Emissions & Air Quality Report

Daniel Rose, Finn McSweeney, Rachel Walter, Isabelle Hyppolite

02/01/23

Introduction:

For this project, our team wanted to look at emissions and see how they were related to air quality. We also wanted to look at how a country's population size and GDP influenced their emission levels and air quality. For the second part of our project, our team wanted to drill down into the United States data and look at the emissions in big cities, industries causing these emissions, and how all of this relates to air quality. Our team found that there was a strong correlation between a lot of these datasets and after cleaning them and filtering properly, meaningful insights were able to be drawn using the data.

Data:

For the first portion of our project, we compared the emissions and air quality of different countries, we also took a look at GDP and population. The datasets we used were:

1. Emissions by Country – 2002-2022
    a. This dataset showed emission totals by country for each year.
2. World Population Dataset
    a. As the title suggests, this dataset had countries' populations from 2000 to 2022.
3. World GDP (Per Capita, Annual Growth)
    a. This dataset contained many different figures on GDP, like Per Capita GDP and GDP growth, but we focused mainly on the actual GDP. The dataset had each country's GDP from 1960 to 2020.

For the second portion of our project, we focused on the United States. The following datasets were used:

1. Daily Summary Data: Criteria Gasses (NO2)
    a. 5 years of data was used from 2017-2021 and there were a total of 873,611 rows after appending the years. 29 total columns were in this dataset however only certain columns were used in order to create meaningful visualizations.
2. Daily PM10/PM2.5 Speciation:
    a. The exact same steps were done for both the PM10 and PM2.5 datasets as the NO2 data in terms of appending the years and gathering all of the data. Each dataset contained 29 columns. The PM10 dataset ended up having 1,287,157 rows and the PM2.5 dataset had 1,204,651 rows.
3. Emissions by Unit and Fuel Type:
    a. This dataset had a total of 17 columns and 229,853 rows and were yearly summaries of CO2, CH4, and N2O emissions by industry facilities.

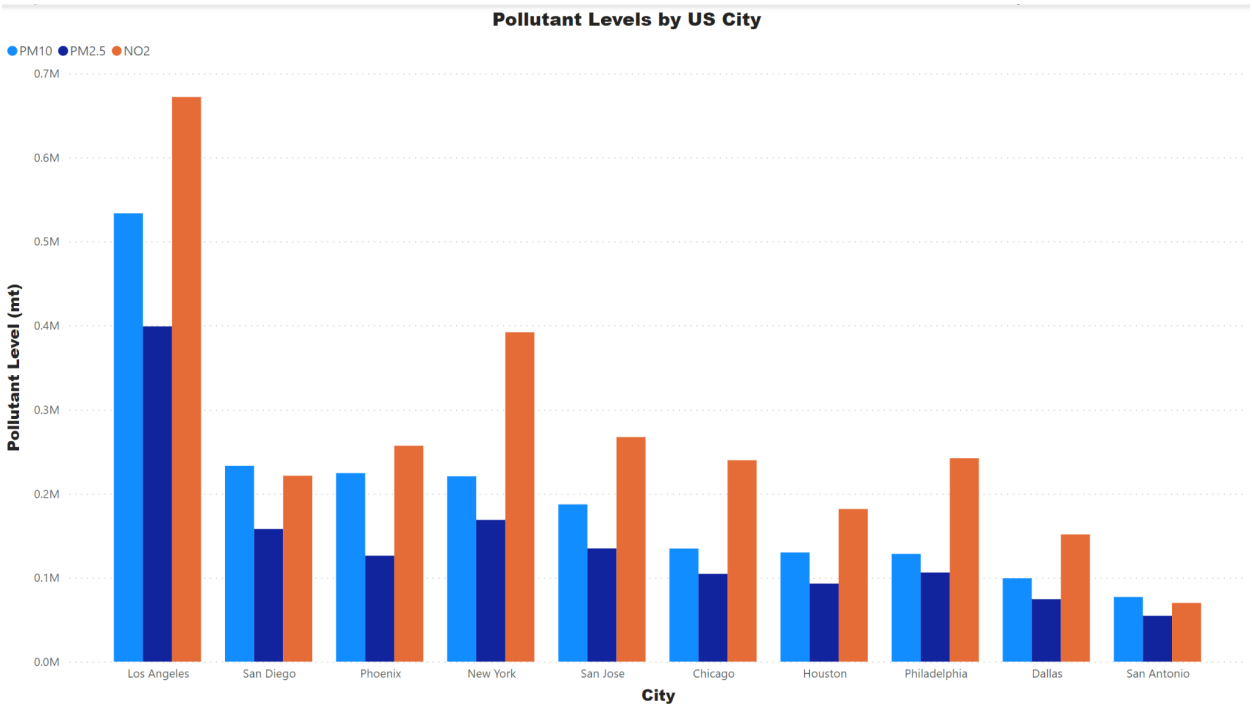
Data Processing:

