that allows you to create applications using only a few lines of Python code. It automatically handles the underlying web technologies and provides a set of built-in widgets and components that you can use to build your application. This makes it easy to create visually appealing and user-friendly applications without needing expertise in web development. Some of the critical features of Streamlit include support for data visualization using popular libraries such as Matplotlib and Plotly, the ability to create interactive components such as sliders and dropdown menus, and the ability to deploy your application to the web with just a few clicks. Overall, Streamlit is a powerful and easy-to-use library that simplifies the process of building and deploying interactive, web-based applications for data science and machine learning. Data scientists and machine learning engineers widely use it to create and share their work with others.

https://ahmedewis-ozone-predictor-by-catboost-cat-api-1x573m.streamlit.app/

Chapter 4
Conclusion and Recommendations

4.1 Conclusion

In conclusion, this thesis this sheds light on the crucial issue of air quality in Kuwait and provides a comprehensive evaluation for the hourly data of the air quality in the urban localities of Al Jahra and Saad Al Abdullah in Kuwait. The research aimed to identify the hourly pollution levels of nitrogen dioxide (NO_2) and ozone (O_3) and establish correlations between them. The study found a clear seasonal pattern in the concentration levels of NO_2 and O_3, with higher NO_2 concentrations in the winter and higher O_3 concentrations in the summer, this study provides insights into the extent and nature of the problem. The use of advanced statistical methods, including machine learning and deep learning techniques, provided insights into the relationship between NO_2 and O_3. The results indicated a strong negative correlation between the two pollutants, with the CatBoost algorithm demonstrating the highest level of accuracy for this type of data. These findings provide a basis for future research in the field and allow for better forecasting and decision making by the authorities. This thesis serves as a valuable contribution to the field of air quality assessment and underscores the importance of monitoring and improving air quality in Kuwait.

4.2 Recommendations

For future recommendations, experts advise collecting more data and retraining the model to capture sudden changes in behavior, to ensure that the model can capture the most recent patterns and variations in the data. This will ensure that the model remains up-to-date and accurate, providing more valuable insights and information for monitoring and improving air quality in Kuwait and beyond. However, it is important to note that data drift, where the underlying statistical properties of the data change over time, can affect the accuracy of the model. To avoid data drift and ensure the sustainability of the model, it is recommended to regularly collect more data and retrain the model to capture sudden changes in behavior. This will help in maintaining the model's accuracy and relevance, and ensure that it continues to provide valuable insights into the relationship between NO_2 and O_3 and their impact on air quality.

References

Raslan Alenezi, A. A. Assessment of Ambient Air Quality in Al Jahra Governorate, for 2008.

Al-Salem, S. M., & Khan, A. R. (2010). Monitoring and modelling the trends of primary and secondary air pollution precursors: The case of the state of Kuwait. International Journal of Chemical Engineering, 2010.

Ettouney, R. S., Zaki, J. G., El-Rifai, M. A., & Ettouney, H. M. (2010). An assessment of the air pollution data from two monitoring stations in Kuwait. Toxicological and Environ Chemistry, 92(4), 655-668.

Ettouney, H., Al-Haddad, A., & Saqer, S. (2012, January). Daily and seasonal changes of air pollution in Kuwait. In Proceedings of World Academy of Science, Engineering and Technology (No. 61). World Academy of Science, Engineering and Technology.

Alenezi, R., Al-Anzi, B., Abusam, A., & Ashfaque, A. (2012). Seasonal influence on the ambient air quality in Al Jahra City for year 2010. Journal of Environmental Protection, 3(12), 1711-1718.

Al-Harbi, M. (2014). Assessment of air quality in two different urban localities. International Journal of Environmental Research, 8(1), 15-26.

Ettouney, H., Al-Haddad, A., & Saqer, S. (2012, January). Daily and seasonal changes of air pollution in Kuwait. In Proceedings of World Academy of Science, Engineering and Technology (No. 61). World Academy of Science, Engineering and Technology.

Abdul‐Wahab, S., Bouhamra, W., Ettouney, H., Sowerby, B., & Crittenden, B. D. (2000). Analysis of air pollution at Shuaiba Industrial Area in Kuwait. Toxicological & Environmental Chemistry, 78(3-4), 213-232.

Al-Salem, S. M., & Khan, A. R. (2008). Comparative assessment of ambient air quality in two urban areas adjacent to petroleum downstream/upstream facilities in Kuwait. Brazilian Journal of Chemical Engineering, 25(4), 683-696.

Al-Salem, S. M., Al-Fadhleeb, A. A., & Khan, A. R. (2009). Ambient air quality assessment of Al-Mansoriah residential area in the state of Kuwait. The Journal of Engineering Research [TJER], 6(2), 52-63.

