Final Project- Air Quality Prediction using Machine Learning

Guruprasad Velikadu Krishnamoorthy and Hemalatha Subbiah

College of Science and Technology, Bellevue University

DSC630-T302: Predictive Analysis

Professor. Fadi Alsaleem

February 26, 2024

Contents

Air Quality Prediction using Machine Learning

Introduction:

Air pollution has become a significant concern in the modern world. The onset of the Industrial Revolution in the 19th century marked a pivotal moment, as the invention of new machinery and technologies led to increased emissions of gases from factories and pollutants from automobiles, leaving an enduring impact on the Earth's environment. (Kiger, 2021). On average, it is estimated that a person inhales about 2000 gallons of air daily. (Association, n.d.) Hence, it is essential that the air we breathe is of good quality. The polluted air, when inhaled, gets straight into our lungs, then enters the bloodstream, and can cause more damage to internal organs such as the brain, heart, etc., and young children are the most affected ones.   In this project, we aim to delve into the effects of air pollution and explore methods to combat it, focusing on the predictive capabilities offered by machine learning algorithms.

Problem statement:

The Air Quality Prediction problem involves forecasting the concentration of air pollutants or overall air quality index (AQI) at specific locations over a future time period. This is crucial for various applications such as public health management, environmental monitoring, urban planning, and pollution control.

The aim of the project is to develop a machine-learning model capable of predicting the Air Quality Index (AQI) based on various environmental parameters and pollutant concentrations. The model should be capable of providing accurate AQI predictions for future time points, allowing for early detection of potential air quality issues.

Why the Problem is important

According to a recent study, air pollution-related ailments are the fourth largest contributing factor to premature deaths, and about 4.5 million deaths around the world are reported to be related to air pollution. Some air pollutants such as mercury, Lead, and benzene can cause several health issues and, in some cases, even death. (10 Things You Never Knew Could Cause Lung Cancer, n.d.) In New Delhi, India, where the air quality is ranked among the worst, people reported that it felt like breathing poison during extreme smog conditions. The level of PM2.5 pollutant, which is small enough to enter the bloodstream, was reported to be 25 times more than the limits recommended by the World Health Organization in the city. ("Like Breathing Poison": Children in India's Delhi Hit Hard by Smog, n.d.). Though some measures were taken in the last decade to bring awareness to people about the impacts of air pollution, it was too little, and it will be too late if all the countries around the globe don't work together to solve this looming Problem. If the current trend continues, in the future, people may be forced to pay for clean, breathable air, making it an absolute priority to take measures to clean up the air and provide a sustainable and healthy environment for our future generations.

Who would be interested in solving the Problem?

Various stakeholders, including government agencies, healthcare organizations, environmental groups, industries, urban planners, researchers, and the public, are interested in solving the problem of air pollution prediction using machine learning algorithms. They are motivated by concerns such as protecting public health, preserving the environment, complying with regulations, advancing knowledge, and improving quality of life. Collaborative efforts among these stakeholders are essential to develop effective solutions that mitigate air pollution and its impacts.

Source of the Data and why is the Data useful to solve the problem?

The data is extracted from the Environmental Protection Agency website. (Download Files | AirData | US EPA, 2015). A total of 9 datasets were used to predict the air quality in this project, such as Ozone concentration datasets, SO2, NO2, and Carbon Monoxide concentration datasets; datasets of pollutants such as PM2.5 and PM10; datasets containing meteorological data such as Temperature, Pressure, Humidity, etc. All these datasets are for the year 2022. Some of the commonly used fields in the dataset are listed below:

- ***Columns to identify the site location where the readings were measured**:*
    - State code, County Code, Site Number, Latitude, Longitude, Local Site name, Address, State Name, County Name, City Name.
- ***Columns to identify the Pollutant/ Gases/ the Metrological quantity**:*
    - Parameter Name: Represents the Parameter of the Pollutant /Particulate/Toxin/Meteorological measure.
    - Parameter Code: Unique code assigned to the parameter describing the Pollutant /Particulate / Toxin /Meteorological measure.
    - Units of Measure: The unit in which the parameter is measured.
    - Observation Count: Number of observations captured on the specified date in the site location.
    - Arithmetic Mean: The Mean value of the quantity of the parameter captured on the given date at the given site.
- ***Column to identify the Air Quality**:*

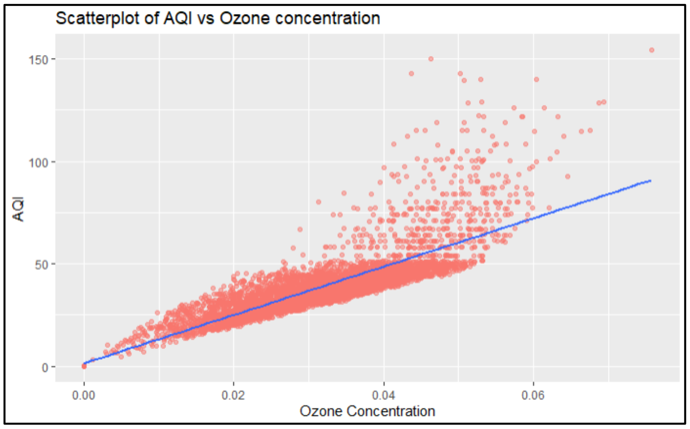
- AQI: Represents Air Quality Index value measured on the specified date at the specified site location.

- ***Date fields***:
  - Date Local: Date when the parameter was measured and recorded.

- ***Columns for Pollutants***:
  - Ozone (O3): Concentration of ozone in the air
  - Carbon Monoxide (CO): The Concentration of carbon monoxide in the air.
  - Nitrogen Dioxide (NO2): Concentration of nitrogen dioxide in the air.
  - Sulfur Dioxide (SO2): Concentration of sulfur dioxide in the air.

Data Exploration and Visualizations

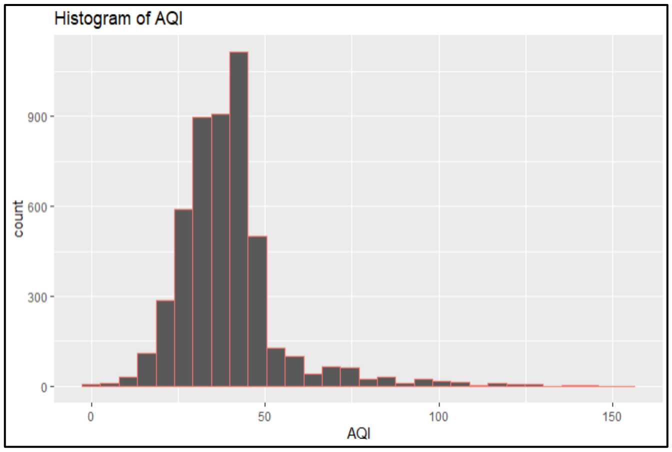
As most of the features used in the project are numerical, scatter plots to show the relationship between the pollutants or harmful gases versus the AQI (Air Quality Index) can be a useful visualization to understand the relationship between the two.



