Forecasting Dangerous Levels of Ground Ozone with Machine Learning

New Mexico

Supercomputing Challenge

Final Report

April 11, 2021

Team 1

V. Sue Cleveland High School

Team Members:

- Eliana Juarez
- Graciela Rodríguez
- Sofia Juarez

Teacher(s):

- Ms. Johnston
- Ms. Huey

Project Mentor:

- Mark Petersen

Table of Contents

Executive Summary

Ground-level ozone is a secondary pollutant that is harmful to urban populations by increasing risk of heart and lung disease and harming agricultural crops, and is particularly high in the developing world. To warn populations of hazardous ozone levels, we developed a code to analyze and compare different machine learning algorithms to reliably predict the ozone concentration 24 hours in advance. The final project was about 4,400 lines of code long and was created using Python 3 and Jupyter notebooks on a personal laptop. This project used hourly records of four weather variables and 12 air pollutant variables over the course of one year in Delhi, India to train multiple predictive models. To create the best model, this project tuned, trained, and tested seven machine learning algorithms and compared their predictive abilities using cross-validation. Among the seven models R^2 values varied from 0.39 to 0.61, with XGBoost, Random Forest, and K-Nearest Neighbors Regression ranking highest. When trained by separate seasons across five years, predictive capabilities of all models were significantly higher, with a maximum R^2 of 0.754 during winter. When tested, the three best performing models could reliably predict tropospheric ozone concentrations 24 hours in advance, where 50% of the predictions had a percentage error of less than 10%. This project differed from previous research on the topic of ozone forecasts by comparing the performance of more and newer machine learning algorithms, testing more variables, and using the most recent years of data available. The best model, Winter XGBoost, had a higher R^2 value than the models developed in other studies, a significant achievement of the project. These results show that weather and pollutant data have sufficient predictive power for 24-hour ozone warnings and that machine learning can greatly improve upon simpler forecasting methods. Thus, advanced data monitoring and computing can improve safety for people worldwide.

1. Introduction

Significance

Air pollution is among the leading causes of premature deaths, causing approximately 4.2 million deaths worldwide each year. Among the causes of death in these cases are lung cancer, heart disease, and respiratory diseases [1]. One of these harmful pollutants is ozone (O_3).

Health effects of tropospheric (or "ground") ozone can include coughing, lung irritation, throat irritation, wheezing, and trouble breathing, especially when doing physical activities outside [2]. According to the CDC, people with asthma, bronchitis, or emphysema, elderly people, people who exercise outside, and children are all examples of people who are affected the most by high levels of ground ozone. Ground ozone also damages plant leaves and can harm agricultural crops [3]. One reason why it is important to be able to predict when ozone levels will be highest is so high-risk people can know to stay indoors, and people in general will know to limit their use of ozone-emitting activities.

Background

Ground Ozone

To avoid confusion, it is necessary to specifically define the problem this project is addressing. "Ground ozone", also known as "tropospheric ozone", is ozone that is located in the troposphere, the lowest layer of the Earth's atmosphere that humans live in and that experiences weather. Unlike ozone located in the stratosphere, which is located about 50 km above the Earth and protects the Earth against the ultraviolet radiation from the sun, tropospheric ozone is harmful to both humans and plants.

Of the myriad pollutants that this project could have chosen to focus on, ground ozone was selected for several reasons. One consideration was that ozone is the main component of photochemical smog. Photochemical smog causes a variety of health issues mentioned previously, and most noticeably, reduced visibility [4]. As the main element of photochemical smog, ozone is a pollutant that has conspicuous immediate effects and therefore is worth tracking. Another factor was ozone's status as a secondary pollutant.

Secondary pollutants are not directly emitted. Instead, they are the product of a chemical reaction between pollutants that have already been emitted (called primary pollutants). Ozone is formed as a result of a chemical reaction between pollutants like nitrogen oxides and volatile organic compounds, both emitted by humans through vehicles, power plants, factories, and more [5]. Due to the presence of heat and sunlight, these primary pollutants react together and form tropospheric ozone. The fact that ozone is a secondary pollutant means it cannot be directly tracked (as it is not directly emitted), which means it is useful to have a model to forecast ozone and its correlations with other factors.

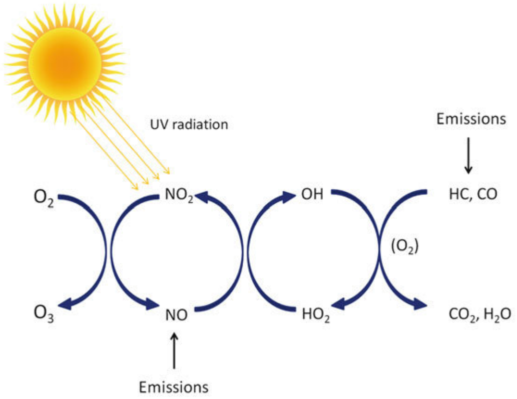


Figure 1 - Diagram of Tropospheric Ozone Formation [6]

Because of this photochemical reaction, we hypothesized that by creating an effective Python code that utilizes machine learning algorithms and conducts proper data clean up, the amount of tropospheric ozone in 24 hours could be reliably predicted based on current meteorological and pollutant data.

Why Delhi?

Delhi, India, is the focus of this project. India was chosen because, out of the 50 most polluted cities in the world (based on the most harmful pollutant, particulate matter 2.5), more than half of the cities were in India [7]. In 2019 alone, about 1.7 million deaths (18% of total deaths in the country) were attributed to air pollution [8]. The visible effects of photochemical smog (ozone) are also prominent in many cities.

© Reuters/A. Fadnavis

Figure 2- Photochemical smog in Delhi, India [9]

India also had freer access to data than other countries with severe air pollution (such as China), so it was chosen as the best country to analyze. The team ended up choosing the city of Delhi due to the relatively abundant data that was available.

Air pollution is a particularly significant problem in Delhi. The average total suspended particulate level (TSP) from 1991-1994 was over the World Health Organization's (WHO) 24-hour standard 97% of the time. The air pollution level was, on average, over five times the WHO annual average standard [10]. This has had severe health effects on the people of Delhi, causing significant increases in respiratory symptoms, including asthma, coughing, breathlessness, and chest discomfort. There was also an increase in the number of all-natural-cause deaths in Delhi, which correlated with the increase in outdoor air pollution [11] as well as an increase in emergency room visits for acute asthma, chronic obstructive airway disease, and acute coronary events [12].

Purpose

The purpose of this project is to forecast ozone levels to avoid these problems, a goal which will be achieved by informing people of the problem in order to give them an opportunity to alter their behaviors in such a way that lessens the severity of the problem and prevents them from putting their health at risk by unknowingly participating in these behaviors at times when the ozone concentration is especially high. This way, people can avoid the adverse effects that come with this problem.

An example of this project's possible application would be a 24 hour advance high-ozone level warning. After receiving this, people could alter their plans so as to avoid things like unnecessary driving or outdoor exercise.

This would be especially useful for vulnerable populations, who would then be able to limit their exposure to these pollutants and avoid contributing to the problem, thus reducing its impact.

The end goal of this project is to create a Python code which will clean up data and use machine learning to analyze relationships between current pollutant and meteorological variables and future ground-ozone levels, using them to train a machine learning model that will be able to produce statistically significant predictions of hourly ozone levels in advance. This way, we could help improve on current methods of pollutant forecasting in the city by using methods that could eventually be applied to other cities worldwide.

2. Description

Scope

Air Pollution in Delhi, India

The air pollution in Delhi, India is primarily due to factories, vehicles, lit garbage, power plants, and stubble burning. Stubble burning, a cheap method for preparing plots of land for the next crop, creates a great amount of pollution due to the smog and smoke. This method of burning crops produces emissions of harmful gases including carbon monoxide and nitrogen dioxide, both precursor chemicals of ozone formation, as well as hydrocarbons, particulate matter, and sulfur dioxide. These levels of pollutants can be responsible for up to 42% of Delhi's air pollution, depending on the season and weather [13].

This project utilized data from Delhi, India, in the year 2015. The reason that this city was chosen is that, like many other cities in developing countries, it faces high air pollution which is a threat to its large urban population, and has properties similar to other highly-polluted cities. Since sunlight and these other primary pollutants that cause ozone formation are commonplace in cities worldwide in both developing and more developed countries, this code can be applied and trained in other cities with high ozone concentrations. Some examples are Los Angeles, Mexico City, Mumbai, and Beijing.

Methods, details

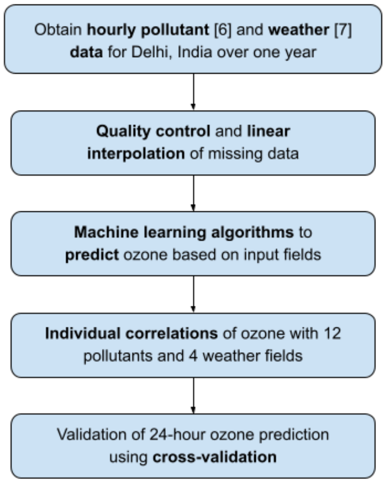
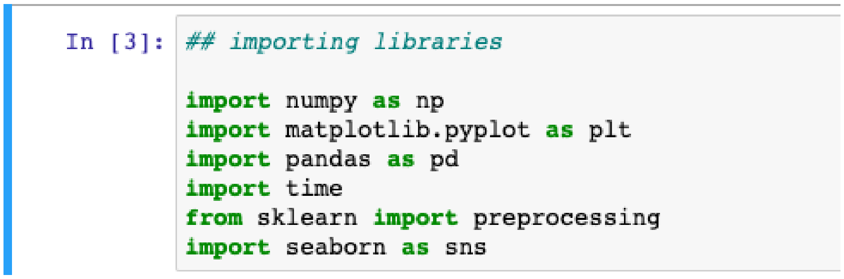

Obtain **hourly pollutant** [6] and **weather** [7] **data** for Delhi, India over one year
Quality control and **linear interpolation** of missing data
Machine learning algorithms to **predict** ozone based on input fields
Individual correlations of ozone with 12 pollutants and 4 weather fields
Validation of 24-hour ozone prediction using **cross-validation**

Figure 3 - Simplified Process Flowchart

Materials

All of the code was done on Jupyter notebooks with a personal laptop computer. The language used was Python 3, and we used a variety of libraries, which are sets of code that can include useful functions and tools. Some of the main libraries that we used were Numpy, Pandas, Matplotlib, and Sklearn.