Ma, J., Yu, Z., Qu, Y., Xu, J., & Cao, Y. (2020). Application of the XGBoost machine learning method in PM2. 5 prediction: A case study of Shanghai. Aerosol and Air Quality Research, 20(1), 128-138.

Harishkumar, K. S., Yogesh, K. M., & Gad, I. (2020). Forecasting air pollution particulate matter (PM2. 5) using machine learning regression models. Procedia Computer Science, 171, 2057-2066.

Rybarczyk, Y., & Zalakeviciute, R. (2018). Machine learning approaches for outdoor air quality modelling: A systematic review. Applied Sciences, 8(12), 2570.

Dobrea, M., Bădicu, A., Barbu, M., Subea, O., Bălănescu, M., Suciu, G., ... & Dobre, C. (2020, October). Machine Learning algorithms for air pollutants forecasting. In 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME) (pp. 109-113). IEEE.

Boonphun, J., Kaisornsawad, C., & Wongchaisuwat, P. (2019). Machine learning algorithms for predicting air pollutants. In E3S Web of Conferences (Vol. 120, p. 03004). EDP Sciences.

Czernecki, B., Marosz, M., & Jędruszkiewicz, J. (2021). Assessment of machine learning algorithms in short-term forecasting of pm10 and pm2. 5 concentrations in selected polish agglomerations. Aerosol and Air Quality Research, 21(7), 200586.