Due to the fact that the industry dataset were yearly totals and the pollutant datasets were daily averages, the first step in data processing was to get the average of each city for a year. Once that was completed, errors in the data were removed and columns were cleaned up in order to

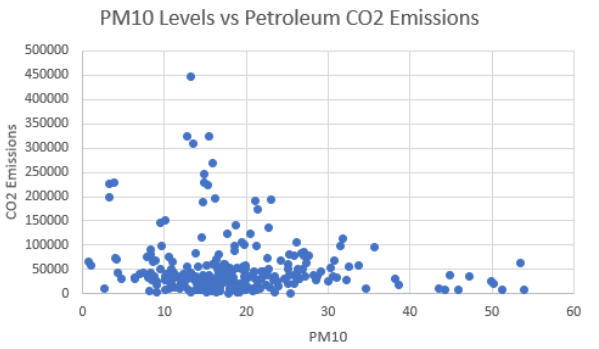
merge these datasets together. Although there were a plethora of columns in all of these datasets, most were not used and after grouping the data by the city and the year, only the pollutant or emission levels were kept in order to have data that was manageable to work with. Once cleaned and grouped, the datasets were merged with each other on inner joins between the city and years and data analysis was then performed.

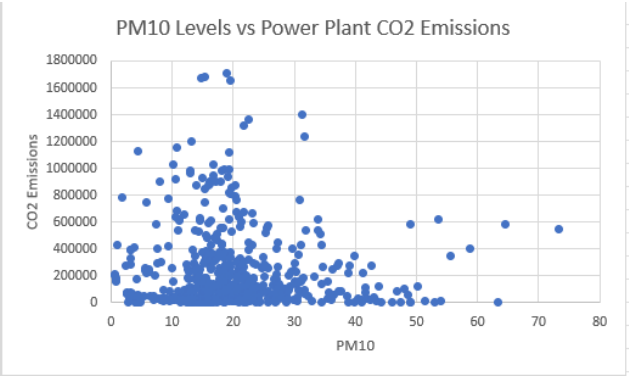
Data Analysis/EDA:

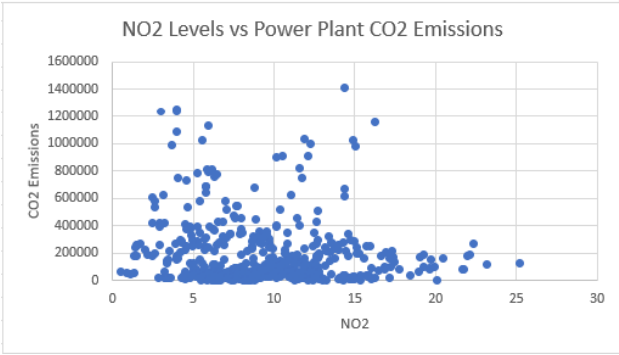
The first question tackled in dealing with USA pollutant levels used live data from the OpenWeather API and the top 10 cities by population were analyzed. The three pollutants mentioned before, NO2, PM10 and PM2.5, were used as the values and this gave our team a start on looking into these cities. The graph of this can be seen below.