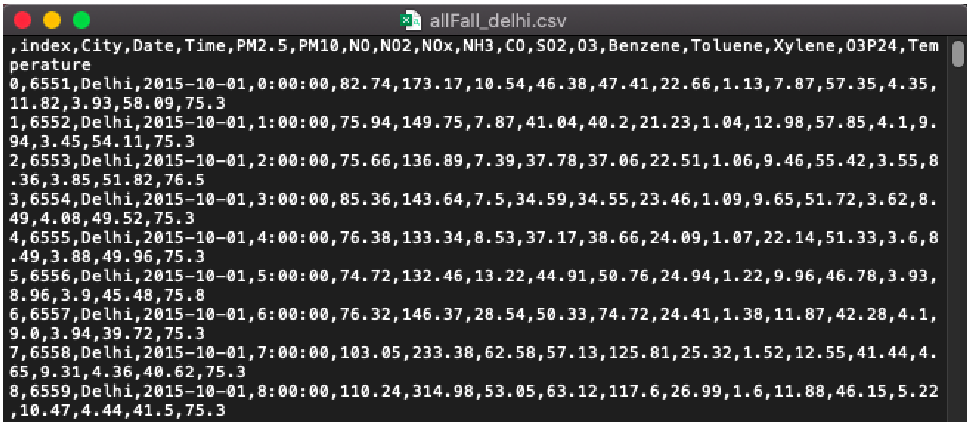
```
In [3]: ## importing libraries

        import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
        import time
        from sklearn import preprocessing
        import seaborn as sns
```

Figure 4 - Screenshot of importing of libraries into Notebook

The final code included two data cleanup programs, one hyper-tuning and testing program, a year-long model training and testing program, four seasonal models that retrained each of the models, and a tropospheric ozone calculator- about 4,400 lines of code in total. There were also several miscellaneous Jupyter notebooks used to test and experiment with different methods as well as generate the visuals throughout this report.

In order to manipulate the hundreds of thousands of rows of data we were using, the Python library called Pandas was used along with comma-separated-values (CSV) files.

- A copy of the code can be found at https://github.com/elianajuarez/ozone-forecast.



```
,index,City,Date,Time,PM2.5,PM10,NO,NO2,NOx,NH3,CO,SO2,O3,Benzene,Toluene,Xylene,O3P24,Tem
perature
0,6551,Delhi,2015-10-01,0:00:00,82.74,173.17,10.54,46.38,47.41,22.66,1.13,7.87,57.35,4.35,
11.82,3.93,58.09,75.3
1,6552,Delhi,2015-10-01,1:00:00,75.94,149.75,7.87,41.04,40.2,21.23,1.04,12.98,57.85,4.1,9.
94,3.45,54.11,75.3
2,6553,Delhi,2015-10-01,2:00:00,75.66,136.89,7.39,37.78,37.06,22.51,1.06,9.46,55.42,3.55,8
.36,3.85,51.82,76.5
3,6554,Delhi,2015-10-01,3:00:00,85.36,143.64,7.5,34.59,34.55,23.46,1.09,9.65,51.72,3.62,8.
49,4.08,49.52,75.3
4,6555,Delhi,2015-10-01,4:00:00,76.38,133.34,8.53,37.17,38.66,24.09,1.07,22.14,51.33,3.6,8
.49,3.88,49.96,75.3
5,6556,Delhi,2015-10-01,5:00:00,74.72,132.46,13.22,44.91,50.76,24.94,1.22,9.96,46.78,3.93,
8.96,3.9,45.48,75.8
6,6557,Delhi,2015-10-01,6:00:00,76.32,146.37,28.54,50.33,74.72,24.41,1.38,11.87,42.28,4.1,
9.0,3.94,39.72,75.3
7,6558,Delhi,2015-10-01,7:00:00,103.05,233.38,62.58,57.13,125.81,25.32,1.52,12.55,41.44,4.
65,9.31,4.36,40.62,75.3
8,6559,Delhi,2015-10-01,8:00:00,110.24,314.98,53.05,63.12,117.6,26.99,1.6,11.88,46.15,5.22
,10.47,4.44,41.5,75.3
```

Figure 5 - Screenshot of CSV "Fall" file

Choosing a Location

The first step was deciding on a city in India on which to conduct the research. In order to analyze the different levels of pollutants in many cities across India, we downloaded a data set containing daily pollutant measurements for over 25 different cities in India.

Next, we made a Python code that created histograms of the daily pollutant levels across every city for each pollutant and added markers on the points at the acceptable level of pollutant (using the Environmental Protection Agency's standards). This gave us an idea of how many days each city experienced over the limits, putting millions of people in potential harm.

We found the cities with the most days over the acceptable limits were Amaravati, Mumbai, Patna, and Delhi. We decided on Delhi due to its consistently high daily pollutant levels and the data's relative lack of nan (not a number or null) values.

Cleaning the Data

After narrowing down our data, we downloaded a dataset of hourly pollutant levels in Delhi from January 2015 to June 2020 from the Central Pollution Control Board of India [14]. Although it was mostly complete, we still needed to clean the data in order to create proper regression models, which could not simply pass over the nan values (like the histograms did). To do that, we created a code that checks each column for nan values. If found, a nan value would be replaced by the average between the previous and the next non-nan value. During the entire process (called linear interpolation), the percentage of nan values before and after being cleaned was printed. To ensure the "trueness" of data, we did not fill in more than 3% of the original values. This was done to satisfy the strict zero-nan requirements of different machine learning algorithms later instead of deleting entire rows of data because of missing one value.

For meteorological data, we used the Visual Crossing Weather application program interface (API) [15] to compile a comma-separated values (CSV) file of hourly weather data in Delhi for the years 2015-2020. The cleaning process was repeated for this data. We also checked that the station where this weather data was recorded was within a 5-kilometer distance of the station recording pollutant levels in the area.

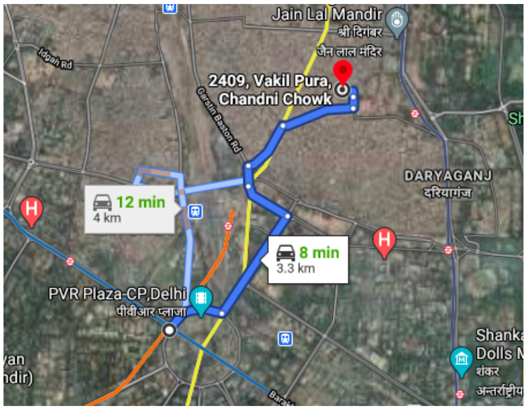
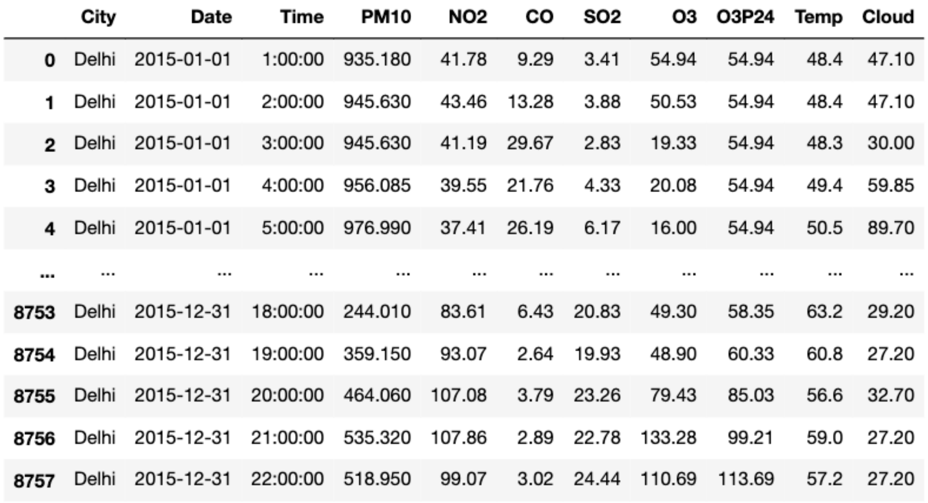


Figure 6 - Distance between the two stations

Once cleaned, the pollutant and weather data was compiled in one CSV file, and was manipulated using the Python library Pandas. The primary data structure of this library is called a data frame, which consists of number-indexed rows and named columns (the variable names).