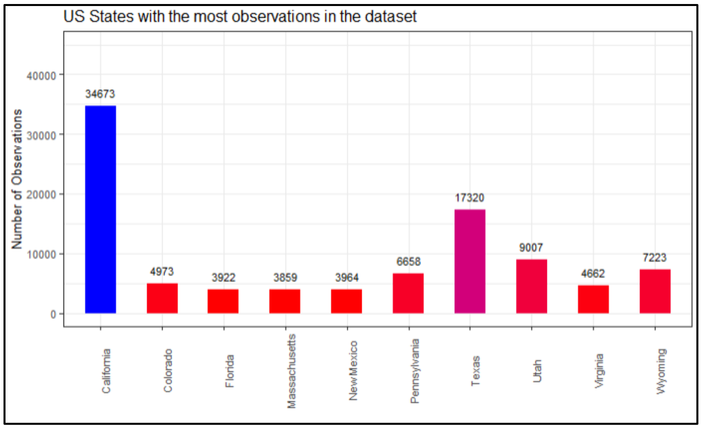
(Figure 1: Scatter Plot of AQI vs Ozone)

The scatter plot (Figure1) of AQI vs Ozone indicates that there is a positive correlation between the two and most of the AQI is between 20 and 50.

Also, a histogram (Figure 2) is plotted on the AQI to identify the outliers, and based on the nature of the outlier, a decision can be made to either eliminate them or impute the data.



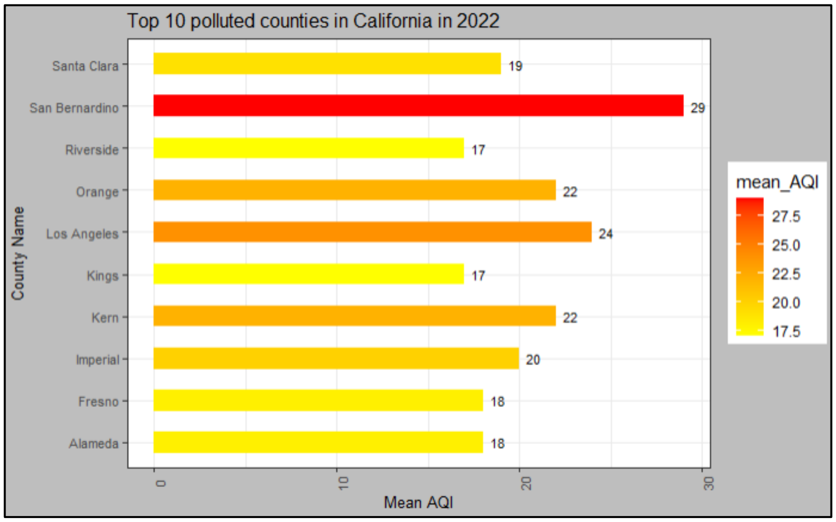
(*Figure 2: Histogram of AQI)*



(Figure 3: Bar plot of most Observations by state)

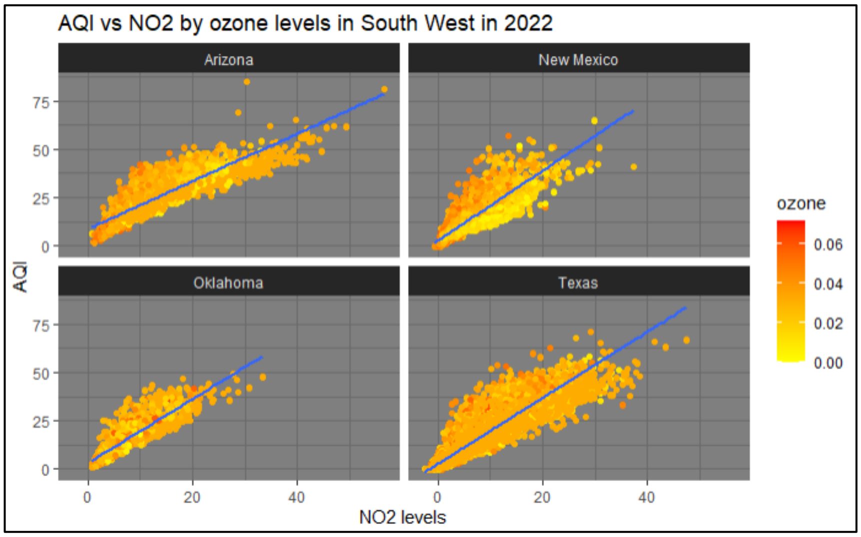
Figure 3 represents bar plot showing the distribution of the data across the US states has been plotted (Figure 3). The plot indicates that most observations in the dataset are from California and Texas followed by Utah and Wyoming.

Figure 4 represents a bar plot that contains the most polluted counties in California in 2022 and their average AQI values. San Bernardino was the most polluted followed by Los Angeles counties based on the NO2 concentration levels.



(Figure 4: Most Polluted counties in California)

Figure 5 represents multiple scatter plots of NO2 concentration versus AQI colored based on ozone values. Each subplot represents a state in the Southwest region. All 4 states represent a similar trend with ozone concentration uniform, though Oklahoma and Texas had higher Ozone levels.



(Figure 5: AQI vs NO2 Concentration by Ozone levels in Southwest)

The Process of Data Preparation:

The data for this project was collected by combining multiple CSV files each containing information about the pollutant concentration, meteorological data, etc. Each row represents information about the pollutant concentration and the Air Quality Index for a location in the US on a given day and has additional details such as state, county, pollutant description, etc.

Data format Conversion

After creating data frames for each dataset, data conversion was performed by converting some character columns to Factors and Dates.

Renaming the features and merging the datasets

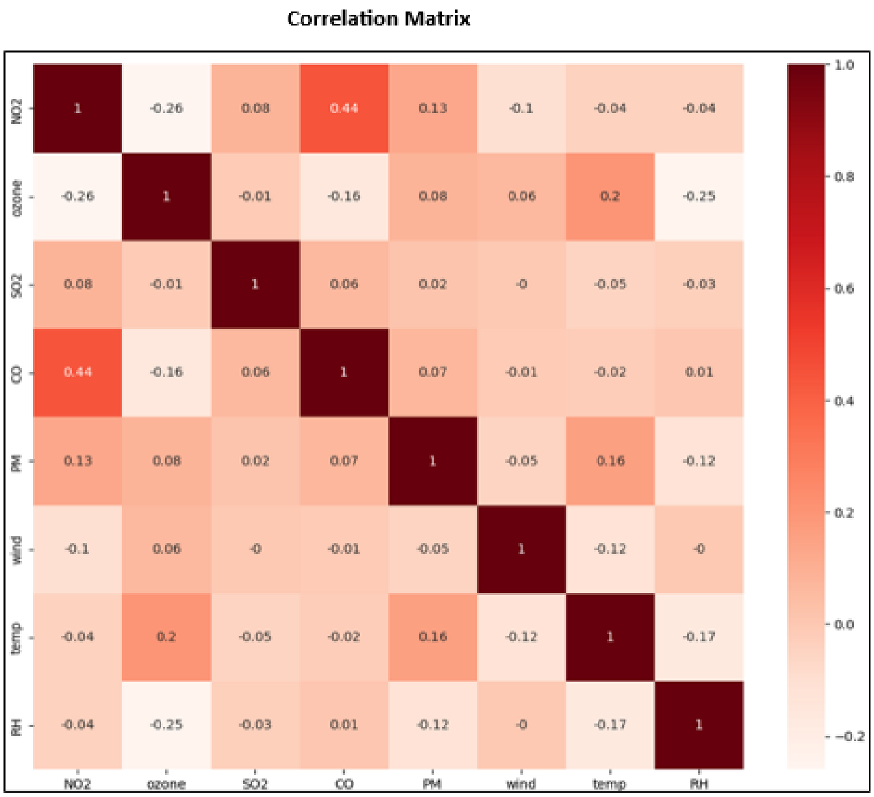
The field names were renamed for easier computations by removing spaces in the column names and replacing them with underscores. To prepare the final data, the Dataset containing NO2 data was joined with all other datasets based on common columns such as State ID, County ID, Site ID, and the Date of Observation.