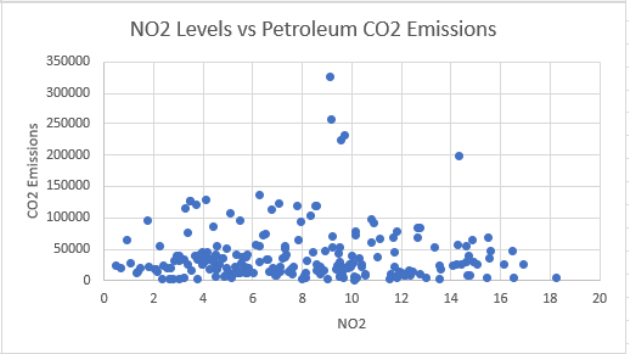


After analyzing this chart, our team wanted to focus on the top industries that have the most carbon dioxide emissions and how that relates to particulate matter and nitrogen dioxide in the air. Power plants and petroleum have the highest co2 emissions and they also had the most amount of data to work with. In addition, methane and nitrous oxide are also highest in power plants and petroleum industries. which is why our team chose these two industries to work with. In looking at this series of scatter plots: the top two representing particulate matter in comparison to either power plants or petroleum facilities and the bottom two representing nitrogen dioxide and its relationship to power plants or petroleum, it is underwhelming to see no clear correlation between any of these.