	City	Date	Time	PM10	NO2	CO	SO2	O3	O3P24	Temp	Cloud
0	Delhi	2015-01-01	1:00:00	935.180	41.78	9.29	3.41	54.94	54.94	48.4	47.10
1	Delhi	2015-01-01	2:00:00	945.630	43.46	13.28	3.88	50.53	54.94	48.4	47.10
2	Delhi	2015-01-01	3:00:00	945.630	41.19	29.67	2.83	19.33	54.94	48.3	30.00
3	Delhi	2015-01-01	4:00:00	956.085	39.55	21.76	4.33	20.08	54.94	49.4	59.85
4	Delhi	2015-01-01	5:00:00	976.990	37.41	26.19	6.17	16.00	54.94	50.5	89.70
...	...	...	...	...	...	...	...	...	...	...	...
8753	Delhi	2015-12-31	18:00:00	244.010	83.61	6.43	20.83	49.30	58.35	63.2	29.20
8754	Delhi	2015-12-31	19:00:00	359.150	93.07	2.64	19.93	48.90	60.33	60.8	27.20
8755	Delhi	2015-12-31	20:00:00	464.060	107.08	3.79	23.26	79.43	85.03	56.6	32.70
8756	Delhi	2015-12-31	21:00:00	535.320	107.86	2.89	22.78	133.28	99.21	59.0	27.20
8757	Delhi	2015-12-31	22:00:00	518.950	99.07	3.02	24.44	110.69	113.69	57.2	27.20

Figure 7 - Sample of Cleaned Pandas Data frame

Individual Correlations and Analysis

The first step in creating a predictive model is exploring the data and preprocessing it. Since we are analyzing the relationships between a variety of meteorological and pollutant variables with ozone, we created scatter plots of these correlations. As pictured in Figure 8, some variables correlated stronger than others.

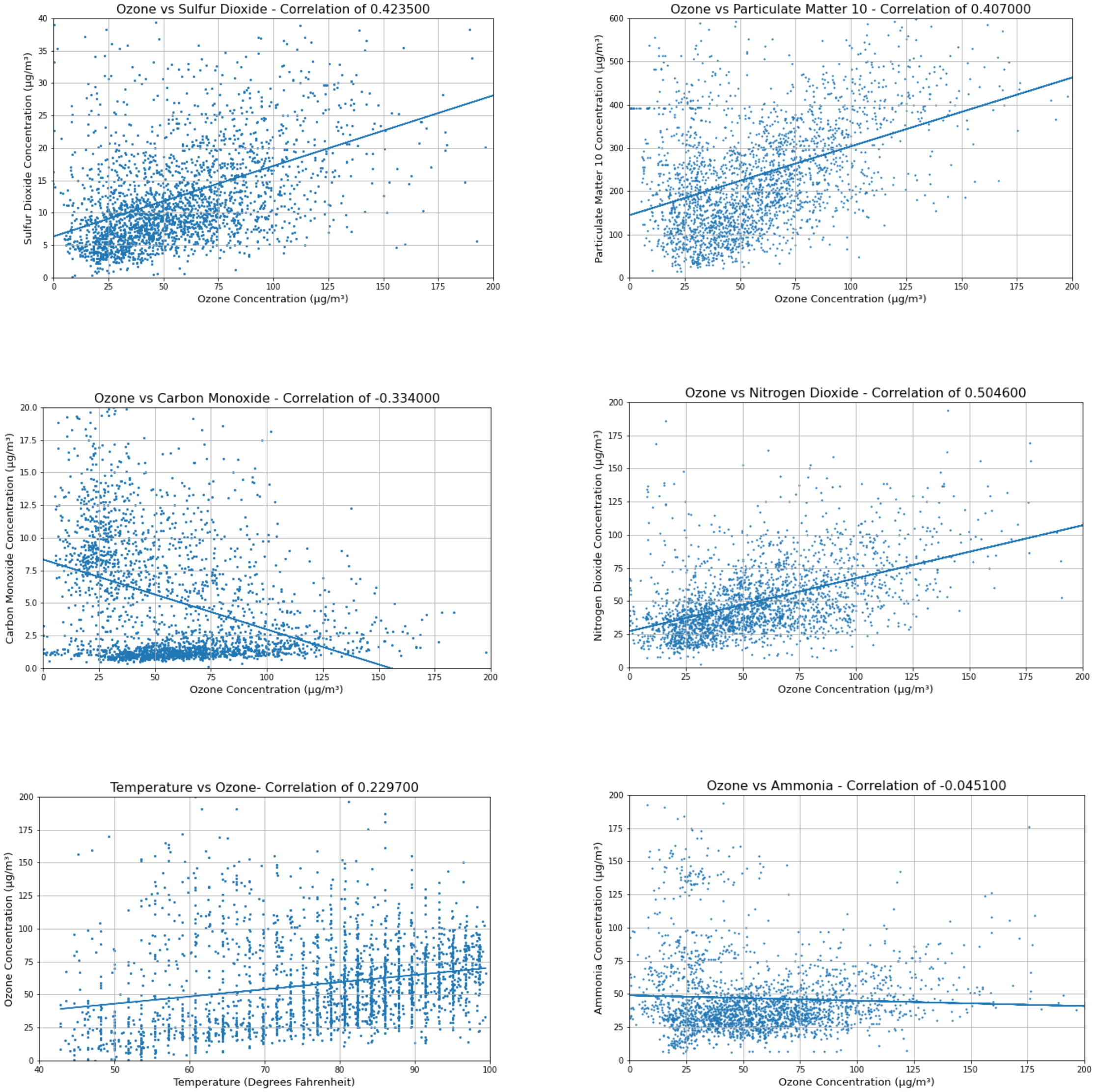
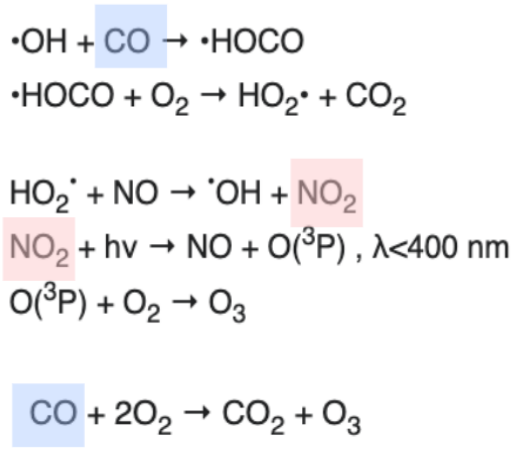


Figure 8 - Individual correlations between variables and ozone

The pollutants involved in the chemical equation of ozone, carbon monoxide and nitrogen dioxide, both showed significant correlations (with the correlation coefficients of -.33 and 0.5, respectively). Carbon monoxide's negative correlation with ozone indicates that CO is a limiting factor and that the formation of ozone causes a decrease in carbon monoxide. On the other hand, nitrogen dioxide is positively correlated with ozone, indicating that there is an abundance of NO_2 and that the more NO_2 available, the more ozone can form.



$$\cdot OH + CO \rightarrow \cdot HOCO$$
$$\cdot HOCO + O_2 \rightarrow HO_2\cdot + CO_2$$

$$HO_2^{\cdot} + NO \rightarrow {}^{\cdot}OH + NO_2$$
$$NO_2 + h\nu \rightarrow NO + O(^3P)\text{ , }\lambda<400\text{ nm}$$
$$O(^3P) + O_2 \rightarrow O_3$$

$$CO + 2O_2 \rightarrow CO_2 + O_3$$

Figure 9 - Ground ozone chemical equation [16], highlighted chemicals involved in this study

Some other noteworthy correlations were with particulate matter 10 (R = 0.4), sulfur dioxide (R = 0.42), and temperature (R = 0.23). These correlations indicate that these factors tend to rise and fall together. Although individually they correlate rather weakly with ozone, together they can be used to train predictive models to get the best possible results. Unrelated variables were also discarded in this step.

In order to further explore the relationship between ozone and the other pollutants involved in its formation, we created plots of the hourly concentrations against one another. The strongest correlating variable, nitrogen dioxide, showed a time lag of about 5 hours between the primary and secondary pollutants (fig. 10).

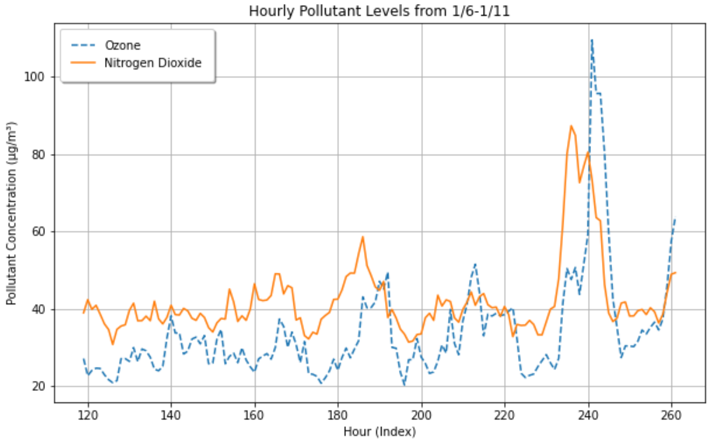


Figure 10 - Plot of hourly pollutant levels of ozone (blue) and nitrogen dioxide (orange)

Regression Analysis

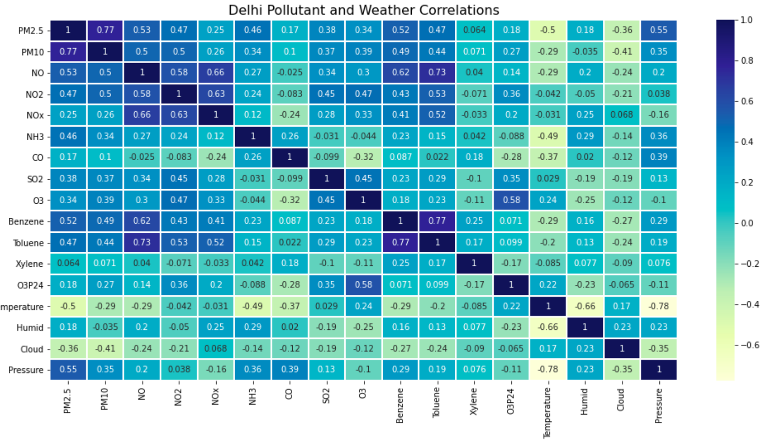
In order to predict future ozone levels, this project used regression analysis. Regression analysis is a type of predictive modeling in which a relationship is determined between one dependent variable (ozone) and one or more independent variables (weather and pollutant data). There are different algorithms by which this process can be done, and the simplest one is multiple linear regression (fig. 11).