Appendix

```r
#### Import the relevant libraries
library(readxl)
library(openair)
library(data.table)
library(latticeExtra)
library(lubridate)
df <- read_excel("C:/Users/Ahmed.Ewis/Data.xlsx")
#colnames(df2)[1] <- "Sr" # Changing Column name of 1st Column. In the original excel sheet its set as Empty
colnames(df)[2] <- "date" # Changing MeasurementDateTime column name to date to make it compatible with summaryPlot
setDT(df) # COnverting your dataframe to Data Table
df <- df[!is.na(date)] # Removing all records where there is a Sr# but no Date and other features
df <- df[,-1]
str(df)
```

```r
##### Data Analysis
summaryPlot(df,selectByDate(df,year = (2015)) )
corPlot(df)
##############
scatterPlot(df, x = "NO2", y = "O3", z = "TH", linear = TRUE)

scatterPlot(df, x = "NO2", y = "O3", z = "RH", linear = TRUE)
polarPlot(df, pollutant = "NO2", statistic = "max", limits = c(0, 0.4))
polarPlot(df, pollutant = "O3 ppb", statistic = "max")
polarFreq(df, pollutant = "NO2", statistic = "max", type = "season", layout =c(4,1))
polarFreq(df, pollutant = "O3 ppb", statistic = "max", type = "season", layout =c(4,1))
percentileRose(df, type = c("season", "daylight"), pollutant = "NO2",
               col = "Set3", mean.col = "black")
percentileRose(df, type = c("season", "daylight"), pollutant = "O3 ppb",
               col = "Set3", mean.col = "black")
corPlot(df)
### THC VS NO2
timeVariation(df, pollutant = c("O3","NO2", "THC"), normalise = TRUE, ci = FALSE, statistic = "mean")
trendLevel(df, x = "NO2", y = "hour", pollutant = "THC", border = "white",
           n.levels = 10, statistic = "max", limits = c(0, 25))
trendLevel(df, x = "THC", y = "hour", pollutant = "NO2",
           border = "white", n.levels = 5, statistic = "max",
           limits = c(0, 0.18), type = "wd")

TheilSen(df, pollutant = "THC", type = "NO2", deseason = TRUE,
         ylab = "THC (ppm)", avg.time = "month", statistic = "mean", alpha = 0.05)

subdata_1 <- subset(df, NO2 < 0.014)
TheilSen(subdata_1, pollutant = "THC", type = "NO2", deseason = TRUE,
         ylab = "THC (ppm)", statistic = "frequency")
subdata_2 <- subset(df,NO2 > 0.007)
subdata_3 <- subset(subdata_2, NO2 < 0.01)
scatterPlot(subdata_3, x ="THC", y = "NO2", z = "RH", linear = TRUE, log.x = TRUE, log.y =TRUE)
polarPlot(df, pollutant = "THC", statistic = "max", limits = c(0, 25))
```

```r
### THC VS O3
scatterPlot(df, x = "O3", y = "THC", z = "TH", linear = TRUE)
trendLevel(df, x= "THC",pollutant = "O3", y = "hour", border = "white",
           cols = "jet", statistic = "max", type = "wd")

### THC VS SO2
scatterPlot(df, x = "SO2", y = "NO2", z = "RH", linear = TRUE, ci = FALSE)
polarPlot(df, pollutant = "SO2", statistic = "max", limits = c(0,0.322))


smoothTrend(df, pollutant = c("NO2", "SO2", "O3"), ylab = "concentration (ppb)",
            main = "Max", statistic = "max", ci = FALSE)
trendLevel(df, x= "SO2",pollutant = "NO2", y = "hour", border = "white",
           statistic = "max", type = "wd")
trendLevel(df, x= "THC",pollutant = "SO2", y = "hour", border = "white",
           statistic = "max", type = "wd")
trendLevel(df, x= "NO2",pollutant = "SO2", y = "hour", border = "white",
           statistic = "max", type = "wd")

trendLevel(df, x= "O3",pollutant = "SO2", y = "hour", border = "white",
           statistic = "max", type = "wd")
timePlot(df2, pollutant = c("NO2 ppb", "O3 ppb"), y.relation = "free")
timePlot(df2, pollutant = c("SO2", "PM10"), y.relation = "free")


episode_NO2 <- selectRunning(df2, pollutant = "NO2 ppb", threshold = 100, run.len = 1)
nrow(episode_NO2)
episode_O3 <- selectRunning(df2, pollutant = "O3 ppb", threshold = 70, run.len =1)
nrow(episode_O3)

#so2
episode_so2 <- selectRunning(df2, pollutant = "so2 ppb", threshold = 75, run.len =1)
nrow(episode_so2)
episode_thc<- selectRunning(df2, pollutant = "THC ppb", threshold = 1200, run.len =1)
nrow(episode_thc)
timePlot(selectByDate(df2, year = 2015,  month =("nov")),
         pollutant = c("NO2"), avg.time = "hour",
         y.relation = "free",windflow = list(col = "pink", lwd
                                             = 2, scale = 0.1))

timeProp(selectByDate(df2, year = 2014),
         pollutant = "NO2 ppb", avg.time = "hour",
         proportion = "wd", date.breaks = 10, key.position = "top",
         key.columns = 8, ylab = "NO2 (ppb)")
```

```r
scatterPlot(df2, x = "THC", y = "RH", z= "SO2", linear = TRUE)

scatterPlot(selectByDate(df2, start = "10/11/2014", end = "12/11/2014"),
            x = "date", y = "NO2", z = "O3",
            col = "increment",
            windflow = list(scale = 0.15),
            key.footer = "o3\n (ppm)", main = NULL, ylab = "no2 (ppm)")
episode_no2_15 <- selectRunning(df2, pollutant = "NO2 ppb", threshold = 100, run.len = 1)
nrow(episode_no2_15)
episode_O3_15 <- selectRunning(df2, pollutant = "O3 ppb", threshold = 70, run.len = 1)
nrow(episode_O3_15)
episode_O3_152 <- selectRunning(df2, pollutant = "ws", threshold = 8, run.len = 1)
nrow(episode_O3_152)
polarPlot(df2, pollutant = "NO2 ppb", statistic = "max")
polarPlot(df2, pollutant = "O3 ppb", statistic = "max")
polarFreq(df2, pollutant = "NO2 ppb", statistic = "max", type = "season", layout =c(4,1))
polarFreq(df2, pollutant = "O3 ppb", statistic = "max", type = "season", layout =c(4,1))
polarPlot(df2, pollutant = "NO2 ppb", statistic = "max", type = "season", layout =c(4,1))
polarPlot(df2, pollutant = "O3 ppb", statistic = "max", type = "season", layout =c(4,1))
pollutionRose(df2, pollutant = "NO2 ppb")
pollutionRose(df2, pollutant = "O3 ppb")
polarAnnulus(df2, pollutant = "O3 ppb", period = "weekday")
percentileRose(df2, pollutant = "NO2 ppb")
polarAnnulus(df2, pollutant = "O3 ppb", statistic = "max", period = "hour", main= "Hour")
```

```python
import matplotlib.pyplot as plt
import pandas as pd
import xgboost as xgb
import lightgbm as lgb
import numpy as np
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings('ignore')
```

```python
df = pd.read_excel('data.xlsx',sheet_name='Sheet1')
df
```