Null Handling and Duplicate Checks:

The nulls were then handled by replacing them with Median values for the column. The dataset was then checked for duplicates in the key columns and only the distinct values were retained.

Feature Reduction based on the Correlation Matrix:

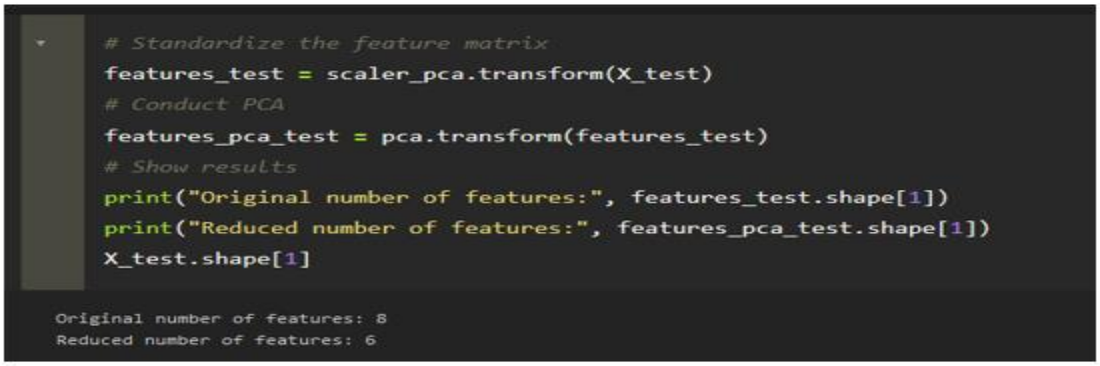
The correlation matrix (Figure 6) was leveraged for analyzing and identifying pairs of features with high correlation coefficients (close to 1 or -1) and the features with high correlation were removed to reduce redundancy. Only the features that had a high correlation with the target variable were retained for model building.



(Figure 6: The Correlation matrix of the Air Quality Dataset)

Feature Reduction using the PCA Method:

The PCA- the dimensionality reduction technique was implemented to preserve most of the variability in the data and reduce the number of features in the dataset. Figure 7 is the snippet of code.



```
# Standardize the feature matrix
features_test = scaler_pca.transform(X_test)
# Conduct PCA
features_pca_test = pca.transform(features_test)
# Show results
print("Original number of features:", features_test.shape[1])
print("Reduced number of features:", features_pca_test.shape[1])
X_test.shape[1]
```

```
Original number of features: 8
Reduced number of features: 6
```

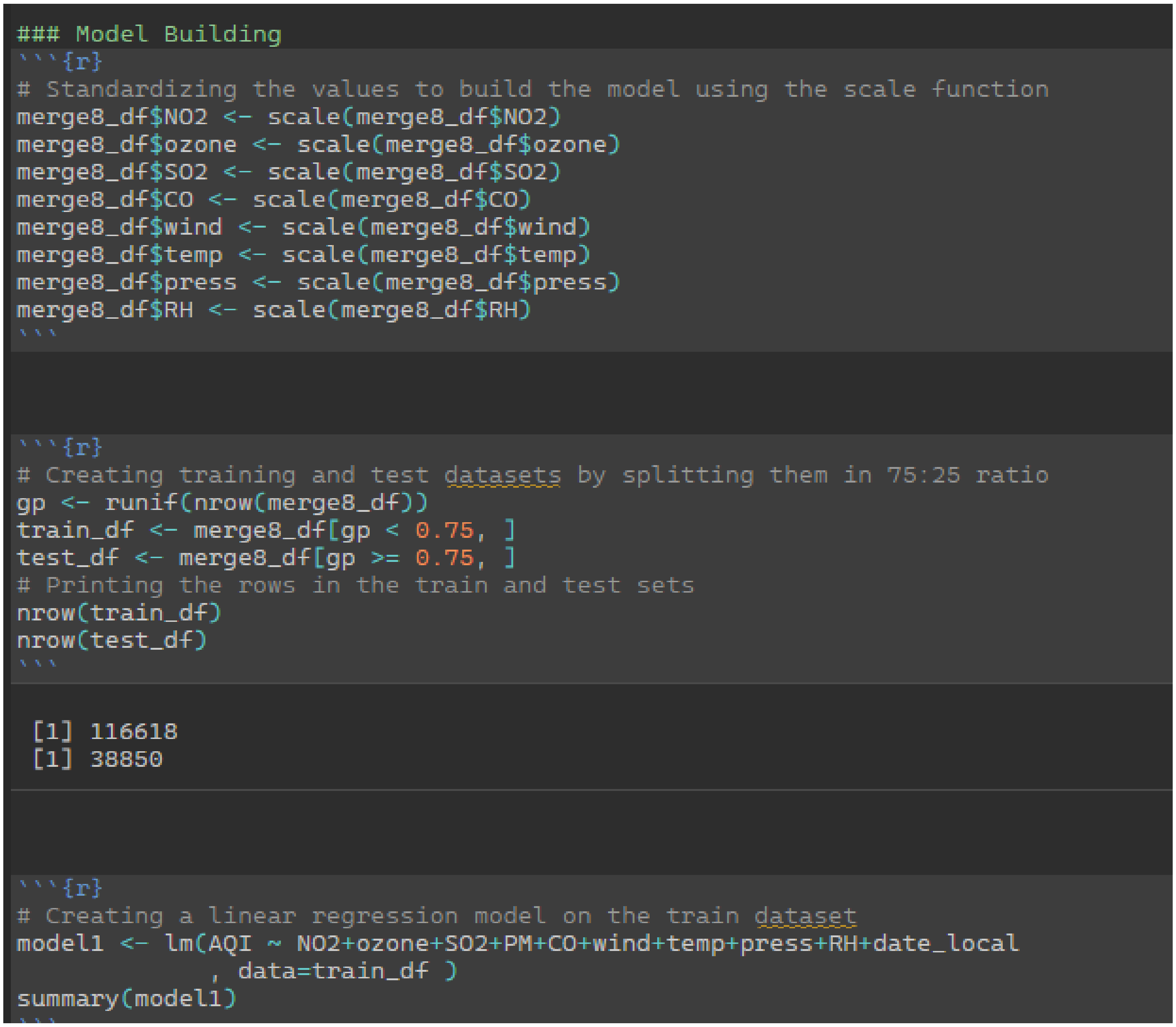
(Figure 7: Screenshot of the PCA Feature Reduction code using Python)

Model Development in R:

As the project is about predicting the Air quality index, which is a continuous Numeric variable, the Regression algorithm was used for building the models. Multiple models were built using both Python and R in this project.