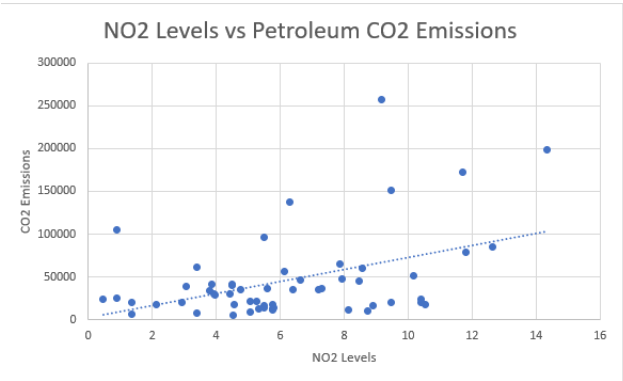


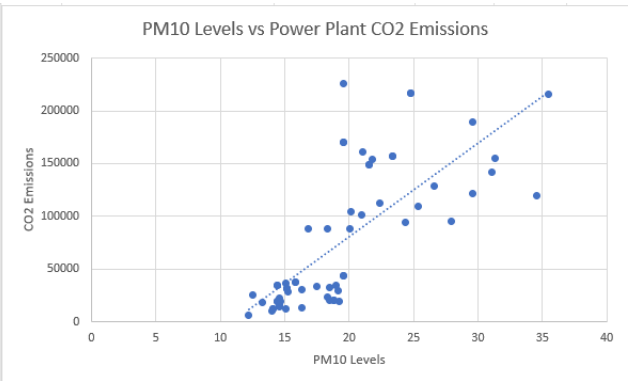




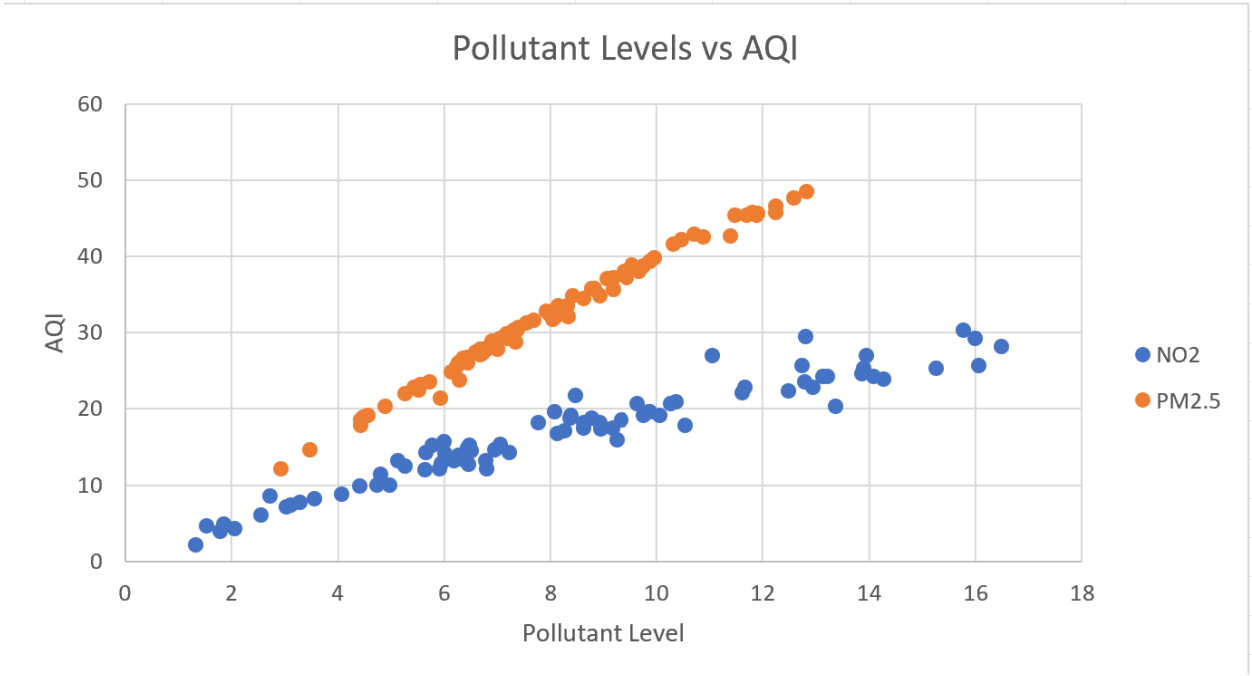


However, when filtering the data using the top 10 cities by population that was shown on the previous slide, trends can be found. Our team has come to the conclusion that particulate matter found in these cities has a direct correlation with power plant emissions. Furthermore there is a slightly positive correlation between the co2 levels produced by petroleum facilities and nitrogen dioxide, however it is not strong enough to draw a direct cause and effect relationship between them.



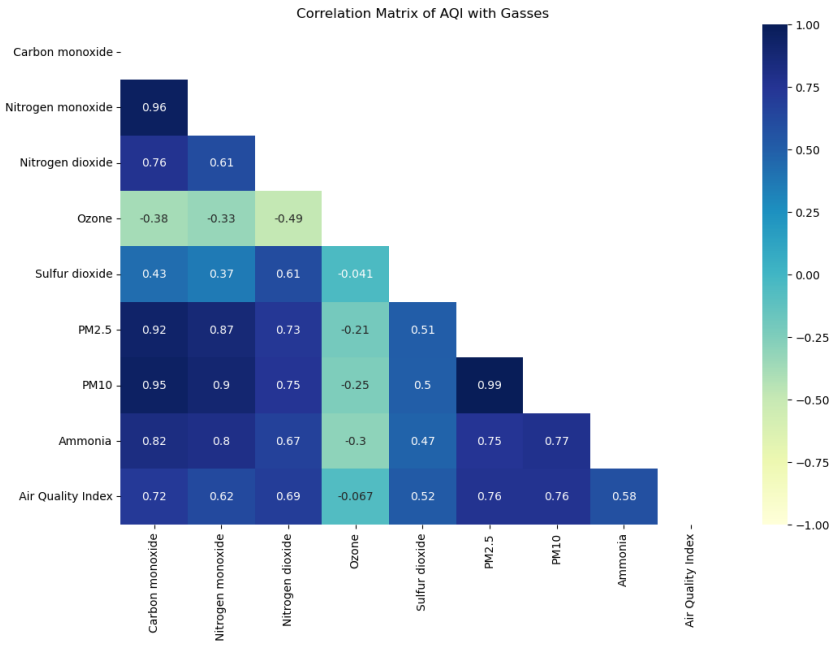


Air quality in relation to these pollutants was also something our team wanted to analyze and as seen in the graph below, both pollutants have a direct impact on air quality across the county. These emissions leading to the increase of pollutants are continuing to rise which is leading to unhealthy air quality across the county.



