****

**DEPARTMENT OF COMPUTER APPLICATIONS**

**NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI 620015**

**Project Report**

**Deepfake AI Detection Model**

**Master of Computer Applications**

**By**

**Ashay Jain (205121028)**

**BONAFIDE CERTIFICATE**

This is to certify that the project **“Air Pollution Detection System”** is a project work successfully done by

Ashay Jain (205121028)

in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications from the National Institute of Technology, Tiruchirappalli, during the academic year 2022-2023 (5th Semester – CA749 Mini Project Work).

Dr. SelvaKumar K. Prof. Dr. P.J.A. Alphonse

Project Guide Head of the Department

Project viva-voce held on …………………………….

**Internal Examiner** **External Examiner**

Acknowledgment

Every project, big or small, is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration, support, and guidance of all those people who have been instrumental in making this project successful.

We express our deep sense of gratitude to **Dr. G. Aghila**, Director, National Institute of Technology, Tiruchirappalli for giving us an opportunity to do this project.

I am grateful to **Dr. P.J.A. Alphonse**, Professor, and Head of the Department of Computer Applications, National Institute of Technology, Tiruchirappalli for providing the infrastructure and facilities to carry out the project.

I express my gratitude to my Project Guide **Dr.SelvaKumar K.**, Professor, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for his support and for arranging the project in a good schedule, and who assisted me in completing the project. I would like to thank him for duly evaluating my progress and evaluating me.

I express my sincere and heartfelt gratitude to **Dr. S. Balasundaram**, Mini Project Work Coordinator, Department of Computer Applications, National Institute of Technology, Tiruchirappalli. I am sincerely thankful for his constant support, care, guidance, and regular interaction throughout my project.

I express my sincere thanks to all the faculty members, and scholars of NIT Trichy for their critical advice and guidance to develop this project directly or indirectly.

Abstract

**Background:**

Air pollution has become a severe environmental issue. It is responsible for hundreds of fatalities each year and it poses a severe threat to human health and the environment. It leads to global warming and the greenhouse effect and it also causes respiratory problems like asthma, lung cancer, etc. It is essential to predict the quality of the air to regulate air pollution. The air quality index (AQI) is a measure of air quality that describes the level of air pollution. Predicting changes in air pollutant concentrations due to human and natural drivers are critical and challenging, particularly in areas with scant data inputs and high variability of parameters. This project builds an Air Quality Index (AQI) model using Machine Learning algorithms and techniques. It employs Machine Learning Algorithms such as Linear regression, LASSO regression, Ridge regression, Decision Tree Regressor, and Random Forest Regressor.

**Objective:**

The main objective of the thesis is to build and train models using machine learning algorithms and find out the most accurate model in predicting the AQI.

**Keywords: AQI (Air Quality Index), Air Quality Monitoring, Machine Learning, Regression, EDA (Exploratory Data Analysis).**

Table of Contents

Bonafide Certificate …………………………………………………………………………….………… I

Acknowledgement …………………………………………………………….……………………….…. II

Abstract ………………………………………………………………………………………………………… III

Table of Contents ……………………………………………………………….…………………….…… IV

Title of the Project ……………………………………………………………….…………………….… V

Chapter 1: Introduction ……………………………………………………………….………………… 1

Chapter 2: Literature Survey ……………………………………………………………….…………. 2-3

Chapter 3: Problem Statement and Experimentation ……………………………………… 4-5

Chapter 4: Dataset ………………………………………………………………………………………… 6

Chapter 5: Implementation Details ………….……………………………………………………… 7

1. Software ………………………………………………………………………………………… 7
2. Hardware ………………………………………………………………………………………. 7

Chapter 6: Project Working and Explanation …………………………………………………… 8-18

1. Data Extraction from website ……………………………………………………. 8
2. Extraction PM2.5 ……………………………………………………………………… 9-11
3. Exploratory Data Analysis …………………………………………………………. 11-13
4. Feature Importance …………………………………………………………………. 13-14
5. Data Split …………………………………………………………………………………. 14
6. Machine Learning Algorithms ……………………………………………………. 15-17
   1. Linear Regression …………………………………………………………... 15
   2. Ridge and Lasso Regression ……………………………………………. 15
   3. Decision Tree (DT) …………………………………………………………. 16
   4. Random Forest (RF) ………………………………………………………. 16-17
7. Performance metrics …………………………………………………………………. 17-18

Chapter 7: Results and Discussion ………………………………………………............................... 19

Chapter 8: Conclusion ……………………………………………………………….………………………... 20

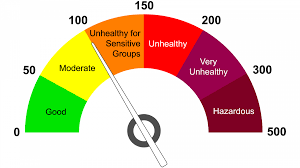
Chapter 9: Future Work ……………………………………………………………….……………………... 20

References ……………………………………………………………….………………………………………… VI

Appendix ……………………………………………………………….……………………………………. VII-XIII

Title of the Project

**Air Pollution Detection System**



1. Introduction

Air is a mixture of many gases and dust particles. It is the clear gas in which living things live and breathe. Air is a mixture of about 78% of nitrogen, 21% of oxygen, 0.9% of argon, 0.04% of carbon dioxide, and very small amounts of other gases. Air quality is measured with the Air Quality Index (PM 2.5). PM 2.5 is a fine particulate matter that is an air pollutant that is a concern for people’s health when levels in the air are high.

In this project, we are working on a dataset that contains different parameters like temperature winds, storms, and snow along with others. These parameters are really important for calculating the PM values. So, for this evaluation, we need to extract live data from different websites through the python programming language. We have extracted the data (independent features from the website **tutiempo.net** and a dependent feature, i.e., P.M. 2.5, from a paid API website **OpenWeatherMap**. The complete project is divided into different steps and each step contains the lifecycle of the project.

1. Literature Survey

**Anikender Kumar, PramilaGoyal (2011)** presented the study that forecasts the daily AQI value for the city Delhi, India using previous record of AQI and meteorological parameters with the help of Principal Component Regression (PCR) and Multiple Linear Regression Techniques. They perform the prediction of daily AQI of the year 2006 using previous records of the year 2000-2005 and different equations. After that this predicted value then compared with observed value of AQI of 2006 for the seasons summer, Monsoon, Post Monsoon and winter using Multiple Linear Regression Technique [1]. Principal Component Analysis is used to find the collinearity among the independent variables. The Principal components were used in Multiple Linear Regression to eliminate collinearity among the predictor variables and also reduce the number of predictors [1]. The Principal Component Regression gives the better performance for predicting the AQI in winter season than any other seasons. In this study only meteorological parameters were considered or used while forecasting the future AQI but they have not considered the ambient air pollutants that may cause the adverse health effects.

**Aditya C R (et al.2018)** employed the machine algorithms to detect and forecast the PM2.5 concentration level on the basis of dataset containing atmospheric conditions in a specific city. They also predicted the PM2.5 concentration level for a particular date [2]. First of all, they classify the air as polluted or not polluted by using Logistic Regression algorithm and then Auto Regression algorithm was used to predict the future value of PM2.5 depending upon previous records.

**HeidarMaleki (et al.2019)** predicted the hourly concentration values for the ambient air pollutants NO2, SO2, PM10, PM2.5, CO and O3 for the stations Naderi, Havashenasi, MohiteZist and Behdasht in Ahvaz, Iran which is the most polluted city in the world. They have also calculated and predicted Air Quality Index (AQI) and Air Quality Health Index (AQHI) for the four air quality monitoring stations in Ahvaz mentioned above. They used Artificial Neural Network (ANN) machine learning algorithm for the prediction of air pollutants concentration (hourly) and two air quality indices AQI and AQHI over the August 2009 to August 2010. Input to ANN algorithms involves the factors such as meteorological parameters, Air pollutants concentration, time and date.