Linear Regression using R:

The final dataset derived by combining multiple datasets was split into Training and Test sets in approximately 75:25 ratio. Also, feature extraction or selection was deployed to reduce the number of features in the dataset. The Linear regression models were then trained using the training dataset and the performance was tested on the test set.



~~~
### Model Building
```{r}
# Standardizing the values to build the model using the scale function
merge8_df$NO2 <- scale(merge8_df$NO2)
merge8_df$ozone <- scale(merge8_df$ozone)
merge8_df$SO2 <- scale(merge8_df$SO2)
merge8_df$CO <- scale(merge8_df$CO)
merge8_df$wind <- scale(merge8_df$wind)
merge8_df$temp <- scale(merge8_df$temp)
merge8_df$press <- scale(merge8_df$press)
merge8_df$RH <- scale(merge8_df$RH)
```
~~~

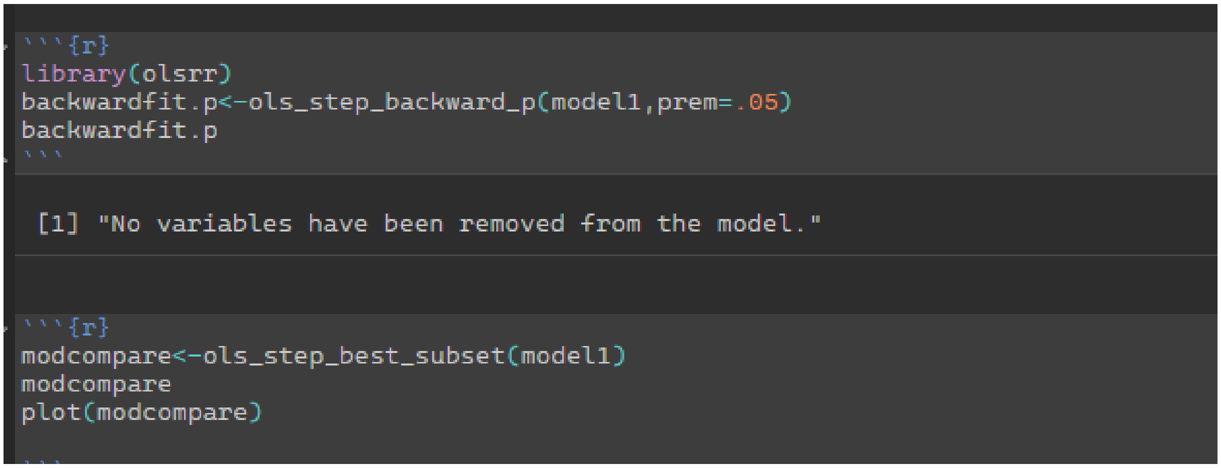
~~~
```{r}
# Creating training and test datasets by splitting them in 75:25 ratio
gp <- runif(nrow(merge8_df))
train_df <- merge8_df[gp < 0.75, ]
test_df <- merge8_df[gp >= 0.75, ]
# Printing the rows in the train and test sets
nrow(train_df)
nrow(test_df)
```
~~~

```
 [1] 116618
 [1] 38850
```

~~~
```{r}
# Creating a linear regression model on the train dataset
model1 <- lm(AQI ~ NO2+ozone+SO2+PM+CO+wind+temp+press+RH+date_local
               , data=train_df )
summary(model1)
```
~~~

(Figure 8: Screenshot of Linear Regression model building code in R)

In R, the backward fit method from the olsrr package was used to identify the less impactful features before building the model. Then ols_step_best_subset from the same package was used to identify the model with better prediction results(R2) from each combination of the features used to build the model. Figure 9 indicates that the model yields the best results with all the features included.



~~~
```{r}
library(olsrr)
backwardfit.p<-ols_step_backward_p(model1,prem=.05)
backwardfit.p
```

 [1] "No variables have been removed from the model."

```{r}
modcompare<-ols_step_best_subset(model1)
modcompare
plot(modcompare)

```
~~~

(Figure 9: Screenshot of olsrr package usage to build models in R)

Model building in Python:

The below regression algorithms were explored as a part of model development:

- Linear Regression
- Decision Tree Regression
- Random Forest Regression
- K Neighbors Regression
- Gradient Boosting Regressor

Before starting the experimentation, the dataset was split into training and testing sets. The training set was used to train the models, while the testing set was used to evaluate the model performance.

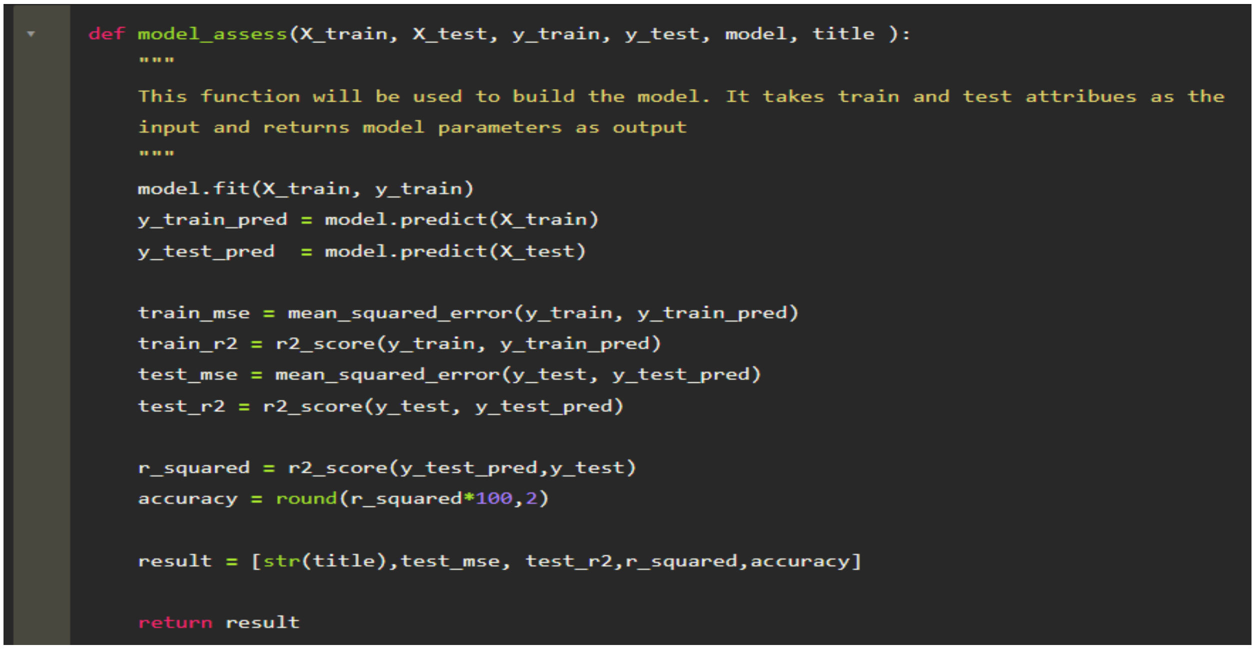
For each regression algorithm chosen, the below steps were followed

a. Train the model on the training data.