	MeasurementDateTime	WD-Hour	WS-Hour	Temp-Hour	SR-Hour	RH-Hour	CO2	PM10	SO2	H2S	NO	NOX	NO2	O3	CO	THC
0	2013-01-27 00:00:00	114	1.7	18.1	5	100	380	101.000000	0.001000	0.003	0.005	0.023	0.017000	2.600000e-02	0.58	2.170000
1	2013-01-27 01:00:00	151	0.4	17.8	3	89	410	101.000000	0.003000	0.017	0.065	0.151	0.086000	1.000000e-03	1.74	2.830000
2	2013-01-27 02:00:00	175	0.3	17.8	2	88	392	100.000000	0.002000	0.004	0.019	0.072	0.053000	2.000000e-03	0.95	3.770000
3	2013-01-27 03:00:00	264	0.8	16.8	2	92	403	92.000000	0.002000	0.014	0.051	0.123	0.072000	1.000000e-03	1.43	2.720000
4	2013-01-27 04:00:00	187	0.4	16.2	2	94	400	69.000000	0.002000	0.018	0.041	0.105	0.063000	1.000000e-03	1.32	2.520000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22774	2015-12-31 18:00:00	85	1.4	14.6	10	76	404	344.238675	0.006386	0.008	0.014	0.081	0.076199	1.747050e-03	2.73	1.464756
22775	2015-12-31 19:00:00	99	1.3	14.6	9	75	410	281.815817	0.006301	0.009	0.023	0.100	0.098153	1.108100e-04	2.81	1.903803
22776	2015-12-31 20:00:00	210	0.7	14.5	10	79	424	51.079508	0.006520	0.009	0.064	0.164	0.106758	8.930000e-06	2.96	2.039618
22777	2015-12-31 21:00:00	185	0.8	14.6	10	81	428	120.974071	0.006437	0.009	0.077	0.182	0.087641	1.000000e-08	3.25	1.754740
22778	2015-12-31 22:00:00	147	0.9	14.1	10	85	425	50.016188	0.006457	0.009	0.050	0.137	0.089011	0.000000e+00	3.45	1.818111

22779 rows × 16 columns

```
[ ] df.columns
```

```
Index(['MeasurementDateTime', 'WD-Hour', 'WS-Hour', 'Temp-Hour', 'SR-Hour', 'RH-Hour', 'CO2', 'PM10', 'SO2', 'H2S', 'NO', 'NOX', 'NO2', 'O3', 'CO', 'THC'], dtype='object')
```

```
[ ] cols = ['MeasurementDateTime','WD-Hour', 'WS-Hour', 'Temp-Hour', 'SR-Hour', 'RH-Hour','NO2', 'O3'
    ]
```

```
df = df[cols]
df
```

	MeasurementDateTime	WD-Hour	WS-Hour	Temp-Hour	SR-Hour	RH-Hour	NO2	O3
0	2013-01-27 00:00:00	114	1.7	18.1	5	100	0.017000	2.600000e-02
1	2013-01-27 01:00:00	151	0.4	17.8	3	89	0.086000	1.000000e-03
2	2013-01-27 02:00:00	175	0.3	17.8	2	88	0.053000	2.000000e-03
3	2013-01-27 03:00:00	264	0.8	16.8	2	92	0.072000	1.000000e-03
4	2013-01-27 04:00:00	187	0.4	16.2	2	94	0.063000	1.000000e-03
...	...	...	...	...	...	...	...	...
22774	2015-12-31 18:00:00	85	1.4	14.6	10	76	0.076199	1.747050e-03
22775	2015-12-31 19:00:00	99	1.3	14.6	9	75	0.098153	1.108100e-04
22776	2015-12-31 20:00:00	210	0.7	14.5	10	79	0.106758	8.930000e-06
22777	2015-12-31 21:00:00	185	0.8	14.6	10	81	0.087641	1.000000e-08
22778	2015-12-31 22:00:00	147	0.9	14.1	10	85	0.089011	0.000000e+00

22779 rows × 8 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22779 entries, 0 to 22778
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   MeasurementDateTime  22779 non-null  datetime64[ns]
 1   WD-Hour              22779 non-null  int64
 2   WS-Hour              22779 non-null  float64
 3   Temp-Hour            22779 non-null  float64
 4   SR-Hour              22779 non-null  int64
 5   RH-Hour              22779 non-null  int64
 6   NO2                  22779 non-null  float64
 7   O3                   22779 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(3)
memory usage: 1.4 MB
```

```
[ ] df['MeasurementDateTime'] = pd.to_datetime(df['MeasurementDateTime'], errors='coerce')
    df
```

	MeasurementDateTime	WD-Hour	WS-Hour	Temp-Hour	SR-Hour	RH-Hour	NO2	O3
0	2013-01-27 00:00:00	114	1.7	18.1	5	100	0.017000	2.600000e-02
1	2013-01-27 01:00:00	151	0.4	17.8	3	89	0.086000	1.000000e-03
2	2013-01-27 02:00:00	175	0.3	17.8	2	88	0.053000	2.000000e-03
3	2013-01-27 03:00:00	264	0.8	16.8	2	92	0.072000	1.000000e-03
4	2013-01-27 04:00:00	187	0.4	16.2	2	94	0.063000	1.000000e-03
...	...	...	...	...	...	...	...	...
22774	2015-12-31 18:00:00	85	1.4	14.6	10	76	0.076199	1.747050e-03
22775	2015-12-31 19:00:00	99	1.3	14.6	9	75	0.098153	1.108100e-04
22776	2015-12-31 20:00:00	210	0.7	14.5	10	79	0.106758	8.930000e-06
22777	2015-12-31 21:00:00	185	0.8	14.6	10	81	0.087641	1.000000e-08
22778	2015-12-31 22:00:00	147	0.9	14.1	10	85	0.089011	0.000000e+00

22779 rows × 8 columns