**Ranjana Waman Gore et.al.in** have proposed an approach in which Naïve Bayes and J48 classification algorithms are used for analyzing the air quality levels. The accuracy of dataset by using Naïve Bayes was 86.66% and the accuracy with J48 decision tree algorithm was 91.99%. And author also justify that J48 algorithm gives more accurate results than Naïve Bayes algorithm.

**Bonny Paulose et.al.in** have proposed mainly focused on analysis of air quality of Delhi and also find the reason behind the pollutants that cause air pollution by using K-means clustering algorithm. And the author showed that Anand Vihar, R k Puramand, Punjabi Bagh are one of the mostly polluted regions.

**Ranjana Gore et.al.in** have proposed a method in which author used Random forest and multiclass classifier classification algorithms for analysis of air quality. The author also showed that multiclass classifier is superior than random forest.

**Mohamed Shakir et.al.in** have proposed a model for investigation of air pollution of Karnataka state. The author used the ZeroR algorithm for air pollutants analysis. And author also shows the dependencies and relationship between pollutants.

**Shweta Taneja et.al.in** have proposed an approach for predicting the air pollution in Delhi. The author used time series analysis techniques namely are Linear regression and multilayer perceptron for predicting air pollutants.

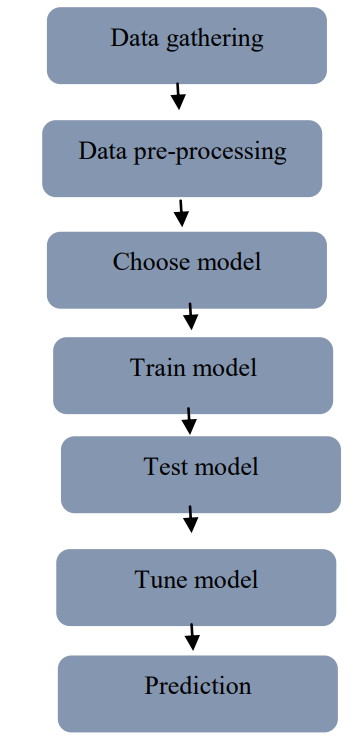
1. Problem Statement and Experimentation

Air Pollution Detection System is a project, that aims to provide the air quality index report of major cities of India. The Air Quality Index is an index that is used for reporting daily air quality. With the help of indexes, we know how clean or unhealthy our air is, and what associated health effects might be a concern. The Air Quality Index mainly focuses on the quality of air within our environment, as we can see most of the popular cities in India has the worst quality index just because of the increasing population and pollution, so our motive is to predict the AQI which is based on different factors like temperature pollutions and many more. Majorly AQI is calculated for four major air pollutants regulated by the Clean Air Act: particle pollution, ground-level ozone sulfur dioxide, and carbon monoxide. EPA has established national air quality standards to protect public health for each of these pollutants.

We’ve performed experimentation using Air quality data in India dataset and algorithms identified from the literature review.

The following stages were involved in experimentation.

* Collecting the dataset from the online resource Kaggle.
* Preprocessing the dataset.
* Build models using the preprocessed dataset and algorithms identified during the literature review.
* Testing the models.
* Evaluating the performance of the models by considering performance metrics.



Stages in Machine Learning

1. Dataset

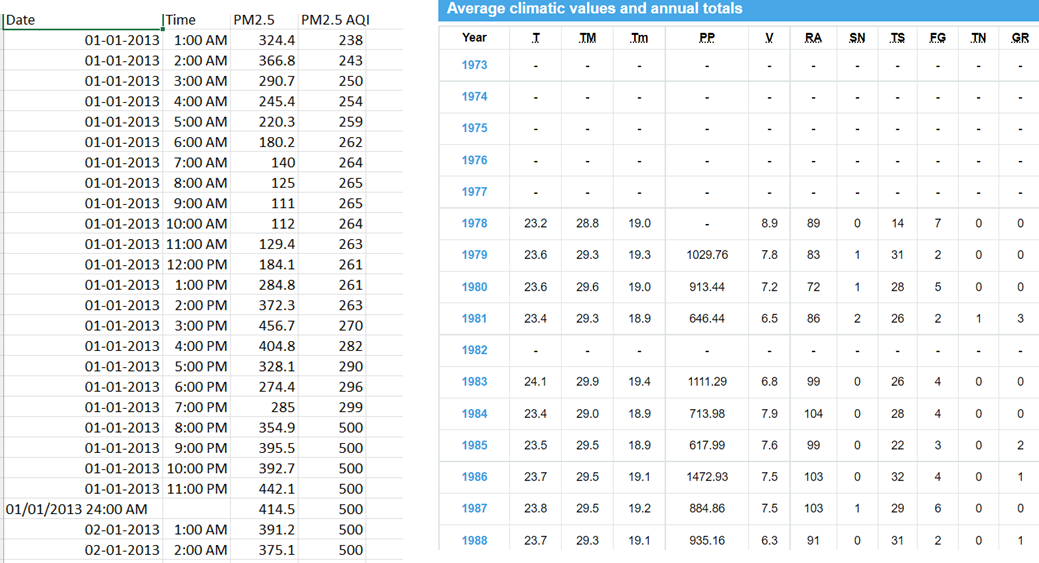


Figure 4. Dependent and Independent features of Dataset

**Dependent feature Independent features**

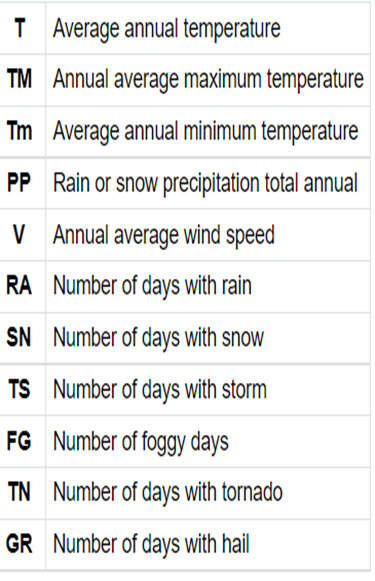


Figure 4. Abbreviations of Features

1. IMPLEMENTATION DETAILS

**Software**

To implement experimentation, we’ve used the following technologies. A brief description of used technologies along with versions is represented below.

* Python - Python is an interpreted, high level and object-oriented programming language. It is an open source.
* Anaconda navigator - It is a graphical user interface (GUI) that allows to launch applications easily and manage packages.
* Jupyter notebook - It is a web-based interactive computing platform. It helps in developing, documenting, and executing code.
* Pandas - It is a free open-source python library. It is mainly used for data analysis. It helps to perform various data manipulation operations.
* Numpy - It is a python library that is used for scientific computing in python. It is used to perform wide mathematical operations on data.
* Matplotlib - It is a Python package used to create static, animated, and interactive visualizations.
* Seaborn - It is a python library that is used for data visualization. It is based on the matplotlib library. It helps to make statistical graphics using python.
* Scikit-learn (sklearn) - It is a python library that is used for machine learning. It is largely written in python. It consists of many machine learning algorithms. It is best suitable for predictive analysis.

**Hardware**

* Hard Disk       1 TB
* Processor       Intel i5 10th gen
* RAM               8 GB

1. Project Working and Explanation

The complete life cycle of this project is divided into 5 challenges. Each of them will be explained in detail in the following step:

1. **Data Extraction from website**

So, in the very first step, we need to extract data from a website that is in HTML format and we need to convert that data into HTML file so that we can perform EDA process on that file. Data, which is shown in this picture contains parameters like temperature, snow level, winds storm, intensity, and many more but this data is available on a website.