$$\boxed{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ ... + \beta_i X_i}$$

Y : Dependent variable
β_0 : Intercept
β_i : Slope for X_i
X = Independent variable

Figure 11 - Multiple Linear Regression Equation [17]

In linear regression as well as some other algorithms, multicollinearity can be a problem. Issues arise when there are strong correlations between the independent variables, and the statistical significance of each variable is diminished. This reduction is because linear regression analyzes the relationship between each independent variable and the dependent variable, and multicollinearity blurs which variable a change in the dependent variable can be attributed to. In order to avoid this, it is important to analyze the relationships between each independent variable before even attempting linear regression. To do so, we used the Python library, Seaborn, to produce a heatmap, which is a color-coded chart that generates a visual representation of the correlation between each variable in a data frame. This chart was used to see which variables could be potential issues later on.



	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	O3P24	Temperature	Humid	Cloud	Pressure
PM2.5	1	0.77	0.53	0.47	0.25	0.46	0.17	0.38	0.34	0.52	0.47	0.064	0.18	-0.5	0.18	-0.36	0.55
PM10	0.77	1	0.5	0.5	0.26	0.34	0.1	0.37	0.39	0.49	0.44	0.071	0.27	-0.29	-0.035	-0.41	0.35
NO	0.53	0.5	1	0.58	0.66	0.27	-0.025	0.34	0.3	0.62	0.73	0.04	0.14	-0.29	0.2	-0.24	0.2
NO2	0.47	0.5	0.58	1	0.63	0.24	-0.083	0.45	0.47	0.43	0.53	-0.071	0.36	-0.042	-0.05	-0.21	0.038
NOx	0.25	0.26	0.66	0.63	1	0.12	-0.24	0.28	0.33	0.41	0.52	-0.033	0.2	-0.031	0.25	0.068	-0.16
NH3	0.46	0.34	0.27	0.24	0.12	1	0.26	-0.031	-0.044	0.23	0.15	0.042	-0.088	-0.49	0.29	-0.14	0.36
CO	0.17	0.1	-0.025	-0.083	-0.24	0.26	1	-0.099	-0.32	0.087	0.022	0.18	-0.28	-0.37	0.02	-0.12	0.39
SO2	0.38	0.37	0.34	0.45	0.28	-0.031	-0.099	1	0.45	0.23	0.29	-0.1	0.35	0.029	-0.19	-0.19	0.13
O3	0.34	0.39	0.3	0.47	0.33	-0.044	-0.32	0.45	1	0.18	0.23	-0.11	0.58	0.24	-0.25	-0.12	-0.1
Benzene	0.52	0.49	0.62	0.43	0.41	0.23	0.087	0.23	0.18	1	0.77	0.25	0.071	-0.29	0.16	-0.27	0.29
Toluene	0.47	0.44	0.73	0.53	0.52	0.15	0.022	0.29	0.23	0.77	1	0.17	0.099	-0.2	0.13	-0.24	0.19
Xylene	0.064	0.071	0.04	-0.071	-0.033	0.042	0.18	-0.1	-0.11	0.25	0.17	1	-0.17	-0.085	0.077	-0.09	0.076
O3P24	0.18	0.27	0.14	0.36	0.2	-0.088	-0.28	0.35	0.58	0.071	0.099	-0.17	1	0.22	-0.23	-0.065	-0.11
Temperature	-0.5	-0.29	-0.29	-0.042	-0.031	-0.49	-0.37	0.029	0.24	-0.29	-0.2	-0.085	0.22	1	-0.66	0.17	-0.78
Humid	0.18	-0.035	0.2	-0.05	0.25	0.29	0.02	-0.19	-0.25	0.16	0.13	0.077	-0.23	-0.66	1	0.23	0.23
Cloud	-0.36	-0.41	-0.24	-0.21	0.068	-0.14	-0.12	-0.19	-0.12	-0.27	-0.24	-0.09	-0.065	0.17	0.23	1	-0.35
Pressure	0.55	0.35	0.2	0.038	-0.16	0.36	0.39	0.13	-0.1	0.29	0.19	0.076	-0.11	-0.78	0.23	-0.35	1

Figure 12 - Heatmap of Dataframe

Further Data Exploration

Another step of the initial data exploration was the creation of plots of the average ozone concentration at each hour during the seasons to get an idea of the possible patterns and cycles that ozone goes through over a day. As seen in Figure 13, fall generally had the highest ozone concentrations throughout the day and winter had the lowest. During all seasons, there is also a notable increase during sunlight hours followed by a decrease in the night, most likely due to the amount of solar radiation available to cause the formation reaction to take place.

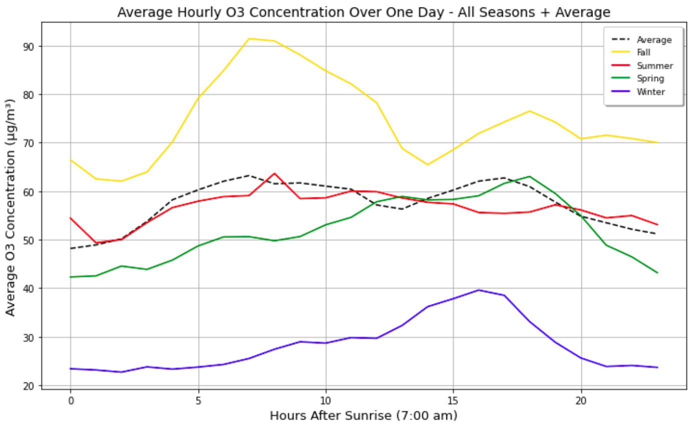


Figure 13 - Average Hourly O_3 Concentration Over One Day for All Seasons

Forecast Time

The next step was determining which hour to actually predict in advance, since the purpose of this project was to forecast future ozone levels. Generally, the further ahead in time the more useful, but the models still had to be significant enough to be helpful. In order to weigh this balance, a code was created that saved the R^2 value of a linear regression model for each season to a list, which was then plotted over time. This graph displayed the significance of each linear regression model over each "future" hour, pictured below (fig. 14).

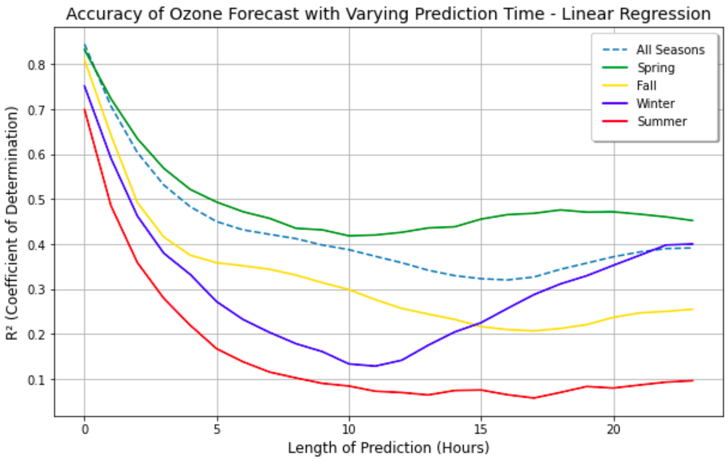
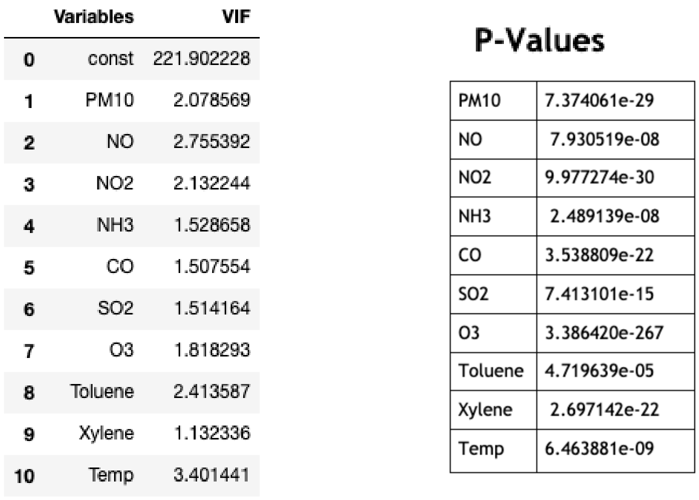


Figure 14 - Accuracy of Ozone Forecast With Varying Prediction Time

As expected, there is a decrease in predictive ability the more hours in advance that the forecast predicts, leveling off a bit at around the ten-hour mark. However, it is notable that although the line dips a bit, the significance of the year-long model increases by the 24-hour mark to be almost equal to the 10-hour mark. Taking this into consideration, it was decided that the models will aim to predict the tropospheric ozone concentration in 24 hours, since its predictive ability is similar to in 10 hours yet is in more than twice the amount of time ahead.