b. Use the trained model to make predictions on the testing data

c. Evaluate the performance of the model using appropriate regression evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared (R2), etc.

d. Repeat steps a-c for each algorithm.



```
def model_assess(X_train, X_test, y_train, y_test, model, title ):
    """
    This function will be used to build the model. It takes train and test attribues as the
    input and returns model parameters as output
    """
    model.fit(X_train, y_train)
    y_train_pred = model.predict(X_train)
    y_test_pred  = model.predict(X_test)

    train_mse = mean_squared_error(y_train, y_train_pred)
    train_r2 = r2_score(y_train, y_train_pred)
    test_mse = mean_squared_error(y_test, y_test_pred)
    test_r2 = r2_score(y_test, y_test_pred)

    r_squared = r2_score(y_test_pred,y_test)
    accuracy = round(r_squared*100,2)

    result = [str(title),test_mse, test_r2,r_squared,accuracy]

    return result
```

(Figure 10: Screenshot of the function to build models using Python)

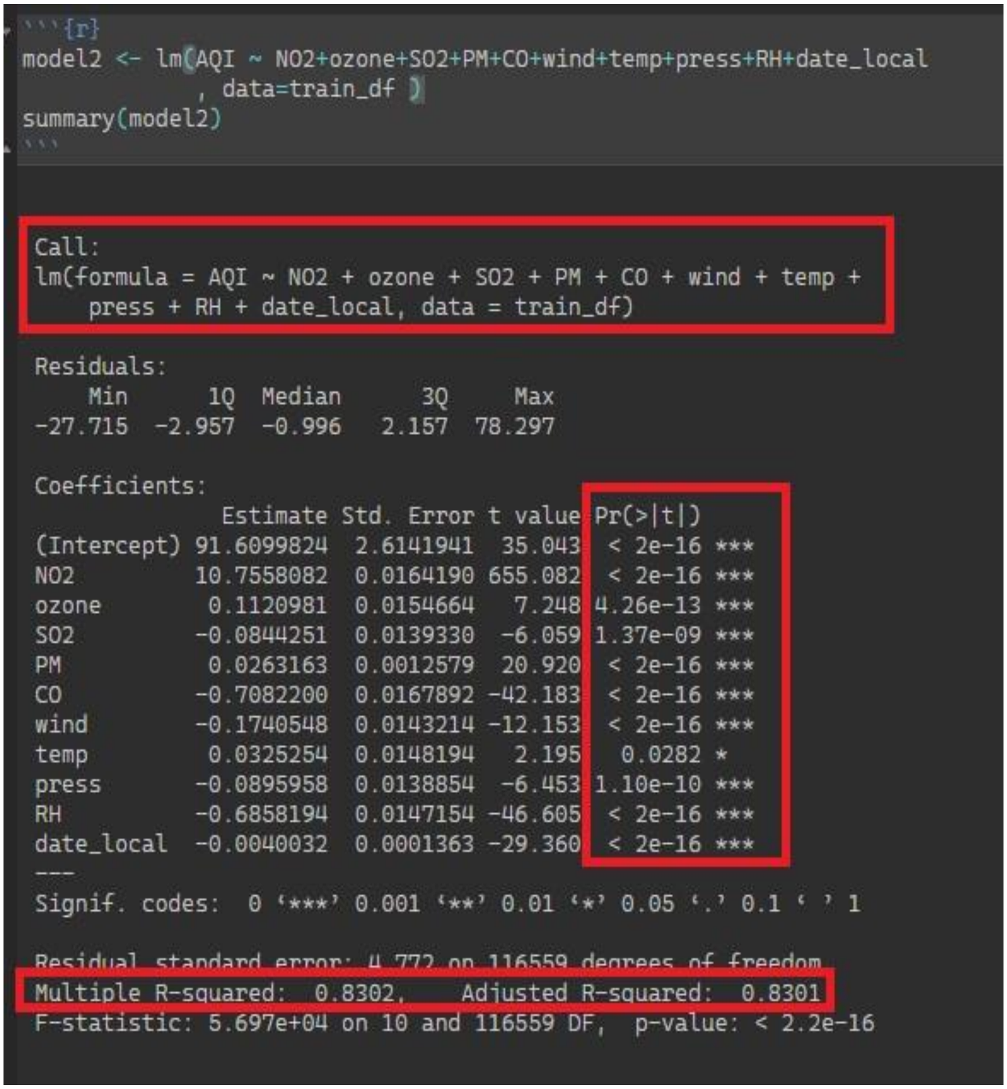
Interpreting the Model results and the Metrics used:

For evaluation of regression models the metrics Mean Squared Error (MSE), R-squared and accuracy were used to assess their performance. These metrics quantify the errors between predicted and actual values, providing a measure of accuracy in terms of the model's ability to make predictions.

Model Results and Metrics evaluation in R:

The results of the linear regression model in R is shown in the figure below.