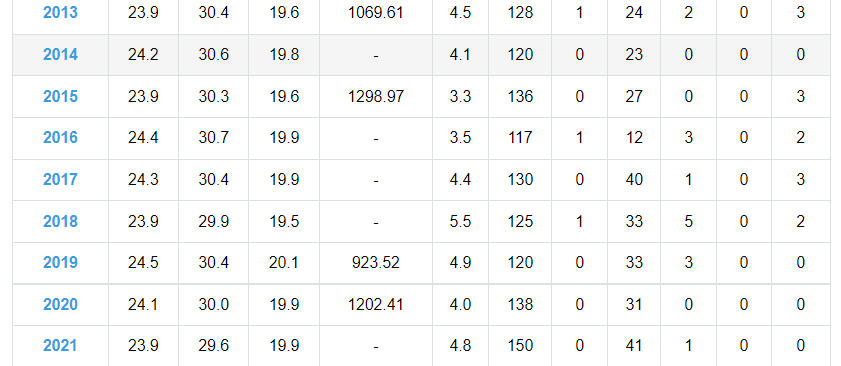


Figure 6.1.1 Data Extracted from website tutiempo.net of several years

So, as we can see that this data is available in table format on www.tutiempo.net but we need this data in an HTML file. This can happen in two ways. We can use API which can directly fetch the data from a particular website and convert that data into the desired format or we can use a python programming language with the help of Beautifulsoup we can convert HTML data into the desired format. [[Figure 1](#_Toc121923184)]

After compiling the code, the HTML file will create in Data>AQI Folder. Now we can easily extract HTML files from that folder and do an EDA process to extract useful information which will provide the required parameters.

1. **Extraction PM2.5**

PM2.5 refers to atmospheric particulate matter. PM has a diameter of 3% of human hair, which is about lesser than 2.5 micrometers. Since they are small and light in weight, fine particles tend to stay longer in the air compared to heavier particles which in turn increases the chances of human beings and animals inhaling them into their bodies. Some particles due to their minute size i.e., smaller than 2.5 micrometers can penetrate deep into the throat and nose and can also penetrate the lungs. In some cases, some may find their way into the circulatory system. The process which we are using here is similar to the previous one, we need to fetch PM2.5 values from 2013 to 2018 through API from a website that gives us this utility. Data that we get through the website is in CSV format here is a screenshot of the data.

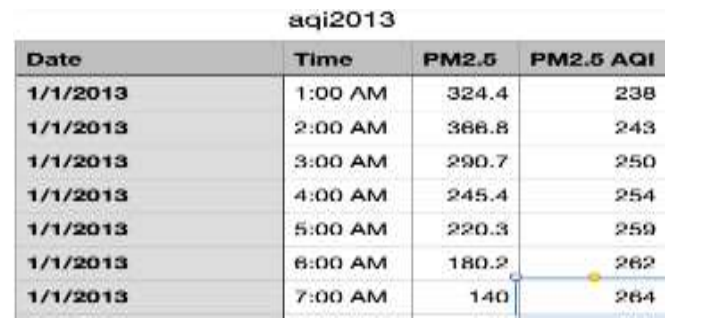


Figure 6.2.1 Data Collected from a paid API website OpenWeatherMap

Here we have PM2.5 value with respect to each hour for 5 years but in the previous file, the data is in day format so now we need to convert this data into day format this can be possible by using a python programming language. [[Figure 2](#_Toc121923185)]

After compiling, this python file we will get the PM data which is on the basis of per day. Here in the output file, we have constructed a plot map that shows the average dataset for each year.

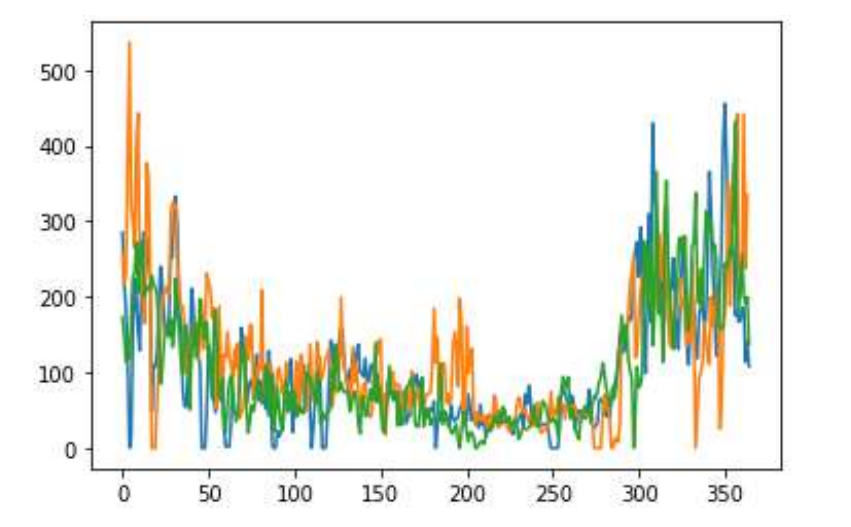


Figure 6.2.2 Plot map that shows the average dataset for each year

Now we need to concatenate this PM parameter to the previous file where we have required parameters but here is a problem in this dataset, this dataset required data cleaning as there are some unwanted characters present in this set so we need to eliminate that part in the cleaning process. [[Figure 3](#_Toc121923186)]

Now, we are converting HTML files to CSV format and after that, we are combining that file to a PM file which is already in CSV Format. By doing this we will get a Single output file in CSV format to apply the EDA process this can be happened by importing the Beautifulsoup library from Python in the build library stack. It is a very useful method to manipulate HTML data. Here we are using convert HTML data into the desired format. At the end of this process, we are creating a new CSV file named “Real\_combine.CSV” which contains all required parameters such as PM values, Temp, Storm, and wind velocity for each day and we have already removed all unwanted now we can easily apply all EDA process on this file. Here is the output of the final CSV File named **real\_combine.csv.** [[Figure 4](#_Toc121923187)]

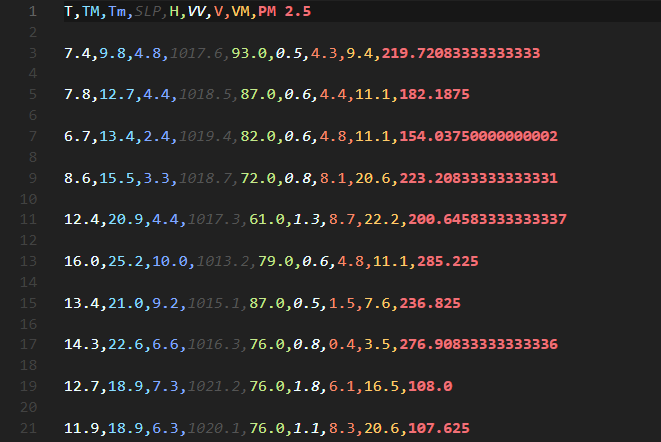


Figure 6.2.3 CSV file named “real\_combine.csv”

1. **Exploratory Data Analysis**

Here we are in our EDA process. As of now, we have Data to analyze, first, we need to find out the null value in the given data. Since we have a large dataset, we can’t do this by searching manually so we have created a heat map that shows negative value in the given dataset by applying a simple python programming module. [[Figure 5](#_Toc121923188)]

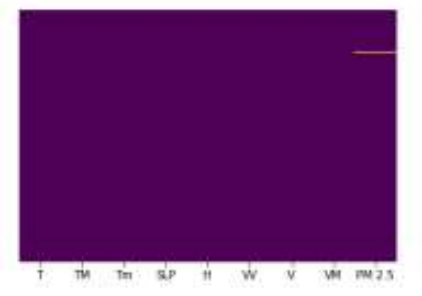


Figure 6..1 Heat Map containing NULL values in PM column

By visualizing the heat map we can clearly see that there is a null value in PM Column so now we need to eliminate that null value, by using the df.dropna() function we can do the same. Now we have done the basic data engineering on the dataset, all null values have been eliminated so far and also, we have done all data cleaning part now we need to divide our data into dependent and independent features. After the division of data features, we created a pair plot on features so that we can visualize the relationship between dependent and independent features.