Feature Selection and Scaling

Next came feature selection of the actual variables to include. The original data frame included 13 pollutant and 14 weather variables, which had been reduced to 12 pollutant and 4 weather variables after removing uncleanable or unrelated variables. The variables were then evaluated to actually be useful or not in explaining ozone concentrations using the significance level of 0.05. The next part of feature selection was only necessary for models impacted by multicollinearity. This was done by using a code to determine the variance inflation factor (or VIF) of each variable. If the VIF was equal to or greater than 10, the variable was either removed or other less relevant variables that the original variable correlated with were removed. The p-values and VIFs of each of these variables is pictured in Figure 15.



	Variables	VIF
0	const	221.902228
1	PM10	2.078569
2	NO	2.755392
3	NO2	2.132244
4	NH3	1.528658
5	CO	1.507554
6	SO2	1.514164
7	O3	1.818293
8	Toluene	2.413587
9	Xylene	1.132336
10	Temp	3.401441

P-Values

PM10	7.374061e-29
NO	7.930519e-08
NO2	9.977274e-30
NH3	2.489139e-08
CO	3.538809e-22
SO2	7.413101e-15
O3	3.386420e-267
Toluene	4.719639e-05
Xylene	2.697142e-22
Temp	6.463881e-09

Figure 15 - P-values and VIF of variables used in models

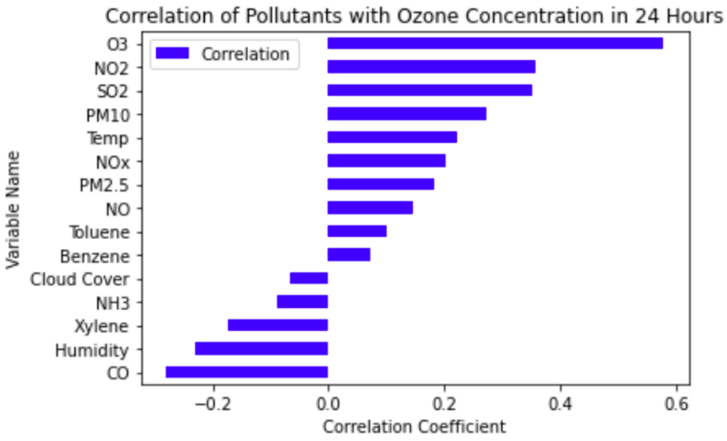


Figure 16 - Histogram of Correlations Between Variables and Future Ozone Levels

Figure 16 displays the strength of the linear correlations between the input variables being used in the models with the ozone concentration in 24 hours. The strongest relationships with future ozone concentration were with current ozone, nitrogen dioxide, sulfur dioxide, particulate matter 10, and carbon monoxide pollutant concentrations. Some of these correlations may be due to the fact that different pollutants tend to rise and fall together, or they could indicate some causal relationship (such as an increase in precursor chemicals driving an increase of ozone concentration).

Another important part of some machine learning models involved scaling the data. This is done to place all independent variables into a fixed range so that different variables with varying units can be handled appropriately. There are different methods of scaling, and the Robust Scaler was found to have the best results by using statistics that are robust to outliers when scaling the data.

Tuning, Training, and Testing the Models

Finally, the time came to tune, train, and test the seven machine learning models. The algorithms that we would be testing included Linear Regression, K-Nearest Neighbors, Support-Vector Machine, Random Forests, Decision Tree, Adaboost, and XGBoost. Excluding linear regression, these models each have a variety of hyperparameters that can be adjusted to improve model performance. A code was created that tries different combinations of parameters and returns the best ones (using the R^2 to evaluate performance), and they also were adjusted and tested manually. This ensured that the models would return the closest ozone predictions possible.

Regression models can be evaluated in several ways, and after consideration, we chose the metrics of R-squared (R^2), Adjusted R-squared, Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). The R-squared, also known as the coefficient of determination, is a measure of how closely data fit the fitted regression line- it is the proportion of variance in a dependent variable explained by the independent variables in a regression model. The Adjusted R-squared is similar, but it takes into account the variables that do not help the model by lowering the score. If the Adj. R-squared value is similar to the R-squared value, it means that the input variables are useful and contribute to the model. The other two are statistics related to the error of the model, with the RMSE being more influenced by outliers than the MAE.

Regression analysis with machine learning is usually done by randomly splitting the dataset into 90% for training and 10% for testing. The 10% portion is data that the model has not yet seen, and it attempts to predict the dependent variable for that portion. The predicted and actual results were compared, and the closer they are to one another, the better and more accurate the prediction is.

Once tuned, the models had to be trained and tested. In order to evaluate the accuracy of each model well, we trained and tested each model 10 times, repeating the 90-10 split 10 times to test all of the data at a certain point. This is done in order to eliminate the possibility that the random 10% of data simply happens to match the predicted values more than in another round, giving that model an arbitrarily higher accuracy score. This process of repeating the training and testing is called cross-validation and was done to fairly evaluate the performance of each model.

Results

In the end, the average performance of each model is listed in Table 1, from highest to lowest R-squared values. Table 1 also provides the rest of the statistics from cross-evaluation, and the correlation coefficient was calculated from the linear relationship between actual vs. predicted ozone values.

Table 1. Cross-Validation Results

Model Name	**Correlation Coefficient**	**R^2**	**R^2 Adjusted**	**RMSE**	**MAE**	**Time Taken (Seconds)**
XGBoost	0.784	0.6142	0.6136	21	14	315.64
Random Forests	0.782	0.6122	0.6116	21	14	740.16
KNN	0.739	0.546	0.5453	22	15	1.92
SVM	0.695	0.4834	0.4827	34	26	172.76
Decision Tree	0.656	0.4307	0.4299	25	17	1.80
Linear Regression	0.626	0.3918	0.3909	26	18	0.33
Adaboost	0.623	0.3881	0.3872	26	20	91.47

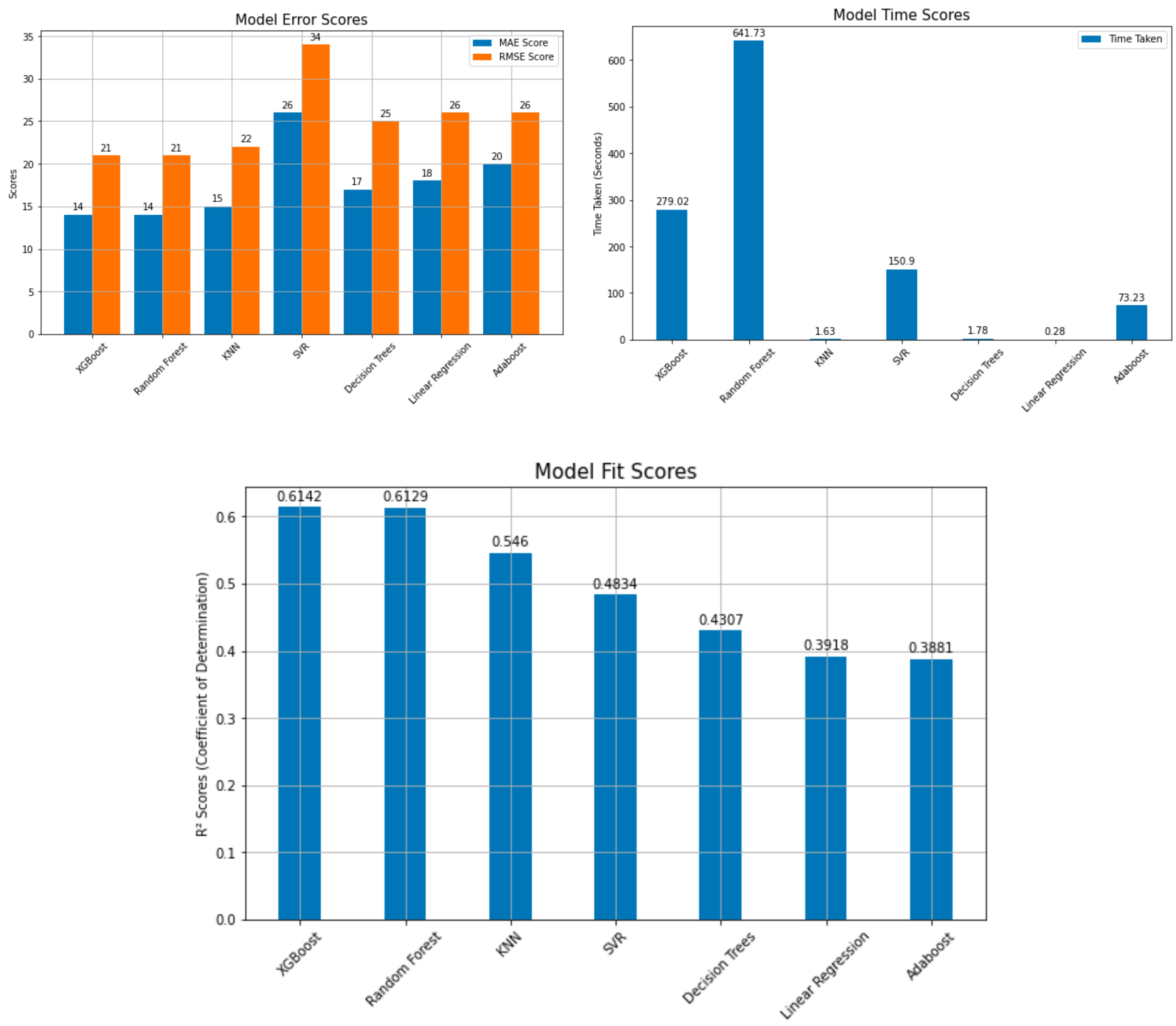


Figure 17- Annual Model Error, Training Time, and Fit Scores

Finally, in order to display the performance of each model on an actual prediction sample, the models were run on a 90%-10% split once, and the results were plotted and explored more deeply (fig. 18).