```
[ ] from datetime import datetime
    df['dayofweek'] = df['date'].dt.dayofweek
    df['quarter'] = df['date'].dt.quarter
    df['month'] = df['date'].dt.month
    df['year'] = df['date'].dt.year
    df['dayofyear'] = df['date'].dt.dayofyear
    df['dayofmonth'] = df['date'].dt.day
    df['datehour'] = df['date'].dt.hour
    df['weekofyear'] = df['date'].dt.weekofyear
    df
```

	MeasurementDateTime	WD-Hour	WS-Hour	Temp-Hour	SR-Hour	RH-Hour	NO2	O3	date	dayofweek	quarter	month	year	dayofyear	dayofmonth	datehour	weekofyear
0	2013-01-27 00:00:00	114	1.7	18.1	5	100	0.017000	2.600000e-02	2013-01-27 00:00:00	6	1	1	2013	27	27	0	4
1	2013-01-27 01:00:00	151	0.4	17.8	3	89	0.086000	1.000000e-03	2013-01-27 01:00:00	6	1	1	2013	27	27	1	4
2	2013-01-27 02:00:00	175	0.3	17.8	2	88	0.053000	2.000000e-03	2013-01-27 02:00:00	6	1	1	2013	27	27	2	4
3	2013-01-27 03:00:00	264	0.8	16.8	2	92	0.072000	1.000000e-03	2013-01-27 03:00:00	6	1	1	2013	27	27	3	4
4	2013-01-27 04:00:00	187	0.4	16.2	2	94	0.063000	1.000000e-03	2013-01-27 04:00:00	6	1	1	2013	27	27	4	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22774	2015-12-31 18:00:00	85	1.4	14.6	10	76	0.076199	1.747050e-03	2015-12-31 18:00:00	3	4	12	2015	365	31	18	53
22775	2015-12-31 19:00:00	99	1.3	14.6	9	75	0.098153	1.108100e-04	2015-12-31 19:00:00	3	4	12	2015	365	31	19	53
22776	2015-12-31 20:00:00	210	0.7	14.5	10	79	0.106758	8.930000e-06	2015-12-31 20:00:00	3	4	12	2015	365	31	20	53
22777	2015-12-31 21:00:00	185	0.8	14.6	10	81	0.087641	1.000000e-08	2015-12-31 21:00:00	3	4	12	2015	365	31	21	53
22778	2015-12-31 22:00:00	147	0.9	14.1	10	85	0.089011	0.000000e+00	2015-12-31 22:00:00	3	4	12	2015	365	31	22	53

22779 rows × 17 columns

```
[ ] df = df.set_index('date').asfreq('h')
    df
```

	MeasurementDateTime	WD-Hour	WS-Hour	Temp-Hour	SR-Hour	RH-Hour	NO2	O3	dayofweek	quarter	month	year	dayofyear	dayofmonth	datehour	weekofyear
date																
2013-01-27 00:00:00	2013-01-27 00:00:00	114.0	1.7	18.1	5.0	100.0	0.017000	2.600000e-02	6.0	1.0	1.0	2013.0	27.0	27.0	0.0	4.0
2013-01-27 01:00:00	2013-01-27 01:00:00	151.0	0.4	17.8	3.0	89.0	0.086000	1.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	1.0	4.0
2013-01-27 02:00:00	2013-01-27 02:00:00	175.0	0.3	17.8	2.0	88.0	0.053000	2.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	2.0	4.0
2013-01-27 03:00:00	2013-01-27 03:00:00	264.0	0.8	16.8	2.0	92.0	0.072000	1.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	3.0	4.0
2013-01-27 04:00:00	2013-01-27 04:00:00	187.0	0.4	16.2	2.0	94.0	0.063000	1.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	4.0	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2015-12-31 18:00:00	2015-12-31 18:00:00	85.0	1.4	14.6	10.0	76.0	0.076199	1.747050e-03	3.0	4.0	12.0	2015.0	365.0	31.0	18.0	53.0
2015-12-31 19:00:00	2015-12-31 19:00:00	99.0	1.3	14.6	9.0	75.0	0.098153	1.108100e-04	3.0	4.0	12.0	2015.0	365.0	31.0	19.0	53.0
2015-12-31 20:00:00	2015-12-31 20:00:00	210.0	0.7	14.5	10.0	79.0	0.106758	8.930000e-06	3.0	4.0	12.0	2015.0	365.0	31.0	20.0	53.0
2015-12-31 21:00:00	2015-12-31 21:00:00	185.0	0.8	14.6	10.0	81.0	0.087641	1.000000e-08	3.0	4.0	12.0	2015.0	365.0	31.0	21.0	53.0
2015-12-31 22:00:00	2015-12-31 22:00:00	147.0	0.9	14.1	10.0	85.0	0.089011	0.000000e+00	3.0	4.0	12.0	2015.0	365.0	31.0	22.0	53.0

25655 rows × 16 columns