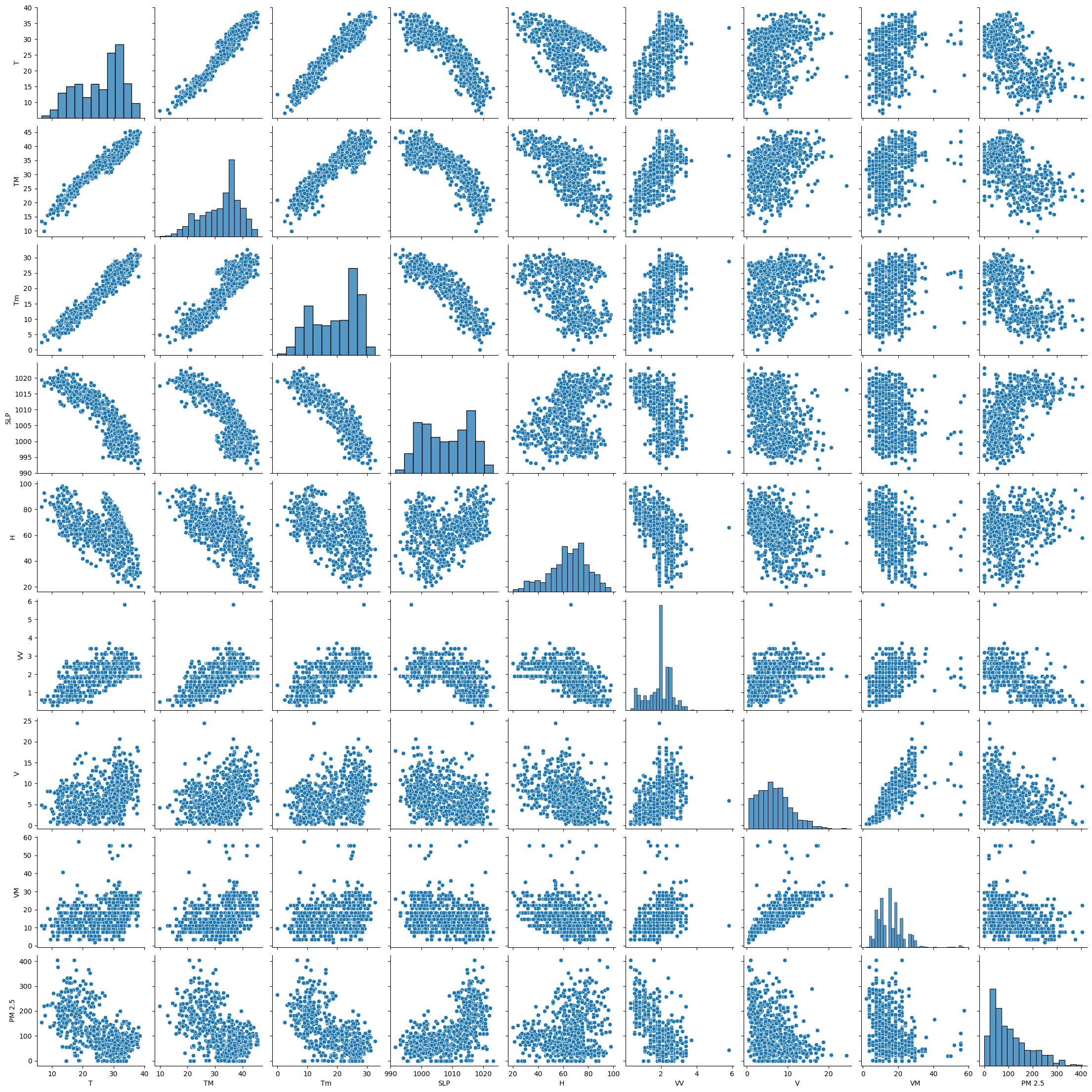


Figure 6.3.2 Pair plot that shows the relationship between dependent and independent features

We can also create a correlation graph between the feature, by doing this we will get to know a better relationship between them because only then we can apply a suitable machine learning algorithm.

It shows the positive and negative relationship between dependent and independent features of the dataset basically correlation states how the features are related to each other or the target variable in the given dataset. Correlation can be positive which shows the increase in one value of a feature increases the value of the target variable and can be negative which indicates an increase in one principle of feature results into decrease in the value of the target variable. Heatmap makes it easy to identify which features are most related to the target variable, we will plot a heatmap of correlated features using the seaborn library. [[Figure 6](#_Toc121923189)]

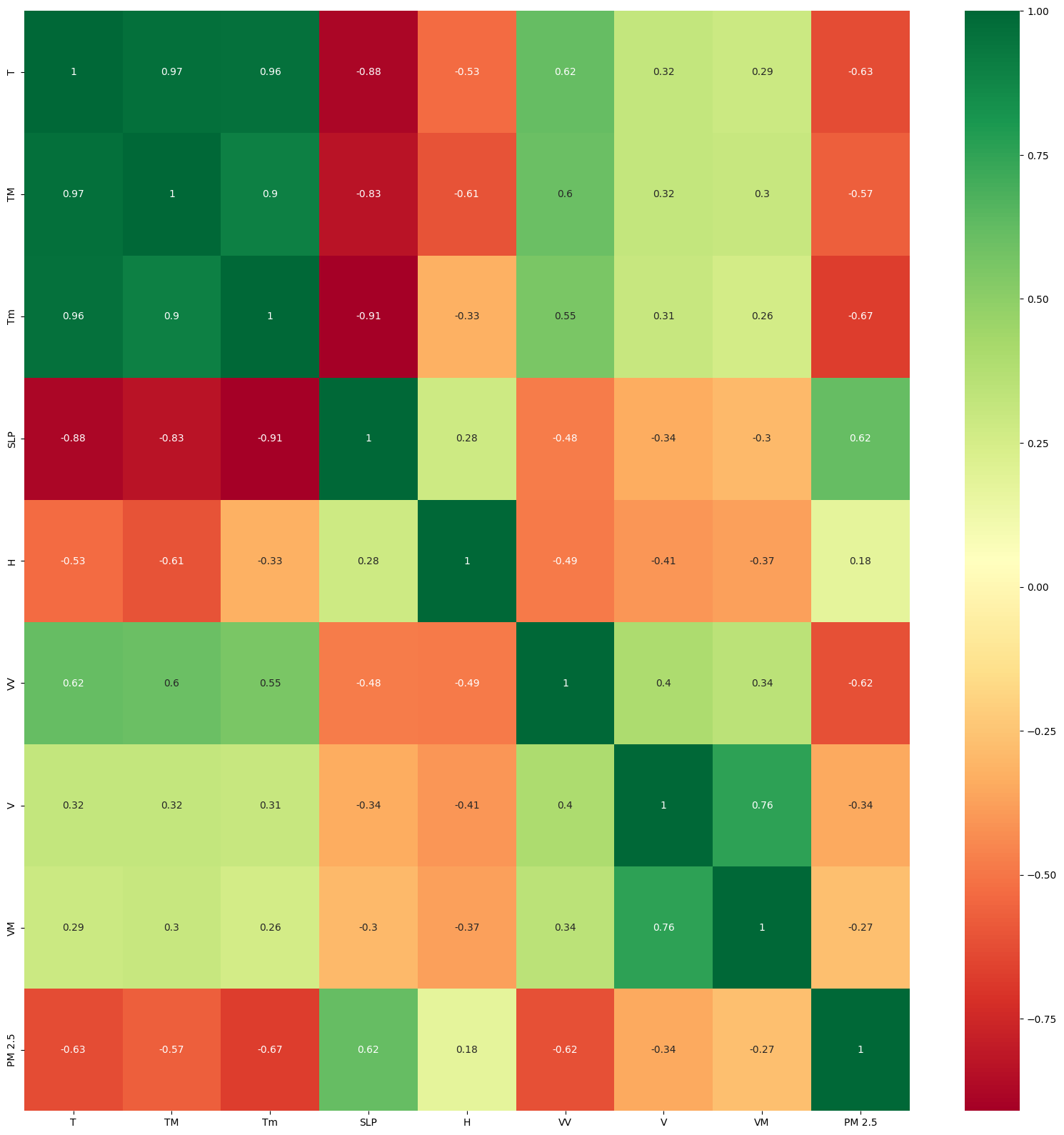


Figure 6.3.3 Heatmap that shows the relevant feature related to the target variable