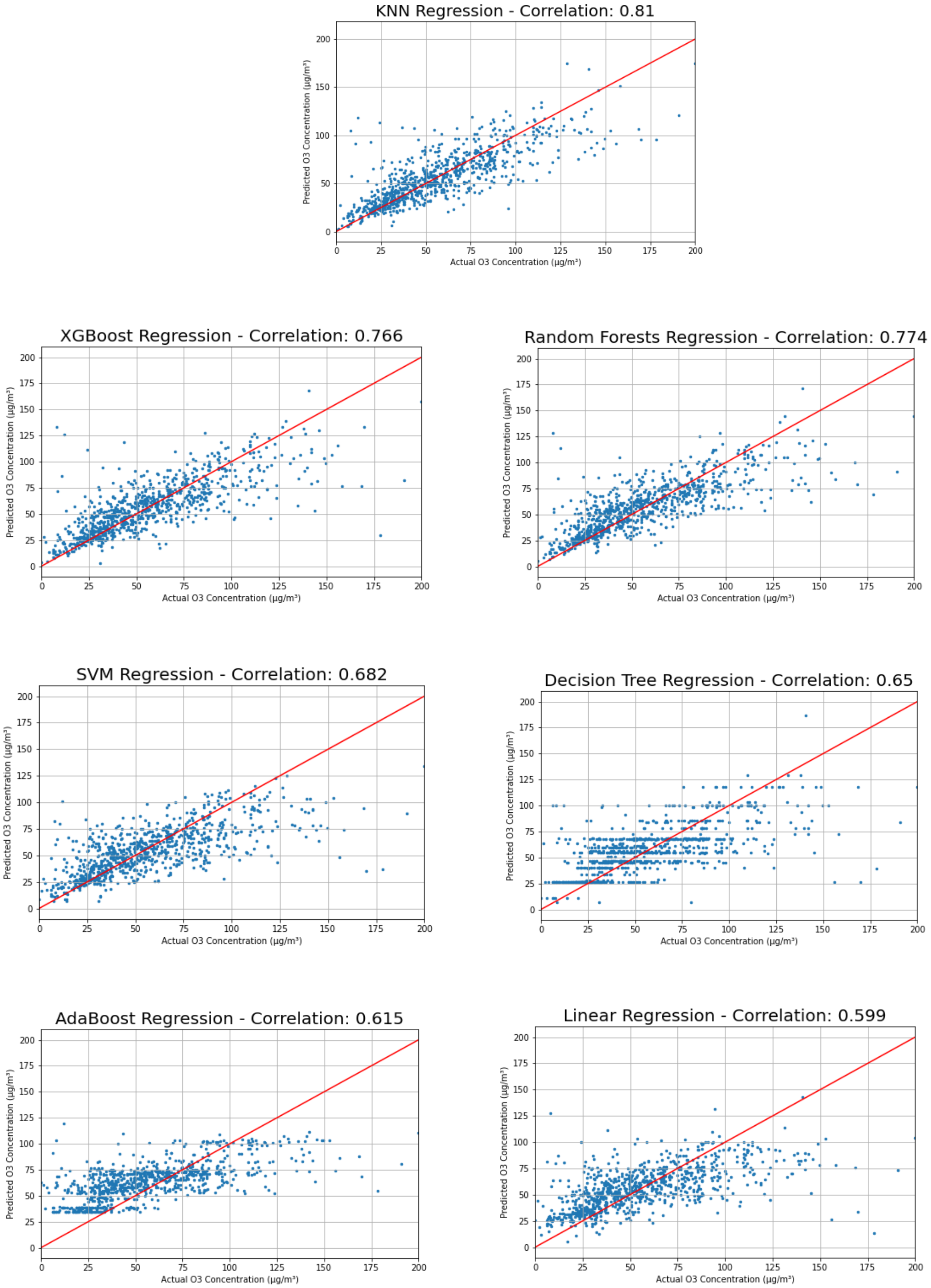


Figure 18 - Actual vs. Predicted results of each predictive model

Each plot displays the relationship between actual vs predicted values, the closer to the line of the perfect fit, the better. As mentioned earlier, some models simply happen to perform better on different sections, and in this case, KNN gave the best results. When tested, the KNN model could predict O_3 concentrations 24 hours in advance, where 68% of the predictions had a percentage error of less than 25%. The model was also able to predict the exact air quality index 92% of the time and was able to predict within one index 98% of the time.

Seasonal Model Splits

All of the results so far were obtained by training the models with data over one year (2015) in Delhi, India. However, as seen earlier in Figure 14, the predictive ability of the linear regression model across seasons differed considerably, performing the lowest during the summer. After further research, it was found that India had unique meteorological seasons, including a four-month long monsoon season [18]. Because of this, we decided to re-train and test the models again, but with each of the four seasons separately across 5 years of data. The new data was split as follows:

- Summer: March, April and May
- Monsoon: June, July, August, September
- Post-Monsoon (Fall): October, November, December
- Winter: January, February

With this new data split, all of the models were retrained for each season. The same methods as before were used to evaluate each model.

Table 2. Summer Cross-Validation Results

Model Name	**Correlation Coefficient (R)**	**R^2**	**R^2 Adjusted**	**RMSE**	**MAE**	**Time Taken (Seconds)**
XGBoost	0.7814	0.6106	0.6102	21	14	259.6255
Random Forest	0.7803	0.6088	0.6084	21	14	795.1088
KNN	0.7469	0.5579	0.5575	23	15	2.8072
SVM	0.7186	0.5164	0.5159	35	27	262.7979

Decision Tree	0.6923	0.4793	0.4787	25	16	2.03220
Linear Regression	0.6856	0.47	0.4694	25	17	0.4092
Adaboost	0.6626	0.439	0.4384	26	19	87.2208

Table 3. Monsoon Cross-Validation Results

Model Name	**Correlation Coefficient (R)**	**R^2**	**R^2 Adjusted**	**RMSE**	**MAE**	**Time Taken (Seconds)**
XGBoost	0.7967	0.6347	0.6344	15	9	306.1112
Random Forest	0.7917	0.6268	0.6265	16	9	1113.1562
KNN	0.7469	0.5722	0.5719	17	10	5.1845
SVM	0.7369	0.543	0.5426	25	18	775.5946
Decision Tree	0.7121	0.5071	0.5067	18	18	2.5892
Linear Regression	0.6733	0.4534	0.453	19	12	0.3607
Adaboost	0.6602	0.4359	0.4354	19	19	116.03747

Table 4. Fall Cross-Validation Results

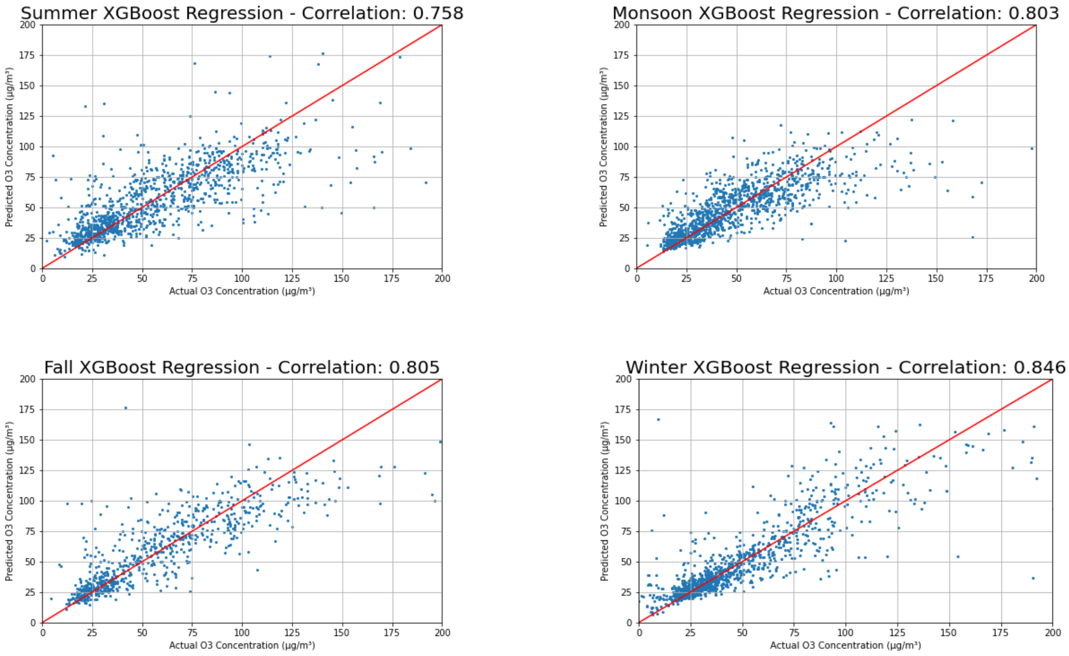
Model Name	**Correlation Coefficient (R)**	**R^2**	**R^2 Adjusted**	**RMSE**	**MAE**	**Time Taken (Seconds)**
XGBoost	0.798	0.6374	0.6368	25	15	200.8
Random Forest	0.797	0.635	0.6344	25	15	604.3107
KNN	0.761	0.5783	0.5777	27	16	1.9284
SVM	0.677	0.4583	0.4575	43	31	158.271
Decision Tree	0.681	0.4642	0.4633	30	18	1.9214

Linear Regression	0.626	0.3925	0.3916	32	21	0.639
Adaboost	0.704	0.4951	0.4942	29	20	57.9426