```
    df1= df.interpolate(method='ffill')
    df1
```

	MeasurementDateTime	WD-Hour	WS-Hour	Temp-Hour	SR-Hour	RH-Hour	NO2	O3	dayofweek	quarter	month	year	dayofyear	dayofmonth	datehour	weekofyear
date																
2013-01-27 00:00:00	2013-01-27 00:00:00	114.0	1.7	18.1	5.0	100.0	0.017000	2.600000e-02	6.0	1.0	1.0	2013.0	27.0	27.0	0.0	4.0
2013-01-27 01:00:00	2013-01-27 01:00:00	151.0	0.4	17.8	3.0	89.0	0.086000	1.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	1.0	4.0
2013-01-27 02:00:00	2013-01-27 02:00:00	175.0	0.3	17.8	2.0	88.0	0.053000	2.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	2.0	4.0
2013-01-27 03:00:00	2013-01-27 03:00:00	264.0	0.8	16.8	2.0	92.0	0.072000	1.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	3.0	4.0
2013-01-27 04:00:00	2013-01-27 04:00:00	187.0	0.4	16.2	2.0	94.0	0.063000	1.000000e-03	6.0	1.0	1.0	2013.0	27.0	27.0	4.0	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2015-12-31 18:00:00	2015-12-31 18:00:00	85.0	1.4	14.6	10.0	76.0	0.076199	1.747050e-03	3.0	4.0	12.0	2015.0	365.0	31.0	18.0	53.0
2015-12-31 19:00:00	2015-12-31 19:00:00	99.0	1.3	14.6	9.0	75.0	0.098153	1.108100e-04	3.0	4.0	12.0	2015.0	365.0	31.0	19.0	53.0
2015-12-31 20:00:00	2015-12-31 20:00:00	210.0	0.7	14.5	10.0	79.0	0.106758	8.930000e-06	3.0	4.0	12.0	2015.0	365.0	31.0	20.0	53.0
2015-12-31 21:00:00	2015-12-31 21:00:00	185.0	0.8	14.6	10.0	81.0	0.087641	1.000000e-08	3.0	4.0	12.0	2015.0	365.0	31.0	21.0	53.0
2015-12-31 22:00:00	2015-12-31 22:00:00	147.0	0.9	14.1	10.0	85.0	0.089011	0.000000e+00	3.0	4.0	12.0	2015.0	365.0	31.0	22.0	53.0

25655 rows × 16 columns

In [18]:

```
from pycaret.regression import *
exp_reg101 = setup(data = df1, target = 'O3', session_id=123)
```

	Description	Value
0	Session id	123
1	Target	O3
2	Target type	regression
3	Data shape	(25655, 15)
4	Train data shape	(17958, 15)
5	Test data shape	(7697, 15)
6	Numeric features	14
7	Preprocess	1
8	Imputation type	simple
9	Numeric imputation	mean
10	Categorical imputation	constant
11	Fold Generator	KFold
12	Fold Number	10
13	CPU Jobs	-1
14	Log Experiment	0
15	Experiment Name	reg-default-name
16	USI	9345

In [19]:

```
compare_models()
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	0.0043	0.0000	0.0063	0.8809	0.0061	4.3849	2.2550
xgboost	Extreme Gradient Boosting	0.0043	0.0000	0.0064	0.8787	0.0061	6.0682	0.4340
et	Extra Trees Regressor	0.0041	0.0000	0.0064	0.8777	0.0061	3.4533	0.9890
lightgbm	Light Gradient Boosting Machine	0.0048	0.0000	0.0070	0.8536	0.0067	9.2379	0.2320
rf	Random Forest Regressor	0.0045	0.0001	0.0071	0.8495	0.0068	3.9257	1.6330
gbr	Gradient Boosting Regressor	0.0062	0.0001	0.0089	0.7629	0.0086	15.3717	0.5790
dt	Decision Tree Regressor	0.0062	0.0001	0.0104	0.6787	0.0100	1.7337	0.0430
lr	Linear Regression	0.0084	0.0001	0.0115	0.6051	0.0110	37.2053	0.6290
lar	Least Angle Regression	0.0084	0.0001	0.0115	0.6051	0.0110	37.5542	0.0130
br	Bayesian Ridge	0.0084	0.0001	0.0115	0.6051	0.0110	37.2033	0.0150
ridge	Ridge Regression	0.0084	0.0001	0.0115	0.6035	0.0110	37.8741	0.0140
knn	K Neighbors Regressor	0.0084	0.0001	0.0118	0.5852	0.0114	48.5364	0.0410
ada	AdaBoost Regressor	0.0110	0.0002	0.0131	0.4864	0.0127	143.3527	0.3020
omp	Orthogonal Matching Pursuit	0.0118	0.0002	0.0151	0.3208	0.0146	159.7325	0.0110
en	Elastic Net	0.0119	0.0002	0.0151	0.3201	0.0146	177.8048	0.0100
lasso	Lasso Regression	0.0121	0.0002	0.0154	0.2932	0.0149	188.8234	0.0140
huber	Huber Regressor	0.0116	0.0003	0.0169	0.0927	0.0159	31.1456	0.2690
llar	Lasso Least Angle Regression	0.0147	0.0003	0.0183	-0.0009	0.0177	233.4750	0.0150
dummy	Dummy Regressor	0.0147	0.0003	0.0183	-0.0009	0.0177	233.4750	0.0120
par	Passive Aggressive Regressor	0.0430	0.0021	0.0458	-5.2891	0.0439	586.9311	0.0130