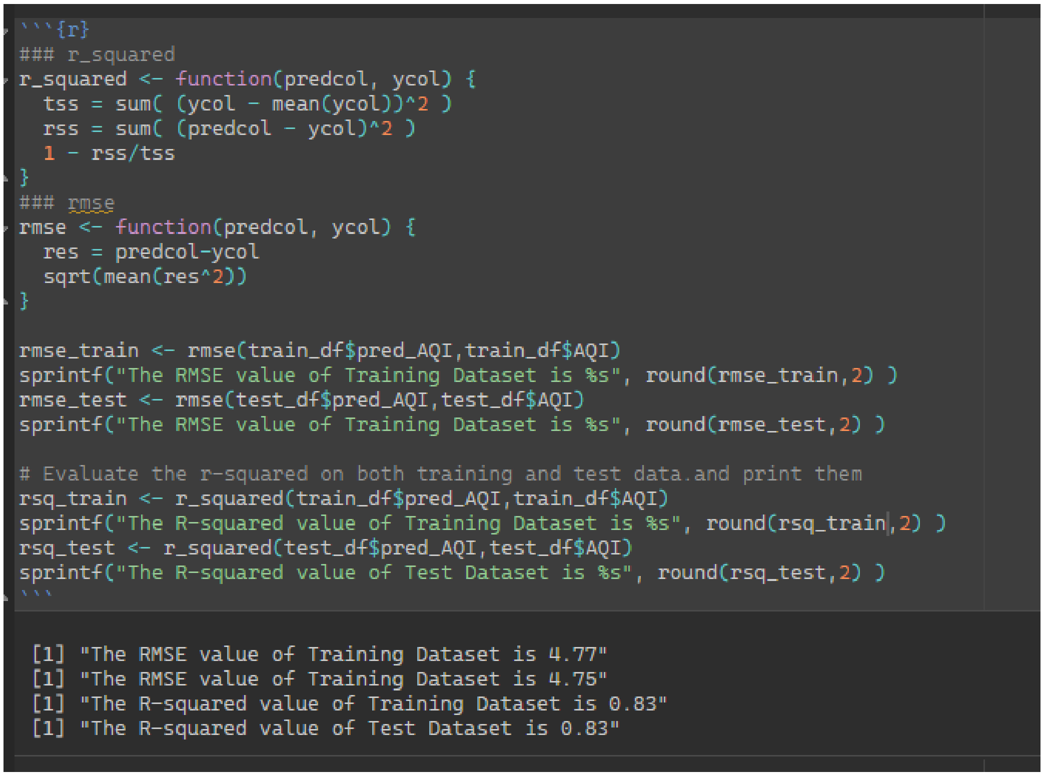
One of the things that stood out in the model results is that the model yielded a high R-squared value of 0.83 which indicates the proportion of the variance in the AQI that is explained by the features of the model. The R-squared and adjusted R-squared values being very close indicates that the additional factors in the model were not being penalized and the model is a good representation of the data.



~~~
```{r}
model2 <- lm(AQI ~ NO2+ozone+SO2+PM+CO+wind+temp+press+RH+date_local
             , data=train_df )
summary(model2)
```

Call:
lm(formula = AQI ~ NO2 + ozone + SO2 + PM + CO + wind + temp +
    press + RH + date_local, data = train_df)

Residuals:
    Min      1Q  Median      3Q     Max
-27.715  -2.957  -0.996   2.157  78.297

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 91.6099824  2.6141941  35.043  < 2e-16 ***
NO2         10.7558082  0.0164190 655.082  < 2e-16 ***
ozone        0.1120981  0.0154664   7.248 4.26e-13 ***
SO2         -0.0844251  0.0139330  -6.059 1.37e-09 ***
PM           0.0263163  0.0012579  20.920  < 2e-16 ***
CO          -0.7082200  0.0167892 -42.183  < 2e-16 ***
wind        -0.1740548  0.0143214 -12.153  < 2e-16 ***
temp         0.0325254  0.0148194   2.195   0.0282 *
press       -0.0895958  0.0138854  -6.453 1.10e-10 ***
RH          -0.6858194  0.0147154 -46.605  < 2e-16 ***
date_local  -0.0040032  0.0001363 -29.360  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.772 on 116559 degrees of freedom
Multiple R-squared:  0.8302,	Adjusted R-squared:  0.8301
F-statistic: 5.697e+04 on 10 and 116559 DF,  p-value: < 2.2e-16
~~~

(Figure 11: Screenshot of the Model results in R)

The Root mean square Error (RMSE) is one of the performance indicators for the model and is consistent for both the training and test datasets. The RMSE, which measures the average difference between the AQI values predicted by the model versus the actual AQI, indicates that we can expect an error of up to 4.77 while using the model predictions.



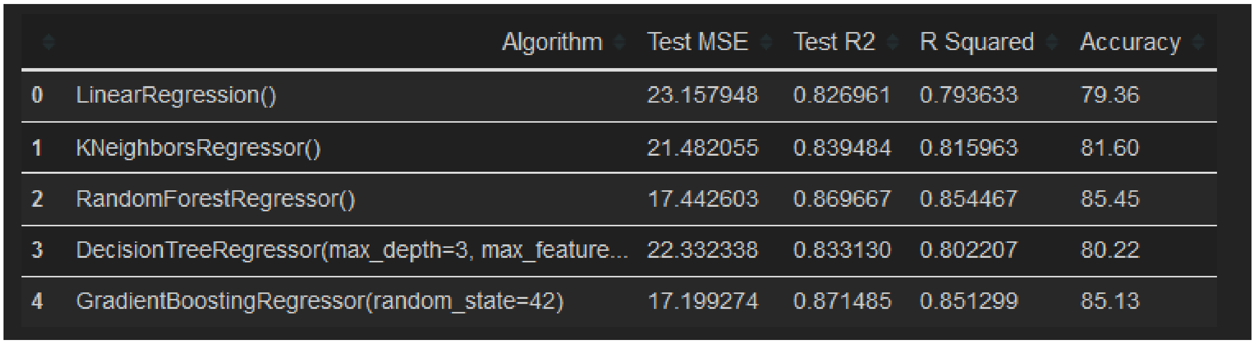
~~~
```{r}
### r_squared
r_squared <- function(predcol, ycol) {
  tss = sum( (ycol - mean(ycol))^2 )
  rss = sum( (predcol - ycol)^2 )
  1 - rss/tss
}
### rmse
rmse <- function(predcol, ycol) {
  res = predcol-ycol
  sqrt(mean(res^2))
}

rmse_train <- rmse(train_df$pred_AQI,train_df$AQI)
sprintf("The RMSE value of Training Dataset is %s", round(rmse_train,2) )
rmse_test <- rmse(test_df$pred_AQI,test_df$AQI)
sprintf("The RMSE value of Training Dataset is %s", round(rmse_test,2) )

# Evaluate the r-squared on both training and test data.and print them
rsq_train <- r_squared(train_df$pred_AQI,train_df$AQI)
sprintf("The R-squared value of Training Dataset is %s", round(rsq_train,2) )
rsq_test <- r_squared(test_df$pred_AQI,test_df$AQI)
sprintf("The R-squared value of Test Dataset is %s", round(rsq_test,2) )
```
~~~

```
[1] "The RMSE value of Training Dataset is 4.77"
[1] "The RMSE value of Training Dataset is 4.75"
[1] "The R-squared value of Training Dataset is 0.83"
[1] "The R-squared value of Test Dataset is 0.83"
```

(Figure 12: Screenshot of the Metrics used in R)

Model Results and Metrics evaluation in Python

The Gradient Boosting Regressor and Random Forest Regressor outperformed the other algorithms based on the given metrics. They had a lower MSE, higher R2, and better accuracy compared to the other algorithms such as Linear Regression, K Nearest Neighbors Regressor, and Decision Tree Regressor. Among the two, the Gradient Boosting Regressor had a slightly lower MSE and higher R2, indicating slightly better performance overall.