Table 5. Winter Cross-Validation Results

Model Name	**Correlation Coefficient (R)**	**R^2**	**R^2 Adjusted**	**RMSE**	**MAE**	**Time Taken (Seconds)**
XGBoost	0.8686	0.7545	0.7542	18	10	288.6828
Random Forest	0.8645	0.7474	0.7471	19	11	852.7333
KNN	0.8389	0.7038	0.7035	20	12	2.1494
SVM	0.7980	0.6368	0.6364	39	25	247.7381
Decision Tree	0.7892	0.6229	0.6224	23	13	1.9715
Linear Regression	0.7622	0.5809	0.5804	24	14	0.4576
Adaboost	0.7407	0.5487	0.5482	25	17	288.9010

Figure 19 - Actual vs. Predicted results of XGBoost for each season



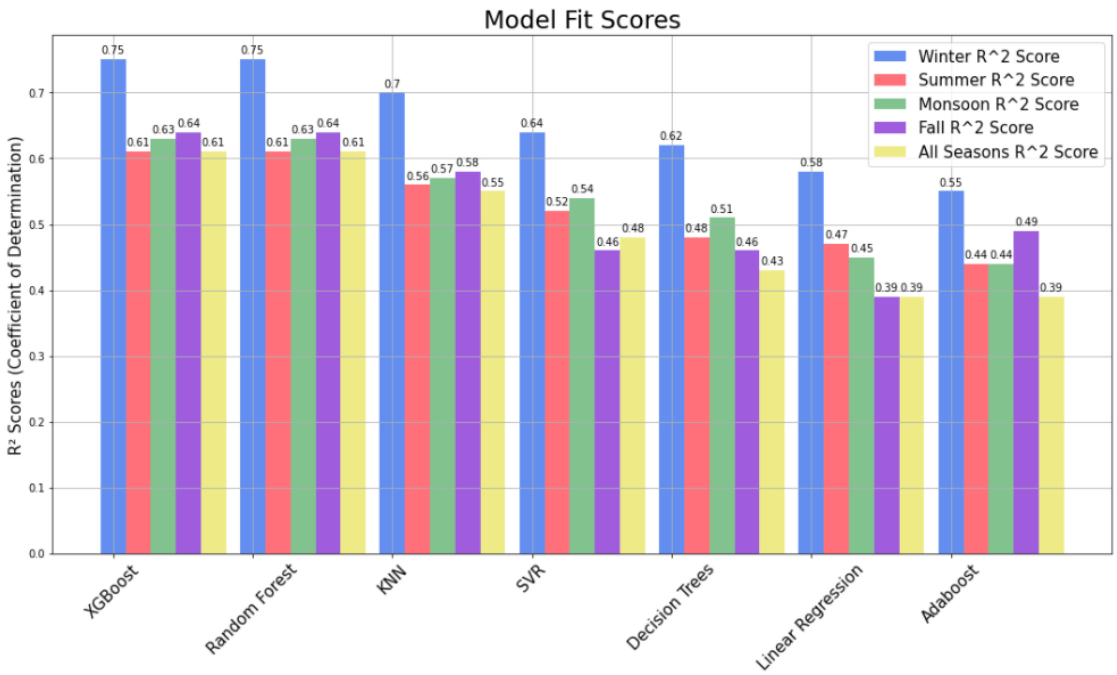


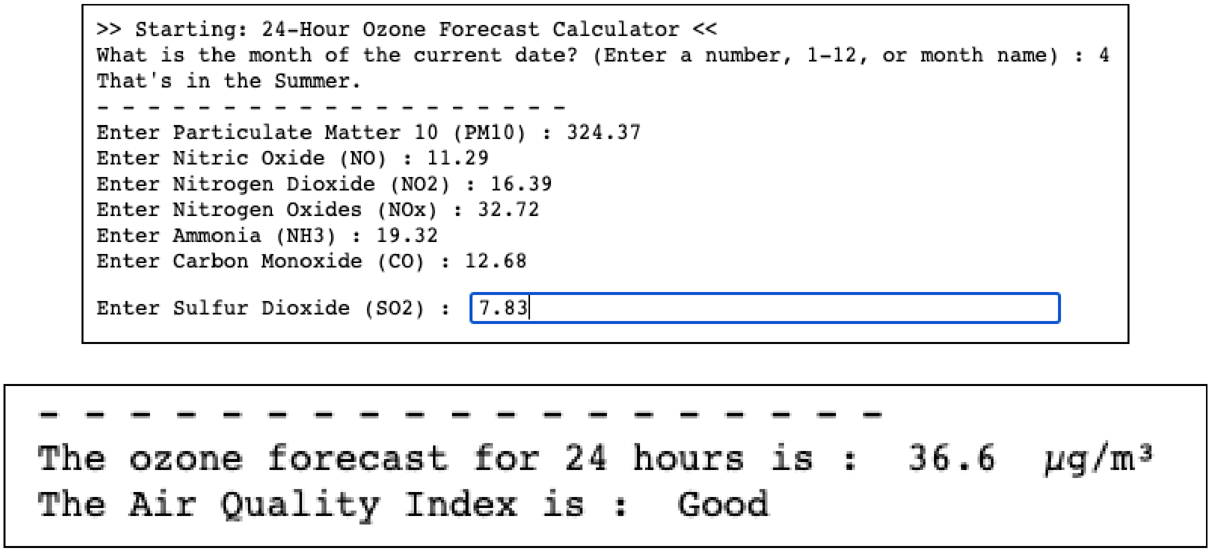
Figure 20- Model Fit Scores for Each Season and Model

Overall, the seasonal models had a higher average accuracy, with the three best performing models (XGBoost, Random Forest, and KNN) remaining the same for all four seasons. The R^2 scores of the highest performing model (XGBoost) for each season were Winter ($R^2 = 0.756$), Fall ($R^2 = 0.637$), Monsoon season ($R^2 = 0.635$), Summer ($R^2 = 0.611$), and all seasons ($R^2 = 0.614$). Since each set of seasonal models had a higher predictive capability (for each season) than the year-long models, they would be preferred for daily tropospheric ozone forecasts in practice.

As seen in Figure 20, there is a noticeably higher predictive capability of the winter seasonal model, which may be due to the fact that January and February have some of the lowest percentages of rainfall during the year, only receiving about 1.5% of the total annual rainfall each month [19]. The two months also experience some of the lowest temperatures of the year, perhaps limiting ozone formation and making it more dependent on precursor chemicals and conditions than in the summer, when temperatures are always high and ozone formation is more likely and less predictable.

Ozone Forecast Calculator

There are several ways one can put the forecast models to use. One can import a CSV file with the current variables in the same order and format as what the models were trained on, which can be found in the code. The particular row that a forecast is wanted for would then be defined as a variable, which the model would then run on and return a forecast value. For a more instant result, we created a calculator that determines which seasonal model to use and allows the user to input the value of each variable for an immediate 24-hour ozone forecast, as well as the air quality index.



```
>> Starting: 24-Hour Ozone Forecast Calculator <<
What is the month of the current date? (Enter a number, 1-12, or month name) : 4
That's in the Summer.
- - - - - - - - - - - - - - - - - - -
Enter Particulate Matter 10 (PM10) : 324.37
Enter Nitric Oxide (NO) : 11.29
Enter Nitrogen Dioxide (NO2) : 16.39
Enter Nitrogen Oxides (NOx) : 32.72
Enter Ammonia (NH3) : 19.32
Enter Carbon Monoxide (CO) : 12.68

Enter Sulfur Dioxide (SO2) : 7.83
```

```
- - - - - - - - - - - - - - - - - - -
The ozone forecast for 24 hours is :  36.6  µg/m³
The Air Quality Index is :  Good
```

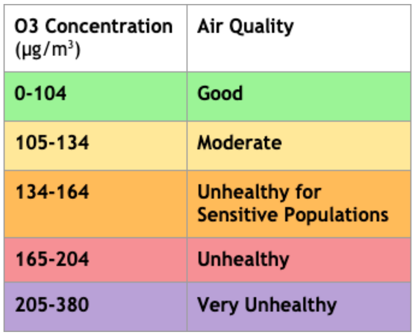
Figure 21- Screenshot of entering input variables and the output of the calculator

In Figure 21, the calculator was tested with a randomly picked row of data from the summer of 2015. The predicted forecast was 36.6 µg/m³, and the actual ozone concentration in 24 hours was 37.4 µg/m³, about a 2% error.

3. Conclusions

Analysis

Tropospheric ozone has become an increasing cause of premature deaths for the past several decades, especially in developing countries such as India. Delhi, India particularly suffers from high ozone levels, experiencing approximately 270 hours each year in the three "unhealthy" ranges of the Air Quality Index (fig. 22) and 30 hours in the most severe "Very Unhealthy" range. To combat this harmful air pollution, we developed a code that generated and analyzed seven different machine learning models to predict ozone concentrations 24 hours in advance. After cleaning and preprocessing one year's worth of hourly data for 4 meteorological variables and 12 pollutant variables, feature selection and hypertuning was conducted to optimize performance of each model. As noted with the similar adjusted R^2 values and VIF values below 10, the models were not overfitted or too complex. The extremely low p-values of the data being used mean that the results of this project are statistically significant, the highest of which being 0.00004 (a 99.996+% chance that the relationships are not errors due to chance). After determining the model with the best annual results (XGBoost, R^2=0.61), it was re-trained on data across 5 years separated by season, which returned significantly better results than when trained over a single year. The best seasonal model was the winter (R^2=0.754), which could reliably predict O_3 concentrations 24 hours in advance, where over 50% of the predictions had a percentage error of less than 10%. The model was also able to predict the exact air quality index 92% of the time, and was able to predict within one index 98% of the time.