1. **Feature Importance**

Feature importance is very important in terms of EDA because in the current dataset we don’t have thousands of features but in a real scenario it might be up to a thousand so we need to evaluate which features need to be analyzed so we have already created a correlation map from where we ca easily extract the required one. We can get the importance of features by using the “feature importance property”, it gives you a score for each feature, the higher the score will get more important but for that first, we need to import the required library.

This feature is a build function in Tree-based regression, we are using the same technique which extracts the top 10 features in the given dataset. [[Figure 7](#_Toc121923190)]

It is a visitation graph that indicates the top required features of the given dataset, here we are taken only 5 features as we need only them but in real-world problems, we can extract as much as we can depending upon the dataset requirement. [[Figure 8](#_Toc121923191)]

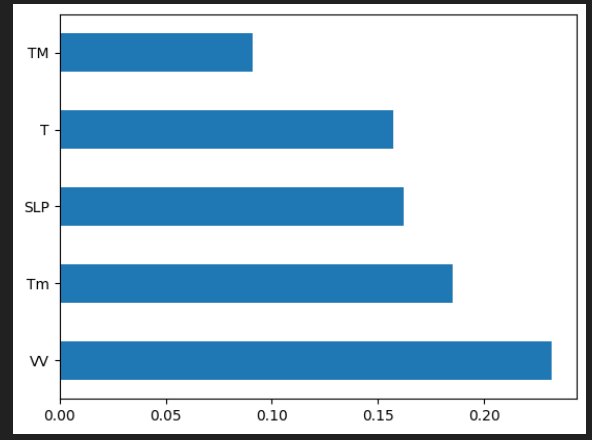
****

Figure 6.4.1 visitation graph that indicates the top required features of the given dataset

1. **Data Split**

Basically, the Machine learning algorithm works in two different stages so we usually split the data to apply the machine learning algorithm. The whole data is divided into two different parts, which are generally 80 and 20 percent.

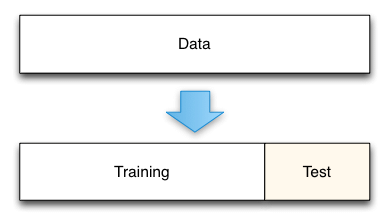


Figure 6.5.1 Splitting the dataset into training and test data

The same approach we are using here by diving out data in the Test and Train dataset. Here is the python code which splits out data into the desired forms.

By using the train test split in we have split our dataset now our task is to do training our dataset so that we can achieve our desired output but to train our dataset we need to apply a machine learning algorithm on our datasets so this is our final and most important task to perform. [[Figure 9](#_Toc121923192)]

1. **Machine Learning Algorithms**
2. **Linear Regression**

Linear regression is a machine learning approach that is used to establish a relationship between the independent variable and the dependent variable. Here, the variable which is predicted is called the dependent variable, and the variable that is used to predict the dependent variable is called the independent variable. Linear regression is said to give the simplest form of the regression function as a linear equation of variables. The interpretation of the parameters is easy due to the linear form of the regression function. The linear regression model gives a sloped straight line describing the relationship within the variables. [[Figure 10](#_Toc121923193)]



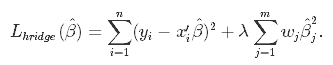
1. **Ridge and Lasso Regression**

This is a regularization technique used in feature selection using a Shrinkage method also referred to as the **penalized regression method**. Lasso is short for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator, which is used both for regularization and model selection. If a model uses the **L1 regularization** technique, then it is called lasso regression. [[Figure 11](#_Toc121923194)]



Similar to the lasso regression, ridge regression puts a similar constraint on the coefficients by introducing a penalty factor. However, while lasso regression takes the magnitude of the coefficients, ridge regression takes the square.

Ridge regression is also referred to as **L2 Regularization. [**[Figure 12](#_Toc121923195)]



1. **Decision Tree (DT):**

The decision tree is considered among the most effective and common algorithms for classification and future prediction. DT is a conceptual tree-alike model, where each internal node represents a feature that best split the data into subsets using statistical measures such as information gain and gain ratio, the process of splitting data is a recursive process until reaching the leaf (normally one of class labels). In machine learning, DT is one of the supervised learning algorithms. [[Figure 13](#_Toc121923196)]

It can be clearly seen that the Decision Tree suffers from the problem of overfitting.

1. **Random Forest (RF)**

It is one of the most popular machine-learning algorithms for regression and classification tasks. RF creates a number of decision trees called forest trees to enhance the prediction process and produce higher accuracy. Building an RF tree is similar to a decision tree (DT) using information gain or other measures. Since RF is a set of DTs; each tree obtains a certain output and RF will choose the majority output produced by DTs or the mean in case of a regression problem.



Figure 6.6.1 Working of Random Forest

**WHY RANDOM FOREST?**

**No overfitting**: Overfitting means we have fit the data so close in the data sample and then we pick up on all the weird parts and instead of predicting the overall data we are predicting the weird stuff. The use of multiple trees reduces the risk of overfitting. Here training time is less.

**High accuracy**: Runs efficiently on large databases. For large data, it predicts highly accurate predictions. In today’s world of big data, this is very important and this is probably where it really shines. Therefore, Random Forest comes in.

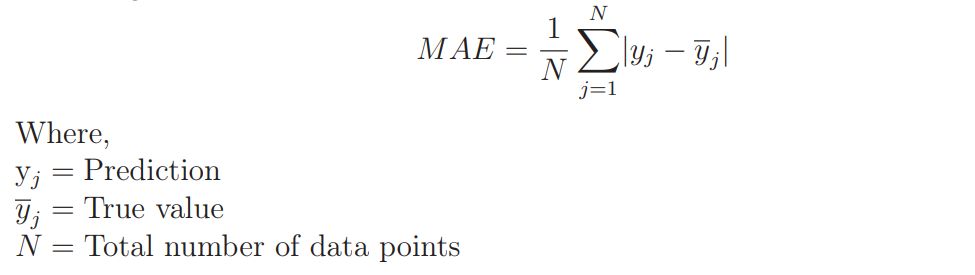
**Estimate missing data**: In today’s world, data is messy, so we have a Random Forest it can maintain accuracy when a large proportion is missing. What that means is when you have data that comes in from five or six different areas. And maybe they took one set of statistics in one area and then took a slightly different set of statistics in the other so they have some of the same shared data but one is missing. Then it will build trees and can do a very good job of guessing which one fits better even though it's missing from the data.