In [18]:
```
from xgboost import XGBRegressor

xgb = (XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=0.7,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.1, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=9, max_leaves=0, min_child_weight=1,
               monotone_constraints='()', n_estimators=290,
              n_jobs=-1, num_parallel_tree=1, predictor='auto', random_state=123,
              reg_alpha=0.05, reg_lambda=0.1))
```

In [26]:
```
train_X, test_X, train_y, test_y = train_test_split(X, y,
                      test_size = 0.2, random_state = 123)
```

In [27]:
```
xgb.fit(train_X, train_y)
```

Out[27]:
```
XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=0.7,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.1, max_bin=256, max_cat_to_onehot=4,
             max_delta_step=0, max_depth=9, max_leaves=0, min_child_weight=1,
             missing=nan, monotone_constraints='()', n_estimators=290,
             n_jobs=-1, num_parallel_tree=1, predictor='auto', random_state=123,
             reg_alpha=0.05, reg_lambda=0.1, ...)
```

In [28]:
```
pred = xgb.predict(test_X)
pred
```

Out[28]:
```
array([0.00045016, 0.02260762, 0.0020332 , ..., 0.03950962, 0.06227764,
       0.01274938], dtype=float32)
```

In [29]:
```
# RMSE Computation
from sklearn.metrics import mean_squared_error as MSE
rmse = np.sqrt(MSE(test_y, pred))
print("RMSE : % f" %(rmse))
```

```
RMSE :  0.005350
```

In [30]:
```
xgb.score(train_X, train_y)
```

Out[30]:
```
0.9756177779346994
```

In [31]:
```
xgb.score(test_X, test_y)

```

Out[31]:
```
0.918161859103899
```

In [32]:
```
df1.columns
```

Out[32]:
```
Index(['year', 'month', 'quarter', 'dayofyear', 'dayofmonth', 'weekofyear', 'dayofweek', 'datehour', 'WD_Hour', 'WS_Hour',
'Temp_Hour', 'SR_Hour', 'RH_Hour', 'NO2', 'O3'], dtype='object')
```

In [33]:
```
xgb.predict(test_X)
```

Out[33]:
```
array([0.00045016, 0.02260762, 0.0020332 , ..., 0.03950962, 0.06227764,
       0.01274938], dtype=float32)
```

In [34]:
```
test = [2015,8,3,214,2,31,6,15,118,2.8,48.4,718,18,0.031]
p = pd.DataFrame([test])
p

```

Out[34]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	2015	8	3	214	2	31	6	15	118	2.8	48.4	718	18	0.031

In [35]:
```
xgb.predict(p)
```

Out[35]:
```
array([0.14080931], dtype=float32)
```

In [36]:
```
#import pickle
#file_name = "xgb_reg_sklearn_updated.pkl"

# save
#pickle.dump(xgb, open(file_name, "wb"))
```

Author: Ahmed Ewis

Supervised By: Dr. Fahad Al Fadli and Dr. Nawaf Al Hajri

Kuwait University

Welcome All to the Ozone Predictor

O3 Predictor ML App in the North of Kuwait

Year

2015

Month

7

Quarter

2

Dayofyear

220

Dayofmonth

28

Weekofyear

33

Dayofweek

2

Datehour

22

WD-Hour

350

WS-Hour

5

Temp-Hour

45

SR-Hour

500

RH-Hour

15

NO2

0.001

Predict

The output is 0.06346364319324493

About

Biography

Born on the 28th of November 1995, Ahmed Ewis is an accomplished Data Scientist with over four years of experience in the field. Graduating with a Bachelor's degree in Chemical Engineering in 2018, Ahmed made a successful transition into data science. Currently serving as a Senior Data Scientist at Warba Bank, Ahmed brings a wealth of expertise to the table.