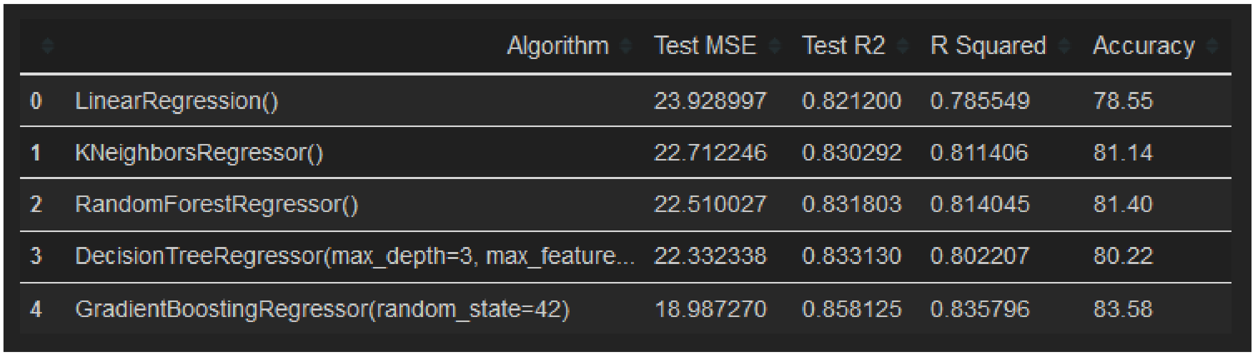


	Algorithm	Test MSE	Test R2	R Squared	Accuracy
0	LinearRegression()	23.157948	0.826961	0.793633	79.36
1	KNeighborsRegressor()	21.482055	0.839484	0.815963	81.60
2	RandomForestRegressor()	17.442603	0.869667	0.854467	85.45
3	DecisionTreeRegressor(max_depth=3, max_feature...	22.332338	0.833130	0.802207	80.22
4	GradientBoostingRegressor(random_state=42)	17.199274	0.871485	0.851299	85.13

(Figure 13: Screenshot of the Model results with no Feature reduction in Python)

Model Results with Pearson's Feature Reduction Method

Figure 14 indicates that the Random Forest Regressor and Gradient Boosting Regressor demonstrate superior performance across all metrics compared to the other algorithms, with the Random Forest Regressor having a slight edge in terms of accuracy. These algorithms are well-suited for regression tasks where accurate prediction is crucial.

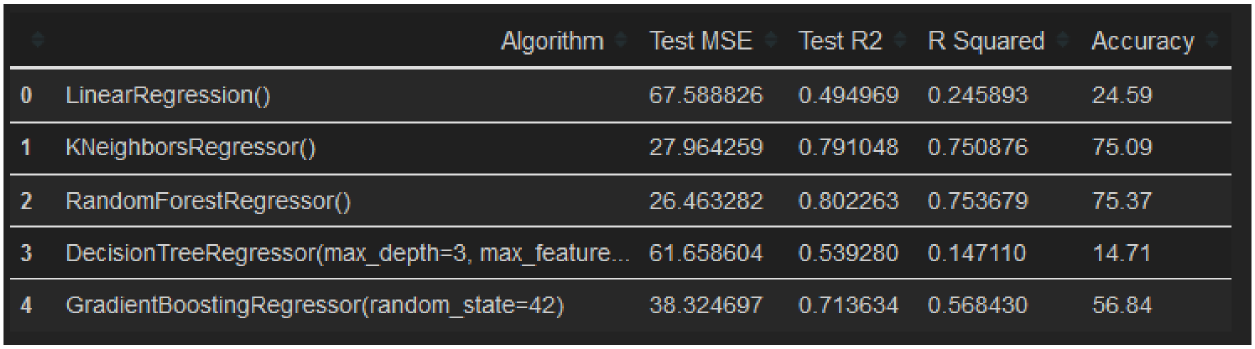


	Algorithm	Test MSE	Test R2	R Squared	Accuracy
0	LinearRegression()	23.928997	0.821200	0.785549	78.55
1	KNeighborsRegressor()	22.712246	0.830292	0.811406	81.14
2	RandomForestRegressor()	22.510027	0.831803	0.814045	81.40
3	DecisionTreeRegressor(max_depth=3, max_feature...	22.332338	0.833130	0.802207	80.22
4	GradientBoostingRegressor(random_state=42)	18.987270	0.858125	0.835796	83.58

(Figure 14: Screenshot of the Model results with Pearson's Feature reduction in Python)

Model Results with PCA Feature Reduction:

The K Nearest Neighbors Regressor performed better compared to the other algorithms, having the lowest MSE, highest R2, and higher accuracy. Linear Regression had the lowest accuracy and relatively poor performance. Random Forest Regressor and Decision Tree Regressor had similar performances but were outperformed by the K Nearest Neighbors Regressor and Gradient Boosting Regressor.



	Algorithm	Test MSE	Test R2	R Squared	Accuracy
0	LinearRegression()	67.588826	0.494969	0.245893	24.59
1	KNeighborsRegressor()	27.964259	0.791048	0.750876	75.09
2	RandomForestRegressor()	26.463282	0.802263	0.753679	75.37
3	DecisionTreeRegressor(max_depth=3, max_feature...	61.658604	0.539280	0.147110	14.71
4	GradientBoostingRegressor(random_state=42)	38.324697	0.713634	0.568430	56.84

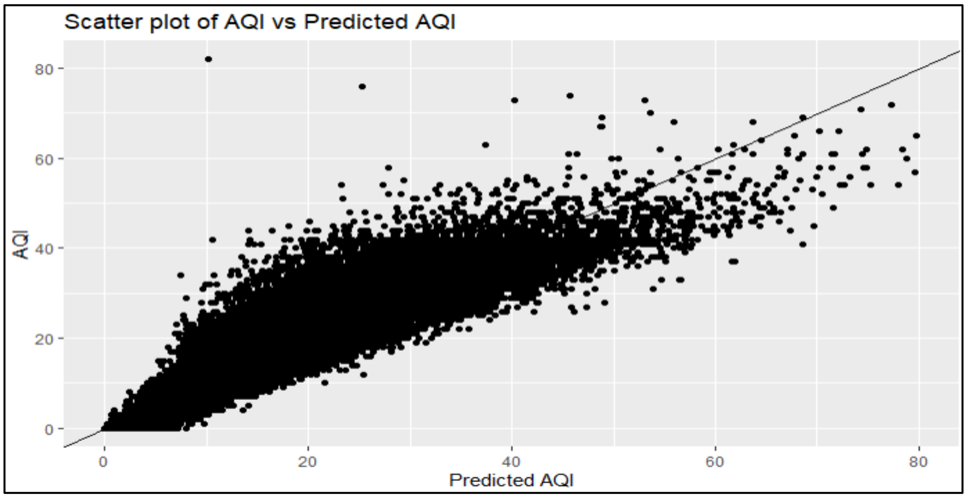
(Figure 15: Screenshot of the Model results with PCA Feature reduction in Python)

Recommendations and Conclusion:

Learnings from the Model results:

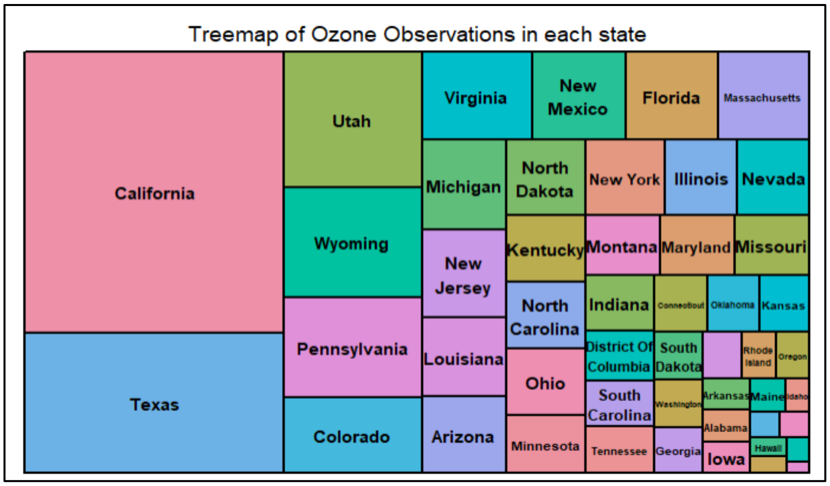
Based on the model results using feature reduction techniques, it was discovered that the model performance with all the features was like the one with Pearson's feature reduction method. Hence the final model was built including all the features in the dataset thus not losing the impact of the pollutants on the AQI.