So now we have applied a random forest algorithm to our given datasets to get the highest accuracy on given true values. Now after applying the Random Forest regressor to the given dataset, we need to evaluate the model. [[Figure 14](#_Toc121923197)]

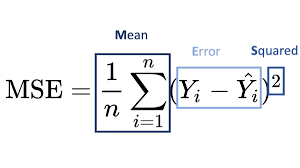
1. **Performance metrics**

To analyze the performance of a machine learning model we need some metrics. These metrics are statistical criteria that can be used to measure and monitor the performance of a model. As our thesis deals with prediction, we’ve considered MAE and RMSE as performance metrics.

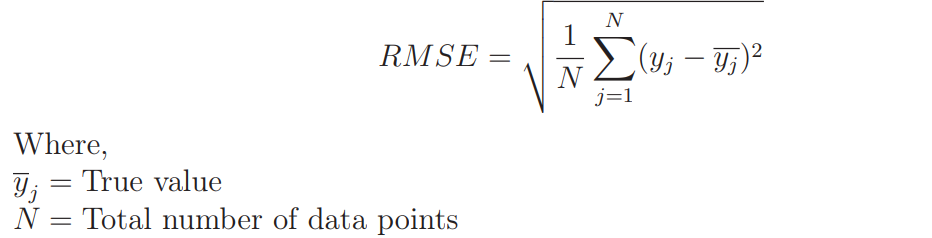
**Mean absolute error (MAE):** MAE is the arithmetic average of the difference between the ground truth and the predicted values. It can also be defined as a measure of errors between paired observations expressing the same phenomenon. It tells us how far the predictions differed from the actual result.



**Mean square error (MSE):** MSE is the mean of the squared errors, itis more popular than MAE because MSE “punishes” larger errors which leads to being useful in the real world.



**Root mean square error (RMSE):** RMSE is the square root of the average of the squared difference between the target value and the value predicted by the model. It is the square root of the mean square error (MSE). The implementation is very much similar to MSE.



The machine learning models are validated by comparing the performance metrics. The lower the MAE and RMSE, the machine learning model performs better.

1. RESULTS AND DISCUSSION

**Coefficient of Determination(R^2)**

|  |  |  |
| --- | --- | --- |
| **Algorithms** | **Training Data** | **Test Data** |
| Linear Regressor | 0.5770502192806163 | 0.5895061006429952 |
| Ridge Regressor | 0.5749714667713393 | 0.5884779534302574 |
| Lasso Regressor | 0.5668662577766788 | 0.5723583735268779 |
| Decision Tree Regressor | 1.0 | 0.2120786882303698 |
| Random Forest Regressor | 0.974158792253491 | 0.7875072311303447 |

**MAE, MSE, RMSE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithms** | **MAE** | **MSE** | **RMSE** |
| Linear Regressor | 36.28452300110422 | 2692.181402925697 | 51.8862351970703 |
| Ridge Regressor | 36.504924239827794 | 2698.92439913085 | 51.95117322189028 |
| Lasso Regressor | 37.71212170403514 | 2804.6429818109573 | 52.958880103444 |
| Decision Tree Regressor | 48.28916666666666 | 4683.957703569122 | 68.4394455235365 |
| Decision Tree Regressor  (Hyperparameter Tuning) | 39.33415879891425 | 3051.79689456849 | 55.243071009570876 |
| Random Forest Regressor | 23.823278201219516 | 1522.255992876853 | 39.01609914992596 |
| Random Forest Regressor (Hyperparameter Tuning) | 24.572813418347607 | 1645.7263253027436 | 40.5675526166263 |

1. CONCLUSION

The reduction of air pollutant levels is one of the most important tasks for protecting both human health and the environment. People must know what the level of pollution in their surroundings is and takes a step towards fighting against it. The conclusions from various studies show that the use of machine learning models helps to detect the quality of air and predict the level of AQI in the future and classify them into different classes. The proposed system will help common people as well as those in the meteorological department to detect and predict pollution levels and take the necessary action following that. Also, this will help the public set up a data source for small localities which are usually left out in comparison to large cities.

1. Future Work

* In this study the data used was static which means the data will be fixed and it remains the same after it’s collected. However, the government updates the data hourly. So, we can use real-time data analysis using the cloud to obtain better outcomes for greater performance as the data updates for every particular interval of time.
* We can further ensemble two or more machine learning algorithms and process large data to get more accurate results.
* Implementing Artificial Neural Networks for better accuracy.
* Deploying the model on localhost

1. References
2. Anikender Kumar, PramilaGoyal, “Forecasting of air quality in Delhi using principal component regression technique”, Atmospheric Pollution Research, 2 (2011) 436-444.
3. Aditya C R, Chandana R Deshmukh, Nayana D K, Praveen Gandhi Vidyavastu, “Detection and Prediction of Air Pollution using Machine Learning Models”, International Journal of Engineering Trends and Technology (IJETT) – volume 59 Issue 4 – May 2018
4. Heidar Malek, Armin Sorooshian, Gholamreza Goudarzi, Zeynab Baboli, Yaser Tahmasebi Birgani, Mojtaba Rahmati, “Air pollution prediction by using an artifcial neural network model”, Clean Technologies and Environmental Policy, (2019) 21:1341–1352.
5. Carbajal-Hernández, José Juan "Assessment and prediction of air quality using fuzzy logic and autoregressive models." Atmospheric Environment 60 (2012): 37-50.
6. Kumar, Anikender and P. Goyal, “ Forcasting of daily air quality index in Delhi”, Science of th Total Environment 409, no. 24(2011): 5517-5523.
7. Singh Kunwar P., et al. “Linear and nonlinear modelling approaches for urban air quality prediction “, Science of the Total Environment 426(2012):244-255
8. Bhanarkar, A. D., et al, “Assessment of contribution of SO2 and NO2 from different sources in Jamshedpur region, India, “Atmospheric Environment 39.40(2005):7745-India." Atmospheric Environment 39.40 (2005): 7745-7760.
9. https://towardsdatascience.com/
10. https://www.youtube.com/@krishnaik06

[11]. https://www.geeksforgeeks.org/machine-learning/

[12]. Hands-On Python for Finance: A Practical Guide to Implementing Financial Book by Krish Naik