Machine Learning:

For our machine learning portion, we used data collected from Open Weather's Air Pollution API. When you request historic data from the API you receive hourly readings on air quality, specifically an Air Quality Index rating and the levels of polluting gasses that influence the overall rating. When creating machine learning models you want large amounts of data to train and test the model, so it was advantageous that air quality was recorded for each hour. On the other hand, the data only went back to November 27th 2020. Optimally, we would have preferred for the data to go back farther so that the model could get a sense of any seasonal trends that might be present. Initially we planned on trying to predict Air Quality Index values, but since this index only ranges from one to five we thought that would be a little arbitrary. Instead, we decided to look at which gas seems to influence the air quality rating the most. To do this, we ran a correlation matrix:



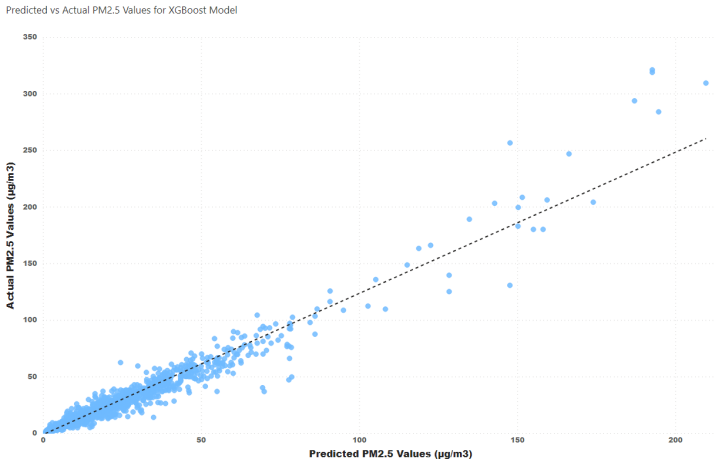
As you can see, out of all the gasses PM2.5 and PM10 shared the highest correlation with the Air Quality Index as they both returned r values of .76. After conducting research on the difference between these two gasses we decided to predict PM2.5 levels instead of AQI. We went with PM2.5 because it is considered more dangerous than PM10. PM2.5 means particulate matter 2.5, so any particles smaller than 2.5 microns. PM10 is made up of larger particles, and is essentially dust, so it can be irritating to your nose and eyes. But because PM2.5 is smaller than PM10, it can get into your lungs or bloodstream, which is why it is considered more dangerous (CDC).

The data from the API is considered time series data, since it is values collected at designated intervals. The traditional machine learning models that we learned about in class are not usually recommended to use with time series data. Instead, data scientists usually use neural networks, which are considered deep learning, or specific time series prediction models. So, we decided we wanted to create and compare two different models, the first one being a traditional model and the second one being one used specifically for time series data. During our research, there were many conflicting articles on the best traditional model to use with time series data, but we decided to go with XGBoost since that was the model we saw being recommended the most. When researching which model to use for the one specifically meant for time series data, three recommendations kept on coming up: LSTM (a neural network), ARIMA (a time series prediction model), and Prophet (an additive model created by Facebook that is often used for time series predictions). Again, there were lots of conflicting opinions on which was best, but we ended up going with LSTM, mostly because we thought it was cool that the model's structure was based on the human brain.

For the XGBoost model, we added time features to serve as our predictors. So we added which hour of the day, day of week, quarter, month, year, and day of year the data comes from. We also added three lags to serve as features, "lag 1" showed the PM2.5 value from an hour before, "lag 2" from two hours before, and "lag 3" from three hours before. For our LSTM

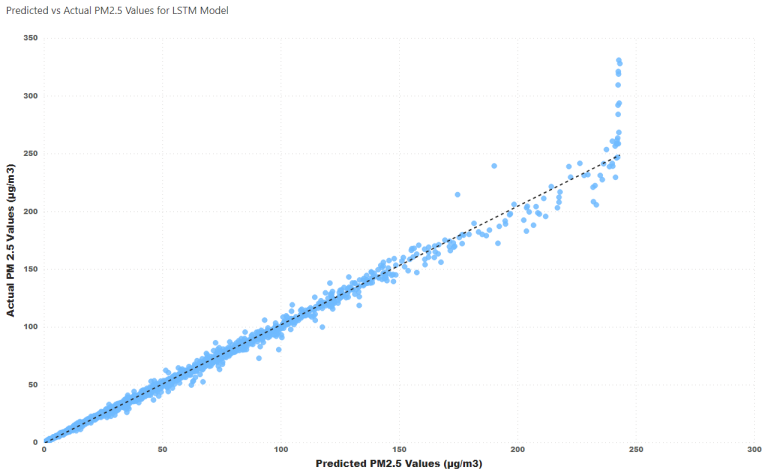
model, we did not have to add any features (we did try to add cyclical features to see if it would improve the model but it did not). Instead, LSTM uses a certain number of values as predictors to predict the next value. So because our window size was eight, the model used values one to eight to predict value number nine, then values two to nine to predict value number ten, and so on.