The results of the linear regression model are plotted in the scatter plot (Figure 16) that compares the AQI versus the predicted AQI values. It indicates the predicted values are not too different from the actual values of AQI in the dataset.



(Figure 16: Scatter plot of the Actual vs Predicted AQI values)

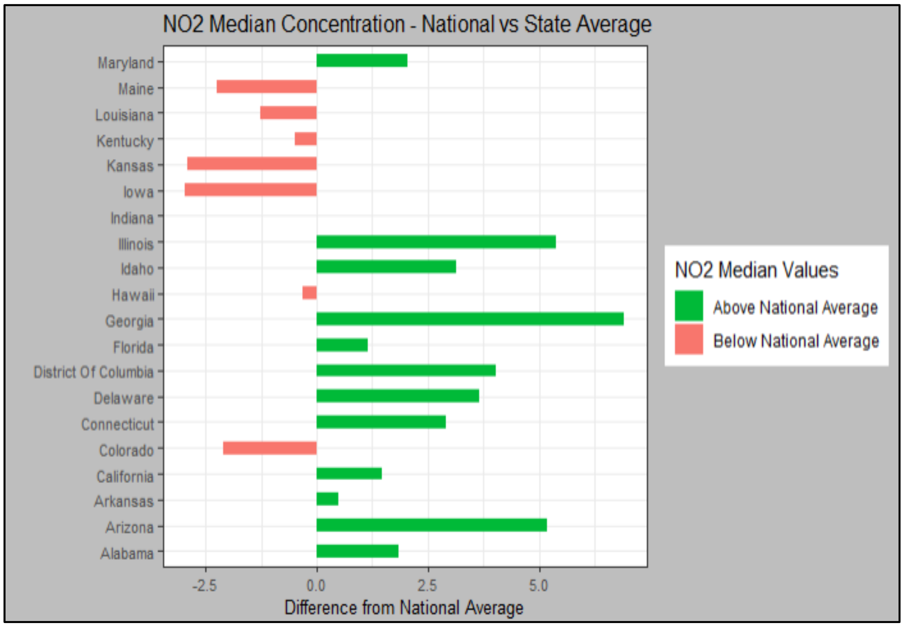
The combined data set contains the concentration of pollutants and metrological parameters such as temperature, pressure, wind speed, etc. for a location on a given day. A few additional graphs were plotted on the final dataset as discussed in this section.



(Figure 17: Tree map of Ozone observations by state)

Figure 17 represents a Tree Map of the number of ozone observations in each state. As expected, California had the highest level followed by Texas and Utah.

Figure 18 represents the comparison of Median values of the national average of NO2 versus the top 20 states in the US. Surprisingly, the states of Georgia, Arizona, and Illinois were in the top states above the national average, while Kansas, Iowa, Colorado, and Maine were the lowest.



(Figure 18: NO2 Median Concentration)

Recommendations:

The model results suggest that NO2 and CO were better predictors for AQI compared to other pollutants. Though ozone was not a strong predictor for AQI as per the model, the effects of ozone cannot be undermined as it can have detrimental effects on Air quality, so it was included during the model building.

What work still needs to be done?

Among all the states in the US, California had the greatest number of observations in the dataset, of which San Bernardino County was found the most polluted. A similar analysis can be extended to other states and regions to find the most polluted counties and measures can be taken to reduce the impacts on the Air quality in those regions. Furthermore, clustering can be done on the dataset to identify the clusters with the higher AQI.

Is the Model ready for Deployment:

Model Deployment is the process in which a Machine Learning model is integrated into a production environment to make practical business decisions based on data. Many Tools such as FastAPI, TFX, Mlflow, Kubeflow ,Sagemaker can simplify the whole process of model deployment, and data scientists can leverage them for Model deployment.

To make this project production ready, there is still some work to be done as indicated in the above sections. The model should be tested with more data to fully understand the impacts of all the pollutants on the Air pollution before the model can deployed in Production.

Ethical Concerns and what could be done to Mitigate them?

Though we don't have control over choosing the air that we breathe, several ethical impacts must be considered while analyzing the impacts of air pollution. In many cases, it is hard to identify the source of the origin of air pollution, reasonable measures can be taken to control it while it cannot be avoided or stopped completely. Some of the ethical implications are listed below:

- Though gas-powered vehicle emissions and industrial smoke have played a significant role in air pollution, they cannot be entirely replaced by sustainable solutions, as they can lead to many job losses affecting many families employed by the manufacturing industries. Care must be taken while publishing the results, keeping in mind the impact it can have on families.
- While determining the acceptable levels of greenhouse gases and pollutants for humans, careful assessment should be made while determining the values, as the acceptable levels for humans may cause significant damage to other ecosystems and species. (Brown, 2001)
- The acceptable levels should also be carefully assessed with international considerations in mind, as the gas emissions and pollutants from the developed countries are no longer a local

issue. These impacts are already seen on the other side of the world, with extreme floods and drought conditions that were not seen in the past.

References:

Kiger, P. (2021, November 9). Seven negative effects of the Industrial Revolution. History. https://www.history.com/news/industrial-revolution-negative-effects

Association, A. L. (n.d.). How Your Lungs Get the Job Done. Www.lung.org. https://www.lung.org/blog/how-your-lungs-work#:~:text=Every%20day%2C%20you%20breathe%20in

Air quality. (n.d.). Transport & Environment. https://www.transportenvironment.org/challenges/air-quality/

10 Things You Never Knew Could Cause Lung Cancer. (n.d.). WebMD. https://www.webmd.com/cancer/ss/slideshow-lung-cancer-surprising-causes

“Like breathing poison”: Children in India’s Delhi hit hard by smog. (n.d.). Al Jazeera. Retrieved December 8, 2023, from https://www.aljazeera.com/gallery/2023/11/10/photos-like-breathing-poison-delhi-children-hardest-hit-by-smog#:~:text=Many%20have%20asthma%20or%20pneumonia

Download Files | AirData | US EPA. (2015). Epa.gov. https://aqs.epa.gov/aqsweb/airdata/download_files.html

Brown, D. (2001). The Ethical Dimensions of Global Environmental Issues. American Academy of Arts & Sciences. https://www.amacad.org/publication/ethical-dimensions-global-environmental-issues

US EPA, O. (2016, August 30). Air Data Basic Information. US EPA. https://www.epa.gov/outdoor-air-quality-data/air-data-basic-information

USA States to region. (n.d.). Www.kaggle.com. Retrieved January 13, 2024, from https://www.kaggle.com/datasets/omer2040/usa-states-to-region