[13]. https://www.aqi.in/blog/aqi/

[14]. https://openweathermap.org/

[15]. http://www.tutiempo.net/

1. Appendix

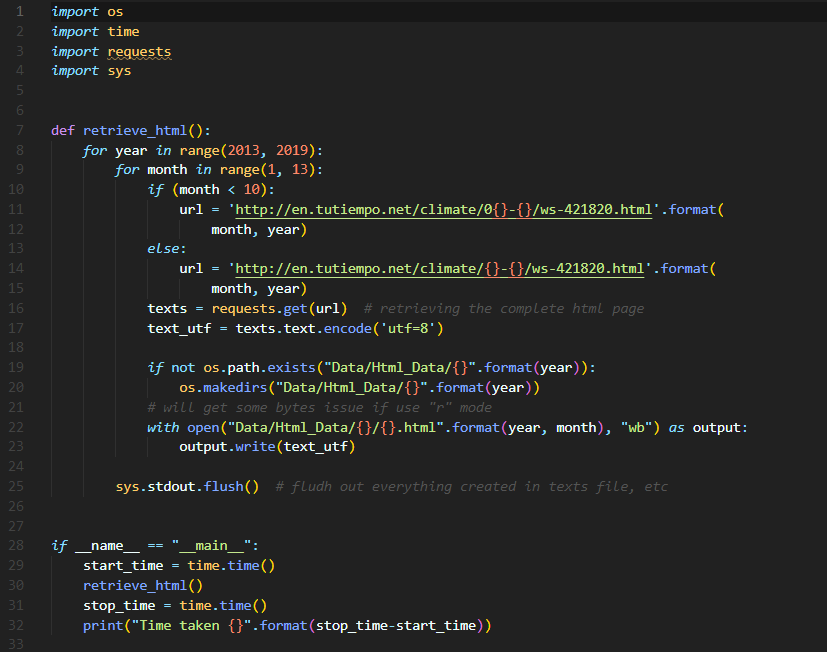


Figure 1 Python code which is used to fetch data from the website tutiempo.net



Figure 2 Python function to average the data of each year

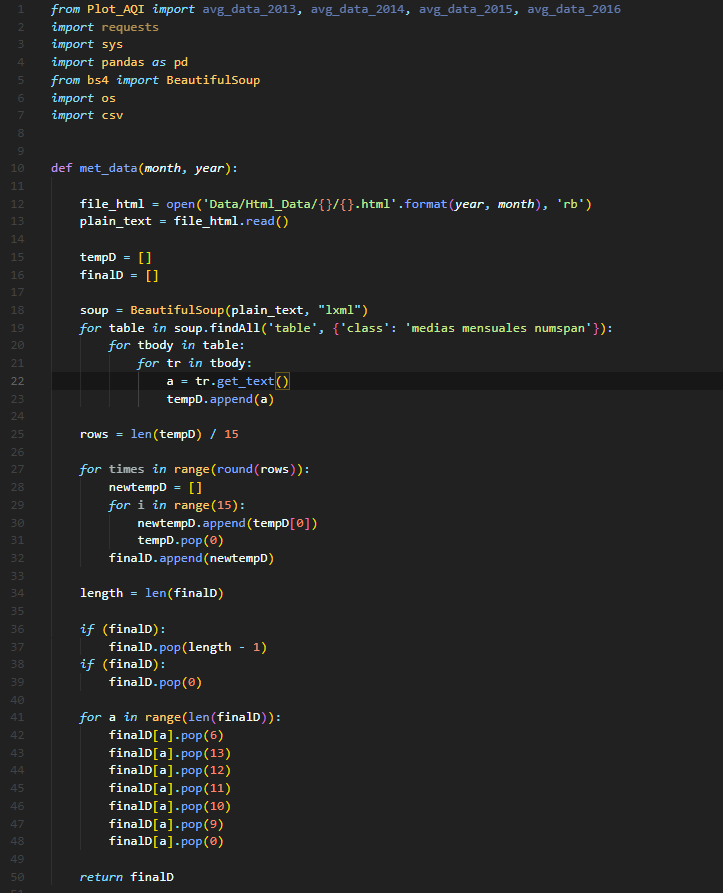


Figure 3 Python code to convert the HTML data to CSV format and clean it

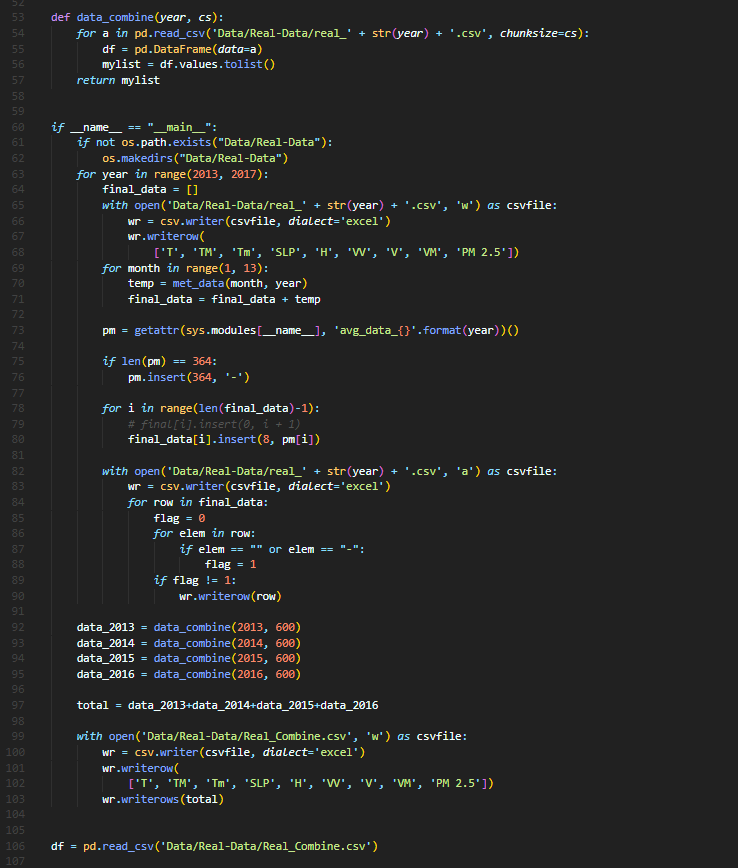


Figure 4 Code to combine the independent and dependent features and get the final CSV file

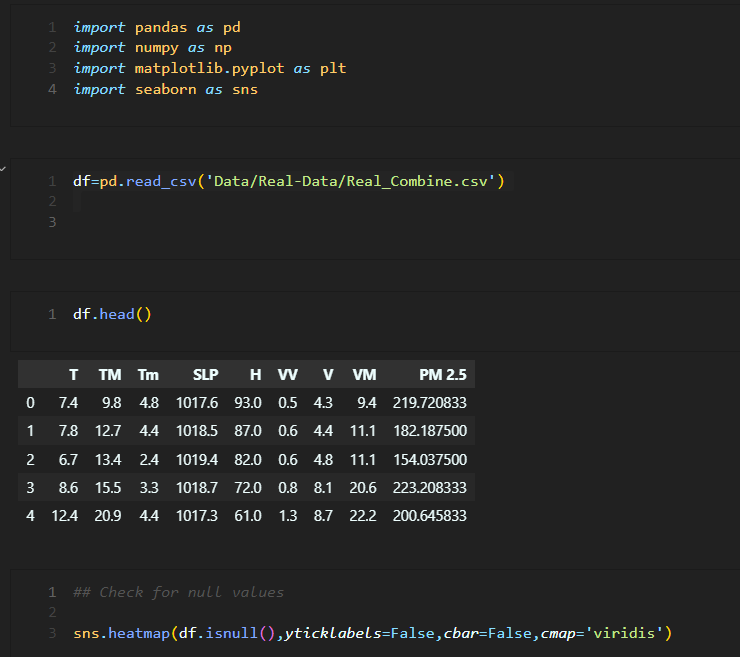


Figure 5 Python code to read the final CSV file and create a heatmap

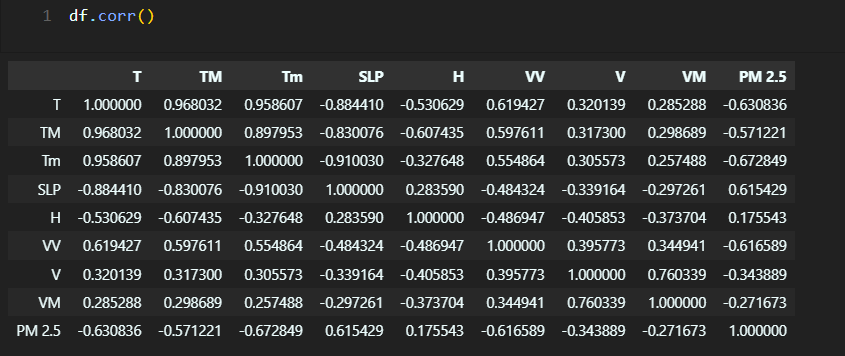


Figure 6 Python code to find the correlation between dependent and independent features