For the XGBoost model, the hyperparameters we altered were “n_estimators,” which decides the number of trees in your model, “learning_rate,” which essentially decides how fast your model progresses from tree to tree, and “early_stopping_rounds,” which sets when the model should stop running if it stops improving. Because we had a large number of trees, we had a very small learning rate to try and prevent the model from overfitting. We also had “early_stopping_rounds” equal to 50 so that the model would stop running if it stopped improving before it hit 50 trees. For the LSTM model, the hyperparameters we modified were the number of hidden layers, the number of neurons in the LSTM layer, the number of units in the dense layers, the optimizer, the learning rate, the epochs, and early stopping. For the number of neurons, 32, 64, 128, and 256 are all common, we went with 64 because increasing neurons can improve performance but making the model too complex can also increase the chances of overfitting. For the number of hidden layers, that refers to any layer between your input and output layers, in this case we have two since you really only need one or two hidden layers. For the number of units in the dense layer, you usually want it to be between five and ten, so we have it at eight. In terms of the optimizer, we went with Adam since that is what we saw recommended the most, and similar to the last model we kept the learning rate low to avoid overfitting. Epochs refers to the number of times the model runs through the data, we set it at 50 because if you set it higher than that it again increases your chances of overfitting. Lastly, we used early stopping again for this model, and set patience to 5 so that the model would stop running if it did not improve after 5 epochs (Chowdhury).



The first model scored a .94 on the training set and a .90 on the testing set, for XGBoost scoring the model returns the coefficient of determination which is r^2. It ranges from zero to one, and optimally you want it to be closer to one since the closer it is to one, the more the variance is explained by the independent variable. The model performed a little better on the training set than the testing set, but the difference is not large. That is a good sign because if the score is much lower

for the test set it suggests that your model is overfitting. The mean squared error is 40.26, the MSE is the average residual, so the difference between the predicted value and the actual value, squared. You want the MSE to be as small as possible, because the smaller the MSE, the smaller the average difference is between your predicted and actual values.



The second model's r^2 values were .99 for the training set, .99 for the validation set, and .98 for the testing set. The MSE was 3.93 for the training set, .49 for the validation set, and 7.38 for the testing set. As stated earlier, the MSE for the last model's test set was 40.26, so that is a RMSE of 6.37 compared to a RMSE of 2.72. That means that on average, this model was closer to predicting the actual value by about 3. While the first model performed well, the many added features were very necessary since before they were added the model scored a .12 on the testing set. But for the LSTM model, the added features were not necessary to the model's performance because it already takes into account how values change over time unlike the XGBoost model. Even though LSTM was not explicitly created to analyze time series data, this makes it better suited to predicting it than XGBoost.

Conclusion:

To conclude, we found that there is a strong correlation between air quality and emissions across the United States. Using our data we were also able to create two different machine learning models that both have a high accuracy of predicting highly correlated air quality measurements like PM2.5. In the future we could also look at creating models for other parts of the world besides the United States. Since we found that the higher the population is the more emissions the country produces, we would recommend that countries with larger populations attempt to lower their emissions through greener energy sources. Really, we would recommend that to every country regardless of their population, but those with larger populations especially. Now that we know which countries have the highest emissions and the worst air quality, a next step would be to look at how these factors could be affecting their population's health.

Works Cited

Centers for Disease Control and Prevention. (2022, November 21). *Particle Pollution*. Centers for Disease Control and Prevention. Retrieved January 31, 2023, from https://www.cdc.gov/air/particulate_matter.html#:~:text=Coarse%20(bigger)%20particles%2C%20called,or%20even%20into%20your%20blood.

Chowdhury, K. (2021, May 25). *10 Hyperparameters to Keep an Eye on for your LSTM Model - and Other Tips*. Medium. Retrieved January 31, 2023, from https://medium.com/geekculture/10-hyperparameters-to-keep-an-eye-on-for-your-lstm-model-and-other-tips-f0ff5b63fcd4