O3 Concentration ($\mu g/m^3$)	Air Quality
0-104	Good
105-134	Moderate
134-164	Unhealthy for Sensitive Populations
165-204	Unhealthy
205-380	Very Unhealthy

Figure 22- Ozone Air Quality Index [20]

Challenges and Limitations

This project also faced several challenges- the low initial R^2 values obtained by the year-long models during the summer (which originally included June, July, and August) may have been due to India's rainy monsoon season, which temporarily clears the sky of air pollution but reduces predictability. However, when trained separately, the R^2 values for the same period (June, July, August) during monsoon season improved significantly, from a maximum R^2 of 0.35 to 0.63. Additionally, there may be other influential variables that are not available in the data, so were not included in the model. For instance, the data used did not provide enough information to make use of wind data, but this points to another possible direction in this project- using a more spatial approach to better account for the effects of topography and wind direction on future ozone concentrations, which may increase the accuracy of results.

Another area of improvement comes from the fact that the formation of ozone is a result of sunlight, so predictions would most likely be more accurate using data of solar radiation level rather than temperature. However, many weather stations only record temperature, so we used that for the model instead of solar radiation levels.

Literature Search

This project differs from previous studies in several ways. Most of the other research on the topic of ozone prediction was conducted in the 1990s to early 2000s, but the data used in this project was from 2015-2020. These other studies also tested with fewer input variables (usually 4-6) while our study used 12, analyzing more relationships and patterns. This study also varied in the number of models being trained and tested- other research publications tested only two or three different prediction methods while this study compared seven, including the newest state-of-the-art machine learning algorithms such as XGBoost.

Regarding the accuracy of the model, the best performing model (Winter XGBoost) outperformed (with a higher R^2 value) the models of other previous studies that used multiple regression or neural networks to predict either current or future ozone levels [21-24]. This is a very significant achievement of the project.

Future Directions

The information and models learned in this project can be applied to virtually any city in the world, and there are many possible directions for the real-life implementation of these models. Some possibilities include:

- Simple retraining and testing seasonal models with data from other cities
- Real-time forecasts of ozone concentration 24 hours in advance for government-issued health warnings
- Using this code to try predicting other pollutants

- Retraining the models to predict current ozone levels based on other variables to “fill in” missing data values
- Further expansion of the models with additional variables such as wind, vehicular traffic levels, crop burning, industrial activity, and anything else that could improve ozone predictions
- Comparison of ozone correlations with weather and pollutant variables of different cities

….and much more. Air pollution is a damaging problem which affects many people. It is our hope that our project can help to help combat this ever-increasing issue.

Summary

The results of this study showed that machine learning has great promise in the field of air pollutant prediction and highlighted different patterns and relationships that are essential to understand in order to know how to regulate pollutants and warn populations about high ozone concentrations. By training and testing seven machine learning algorithms, we determined that the model with the best performance was XGBoost, and it had better results when trained seasonally rather than annually. The models developed in this project outperformed models made in previous studies on the topic and have identified several possibilities for further research to improve predictive capabilities. In this way, we were able to create satisfactory machine-learning models to predict ozone levels 24 hours in advance in a developing country on the other side of the world.

Acknowledgements

- Thanks goes out to Mark Petersen, Jenny Juarez, Ashli Johnston, and Lauren Huey. Mark Petersen mentored Eliana in the basics of Python and data analysis, starting in August of 2020. He also helped with the development of the research topic and how to go about researching the subject. Jenny Juarez proofed the final research report. Ashli Johnston and Lauren Huey are the sponsors of our school's science club and have always offered support when needed.

Academic Biographies

- **Eliana Juarez**: I am the team captain of Team 1 and am a 10th grader at V. Sue Cleveland High School. Around August last year, I began to learn Python to compete in the Supercomputing Challenge and have come a long way since. I plan on continuing to pursue research and computer science in the future as a career, and am particularly interested in environmental and bioinformatic sciences.

- **Graciela Rodríguez**: I am a freshman at V. Sue Cleveland High School. My role in this project was primarily focused in writing our various reports. I have a wide variety of interests, including reading, playing musical instruments, filming and editing videos, amateur astronomy, and learning languages. I haven't settled on a course in my life because I have so many interests, but some of my career choices include law and STEM fields.

- **Sofia Juarez**: I helped with the written report, and I am in 9th grade at V. Sue Cleveland high school. I am most interested in learning languages and my goal is to become a polyglot to be able to have job opportunities around the world, possibly as a translator. I am very interested in learning about the world to be able to travel and live in many places. I plan on continuing with my language studies in Spanish and French and hopefully more in the future.

Bibliography

1. Air pollution- World Health Organization. (n.d.). Retrieved from https://www.who.int/health-topics/air-pollution
2. Air Quality - Ozone and Your Health. (2019, September 04). Retrieved December 11, 2020, from https://www.cdc.gov/air/ozone.html
3. Ozone Effects on Plants. (n.d.). Retrieved from https://www.nps.gov/subjects/air/nature-ozone.htm
4. Photochemical smog. (2018). Retrieved April 3, 2021, from https://energyeducation.ca/encyclopedia/Photochemical_smog
5. Ground-level ozone basics. (2021, April 1). Retrieved April 2, 2021, from https://www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics
6. 6. Saxena, Pallavi & Sonwani, Saurabh. (2019). Secondary Criteria Air Pollutants: Environmental Health Effects. 10.1007/978-981-13-9992-3_4.
7. Air pollution in India: Earth.org - Past: Present: Future. (2020, August 17). Retrieved April 2, 2021, from https://earth.org/data_visualization/air-pollution-in-india/
8. India State-Level Disease Burden Initiative Air Pollution Collaborators. (2020). Health and economic impact of air pollution in the states of India: The Global Burden of Disease Study 2019. The Lancet Planetary Health, 5(1). doi:10.1016/S2542.5196
9. Deutsche Welle (2019, January 1). India: Smog causes health emergency during merkel Visit. Retrieved April 2, 2021, from https://www.dw.com/en/india-smog-causes-health-emergency-during-merkel-visit/a-51083303
10. Nongkynrih, B., Gupta, S., &amp; Rizwan, S. (2013). "Air pollution in Delhi: Its magnitude and effects on health". Indian Journal of Community Medicine, 38(1), 4. doi:10.4103/0970-0218.106617
11. Rajarathnam, U., Sehgal, M., Nairy, S., Patnayak, R. C., Chhabra, S. K., Kilnani, Ragavan, K. V., & HEI Health Review Committee (2011). Part 2. Time-series study on air pollution and mortality in Delhi. Research report (Health Effects Institute), (157), 47–74.
12. Pande, J. N., Bhatta, N., Biswas, D., Pandey, R. M., Ahluwalia, G., Siddaramaiah, N. H., & Khilnani, G. C. (2002). Outdoor air pollution and emergency room visits at a hospital in Delhi. The Indian journal of chest diseases & allied sciences, 44(1), 13–19.
13. Kumar, S., Joshi, L., &amp; Kumar, P. (2015). Socioeconomic and Environmental Implications of Agricultural Residue Burning: A Case Study of Punjab, India. Springer India.
14. Central Pollution Control Board. (n.d.). Retrieved from https://cpcb.nic.in/automatic-monitoring-data/

15. Visual Crossing Weather API Documentation (visual-crossing-corporation-visual-crossing-corporation-default). (n.d.). Retrieved from https://rapidapi.com/visual-crossing-corporation-visual-crossing-corporation-default/api/visual-crossing-weather
16. Chemistry in the Sunlight. (n.d.). Retrieved from https://earthobservatory.nasa.gov/features/ChemistrySunlight/chemistry_sunlight3.php
17. Kurniawan, D. (2020, July 01). Multiple linear regression for manufacturing analysis. Retrieved April 3, 2021, from https://towardsdatascience.com/multiple-linear-regression-for-manufacturing-analysis-c057d4af718b
18. Climate of India. (n.d.). Retrieved April 3, 2021, from https://www.newworldencyclopedia.org/entry/Climate_of_India
19. Gajinkar, A. (2019, October 25). Exploratory Data Analysis of Indian Rainfall Climate Data. Retrieved from https://medium.com/@anusha.gajinkar/exploratory-data-analysis-of-indian-rainfall-data-f9755f2cc81d
20. AQI Basics. (n.d.). Retrieved from https://www.airnow.gov/aqi/aqi-basics/
21. Abdul-Wahab, S., Bouhamra, W., Ettouney, H., Sowerby, B., &amp; Crittenden, B. D. (1996). Predicting ozone levels. Environmental Science and Pollution Research, 3(4), 195-204. doi:10.1007/bf02986958
22. Yi, J., &amp; Prybutok, V. R. (1996). A neural network model forecasting for prediction of daily MAXIMUM ozone concentration in an industrialized urban area. Environmental Pollution, 92(3), 349-357. doi:10.1016/0269-7491(95)00078
23. Käffer, M. I., Domingos, M., Lieske, I., &amp; Vargas, V. M. (2019). Predicting ozone levels from CLIMATIC parameters and leaf traits OF Bel-W3 tobacco VARIETY. Environmental Pollution, 248, 471-477. doi:10.1016/j.envpol.2019.01.130
24. Elkamel, A., Abdul-Wahab, S., Bouhamra, W., &amp; Alper, E. (2001). Measurement and prediction of ozone levels around a heavily industrialized area: A neural network approach. Advances in Environmental Research, 5(1), 47-59. doi:10.1016/s1093-0191(00)00042-3