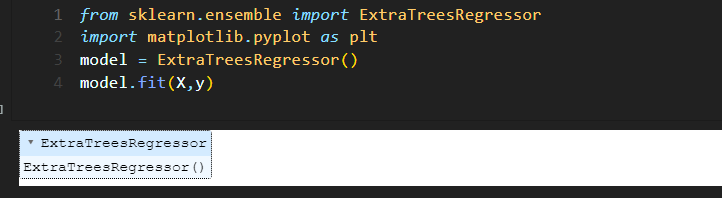


Figure 7 Code to create an object and fit the ExtraTreesRegressor

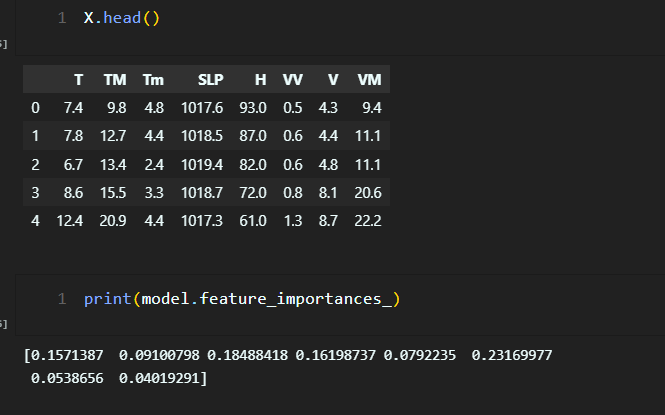


Figure 8 Code to get the feature importances

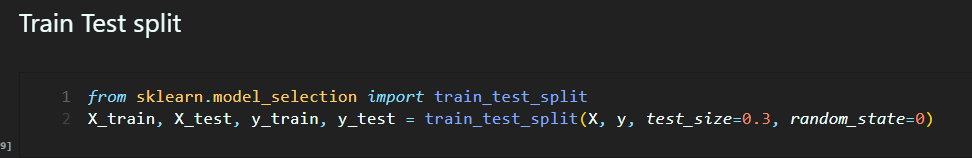


Figure 9 Code to split the dataset to apply the machine learning algorithms

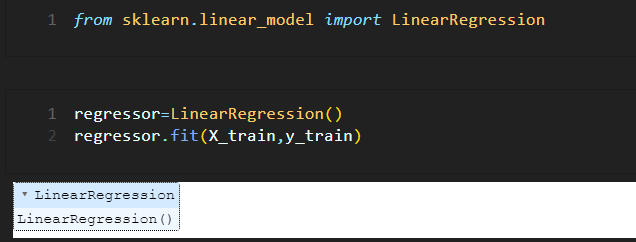


Figure 10 Applying Linear Regression

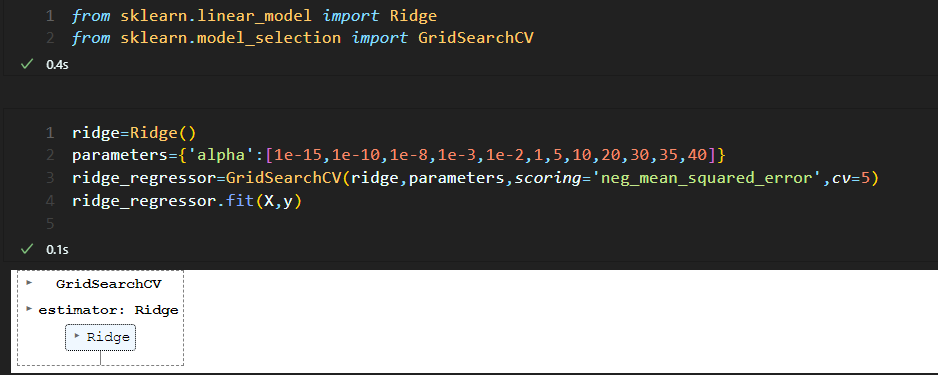


Figure 11 Applying Ridge Regression

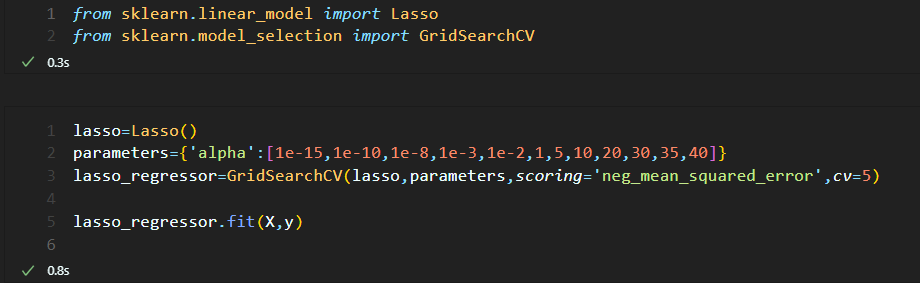


Figure 12 Applying Lasso Regression

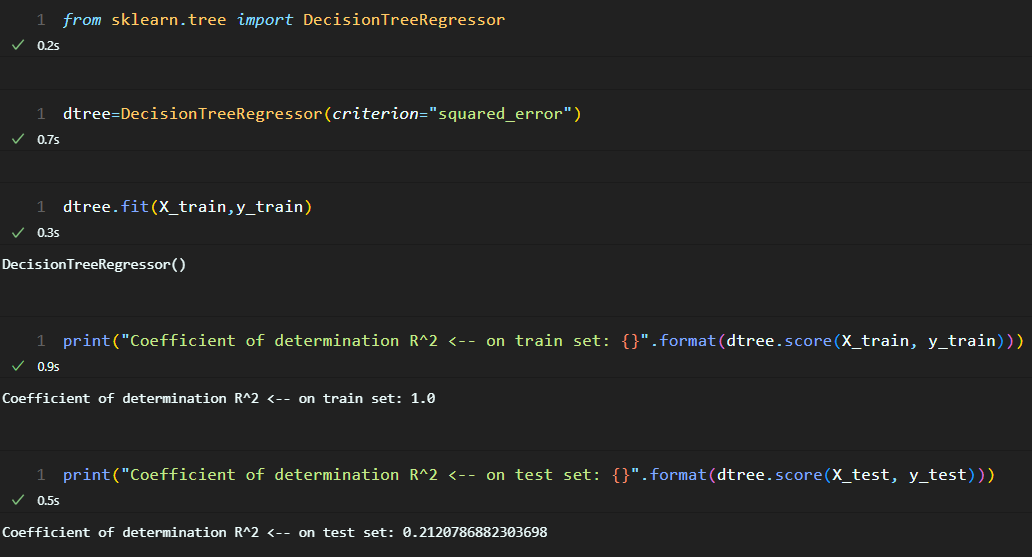


Figure 13 Applying Decision Tree Regression

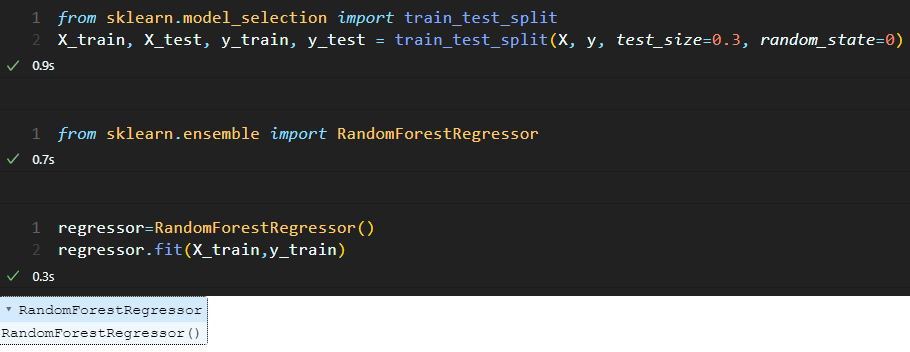


Figure 14 Applying Random Forest Regression