Ahmed holds several prestigious certifications, including Microsoft Data Scientist certification, Professional Machine Learning Engineer certification, and Google Cloud Engineer certification. These certifications reflect Ahmed's commitment to staying at the forefront of data science and machine learning.

With a versatile skill set, Ahmed is proficient in various programming languages, including Python, R, Scala, SQL, and MATLAB. This proficiency allows Ahmed to excel in data analysis, modeling, and decision-making.

In addition to his technical acumen, Ahmed possesses a strong foundation in Chemical Engineering, which adds a unique dimension to his problem-solving abilities. This diverse background enables Ahmed to approach data science challenges from a multidisciplinary perspective. (Resume attached below)



Ahmed Ewis - Sr.
Data Scientist update

الملخص

تصبح مسألة جودة الهواء قضية ملحة في مختلف المناطق حول العالم، بما في ذلك الكويت، حيث تُعرف الأنشطة الصناعية بإصدار الملوثات في الجو. هدفت هذه الدراسة إلى تقييم البيانات الساعية لجودة الهواء في المواقع الحضرية بالجهراء وسعد العبد الله من خلال فحص مستويات الملوثين ثاني أكسيد النيتروجين والأوزون. كان الهدف هو تحديد مستويات التلوث الساعية لهذه الملوثات وإقامة العلاقات بينها. ولهذا الغرض، للاستمرار في العلاقة بين R في لغة البرمجة openair تم تطبيق طرق إحصائية متقدمة باستخدام حزمة الأوزون وثاني أكسيد النيتروجين. تسلط الدراسة أيضًا الضوء على عدد مرات تجاوز الملوثات للمعايير المقررة من قِبل هيئة البيئة العامة الكويتية. أظهرت نتائج تحليل البيانات والاستكشاف نمطًا موسميًا واضحًا في مستويات تركيز ثاني أكسيد النيتروجين والأوزون. وُجد أن تراكيز ثاني أكسيد النيتروجين كانت أعلى باستمرار في موسم الشتاء مقارنة بموسم الصيف. بالمقابل، كانت تراكيز الأوزون أعلى باستمرار في موسم الصيف مقارنة بموسم الشتاء. جانب أساسي من البحث هو التصميم، الذي شمل تحليل تشخيصي لفحص أداء نماذج التعلم الآلي المستخدمة. تم إجراء تحليل مقارن لتحديد النماذج المناسبة الأكثر من التعلم الآلي ونماذج التعلم العميق لهذا النوع من البيانات، بما في ذلك تقنيات التعزيز، وتقنيات التجميع والشبكات العصبية. أظهرت نتائج الدراسة ارتباطًا سلبيًا قويًا بين ثاني أكسيد النيتروجين والأوزون. وُجد أن استخدام طريقة التعزيز، وتحديدًا خوارزمية كات بوست ، قد أظهر أعلى مستوى من الدقة لهذا النوع من البيانات، حيث تم تحقيق معامل التحديد (آر مربع) بقيمة ٠.٩١ على مجموعة البيانات الاختبارية بعد تونة المعلمات الهامة. توفر هذه النتيجة مستوى أعلى من الدقة للتنبؤ الساعي بالملوثات. كانت النتائج متماشية مع توصيات الخبراء الذين يقترحون استخدام تقنيات التعزيز للبيانات الجدولية والتعلم العميق لمجموعات بيانات أكبر أو أنواع مختلفة من المدخلات مثل الصور والنصوص والصوت. تُظهر نتائج الدراسة أهمية مراقبة وتحسين جودة الهواء في الكويت وتوفر أساسًا

للأبحاث المستقبلية في هذا المجال. كما تسهل التنبؤ وتتيح للسلطات اتخاذ الاحتياطات اللازمة وزيادة الوعي إذا احتاج الأمر. تقدم الرسالة فهمًا شاملًا للعلاقة بين ثاني أكسيد النيتروجين والأوزون وتعد مساهمة قيمة في ميدان تقييم جودة الهواء.

الكلمات المفتاحية: تلوث الهواء، جودة الهواء، ثاني أكسيد النيتروجين، الأوزون، الجهراء، سعد العبد الله، الكويت.

جامعة الكويت

تقييم التلوث الجوي والتحليل الحاسوبي لجودة الهواء في شمال الكويت

المقدمة من الطالبة:

أحمد عويس

أطروحة مقدمة لكلية الدراسات العليا لاستيفاء جزء من متطلبات درجة الماجستير في :

الهندسة الكيميائية

بإشراف:

د. فهد الفضلي

د. نواف الهاجري (المشرف المشارك)

الكويت

يناير/2024