Applying Machine Learning to Analyze Air Quality's Relationship with Atmospheric Conditions

BUS-395 Data Analytics Capstone

Molly Nelson, Ryan Barry

Spring 2024

I. INTRODUCTION

In recent decades, the issue of air quality has emerged as a concern globally, with implications for public health, environmental sustainability, and socioeconomic well-being. The release of pollutants into the atmosphere poses significant risks to human health, including respiratory and cardiovascular diseases, as well as impacts on ecosystems and climate systems.

The complex interplay between air quality, weather patterns, and climate change has become a topic of increasing interest among researchers, policymakers, and the general public. Understanding the relationships between these factors is essential for devising effective strategies to mitigate air pollution and its associated impacts on human health and the environment.

This project aims to explore the relationships between air quality, weather conditions, and climate change using advanced analytical techniques, including machine learning algorithms, applied to comprehensive datasets. By leveraging machine learning, such as regression and predictive modeling, we hope to discover patterns, trends, and correlations that can provide insights into air pollution and its interactions with weather and climate change.

II. PRIOR RESEARCH

https://www.hindawi.com/journals/jeph/2023/4916267/

A paper aiming to predict Air Quality by N. Srinivasa Gupta uses data from cities in India. The paper discusses the air quality index (AQI), an important indicator of air pollution's short-term health impacts. The research focuses on developing accurate AQI prediction models for Indian cities using data mining techniques, particularly emphasizing the use of the synthetic minority oversampling technique (SMOTE) for dataset balancing. Three predictive models—support vector regression (SVR), random forest regression (RFR), and CatBoost regression (CR)—were evaluated in cities like New Delhi, Bangalore, Kolkata, and Hyderabad. The study finds that random forest regression generally yields the lowest root mean square error (RMSE) and highest accuracy in most cities, especially when SMOTE is applied, highlighting its effectiveness in managing imbalanced data.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10280551/

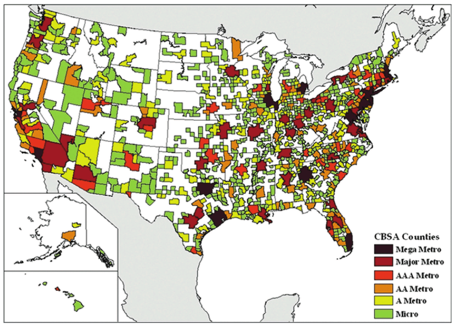
Another article in this field gathers its data from Selangor, Malaysia. This article explores the intersection of rapid urbanization and environmental degradation. It details an air quality predictive model that uses machine learning and deep learning techniques, including AdaBoost, SVR, RF, KNN, MLP regressor, and LSTM. The model incorporates data on pollutants like PM2.5, PM10, O3, and CO, alongside meteorological factors, to predict pollution levels. Results show the LSTM model excelling in predicting PM2.5 and PM10 concentrations, with high accuracy across different urban locations. The study emphasizes the importance of feature optimization, which enhances model interpretability and efficiency by reducing dimensionality. This approach aids policymakers in creating effective pollution control strategies and underscores the role of advanced technologies in sustainable urban planning.

III. DATA COLLECTION

A. AQI DATA COLLECTION

For this project, the Air Quality Index (AQI) data for various U.S. cities was collected through an API provided by the Environmental Protection Agency (EPA). The data includes daily and sometimes hourly samples from early as 1960 to 2024 for the pollutant sulfur dioxide and is identified by state and county codes specific to each city. We wrote a script in Python that made repeated API calls from 1960-2023 for each city. This data was then placed into a csv file.

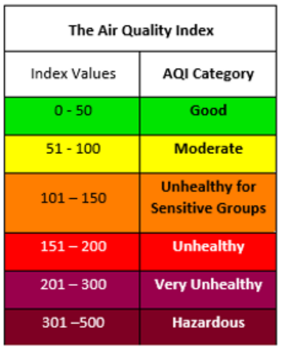
A separate dataset was collected for the climate change analysis. This dataset was also retrieved from EPA but used the daily pre-generated files instead of the API. Because climate change reflects atmospheric conditions on a scale much larger than any one city, we needed a broader scope of data and not exclusive to any one location. The pre-generated data that we chose covered the areas shown in the figure below.



More specifically, core-based statistical areas (CBSA) were utilized. CBSA is a government selected collection of 393 areas that have large population nucleuses or urban areas. We chose this instead of air quality for every county since these areas are more relevant to our project goals. Introducing data for areas that would not reveal patterns or correlation would add noise to the data. Since this data was not accessible through an API, but instead their website, we needed to make a that would scrape the data. By downloading each file and then joining it into a collective CSV, we were able to amass a 18 million row dataset of daily Ozone levels from 1980 to 2023 for each of the 393 locations.

B. AQI DATA PREPROCESSING

After collecting the raw data, we then performed some preprocessing to clean up the datasets and prepare them to run through the models. We first removed rows with missing values in the target feature column, the sample_measurement. We also removed irrelevant or redundant features such as 'site_number', 'datum', 'sample_duration_code', 'date_local', 'time_local', 'date_of_last_change', and 'cbsa_code'. The date of each data point was also changed. The 'date_gmt' column was parsed into datetime format, extracting 'year', 'month', 'day', and ‘time’ into separate columns. This would be useful later on when the dataset would be merged with the openweather data.



The Air Quality Index	
Index Values	AQI Category
0 - 50	Good
51 - 100	Moderate
101 – 150	Unhealthy for Sensitive Groups
151 – 200	Unhealthy
201 – 300	Very Unhealthy
301 –500	Hazardous

The original dataset included a sample measurement of Sulfur dioxide found in the air. We used the following formula from the U.S. Environmental Protection tv Agency to calculate its corresponding AQI value:

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{HI} - BP_{Lo}}(C_p - BP_{Lo}) + I_{Lo}.$$

Where I_p = the index for pollutant p
C_p = the truncated concentration of pollutant p
BP_{Hi} = the concentration breakpoint that is greater than or equal to C_p
BP_{Lo} = the concentration breakpoint that is less than or equal to C_p
I_{Hi} = the AQI value corresponding to BP_{Hi}
I_{Lo} = the AQI value corresponding to BP_{Lo}

A new 'aqi' column was created by applying this AQI calculation across all rows, ensuring a comparable air quality index represented every data point. This will be the target variable for the continuous models. We also imputed a new AQI category column based on the levels defined by the EPA. This could be used as the target variable for any categorical models we want to use.

Within the climate change AQI dataset, we realized the importance of seasons and the cyclical pattern. For the models to properly recognize this, instead of using the date as a numerical value, string, or object, we opted for using cosine and sine to represent where in the cycle of a year the date is. By finding what day in the year divided by 365 or 366 we were first able to remap the value to a range of zero to 1. Then by multiplying that value by 2π and taking the cosine and sine of that value for two separate columns, we now have a cyclical representation of the date.

C. WEATHER DATA COLLECTION

Unlike the AQI data where we wrote a script to build the csv file, this data was collected using the history bulk download from the openweather API. We collected the data as far back as possible for each city. The data includes features such as temperature, wind speed, visibility, and weather type.

D. WEATHER PREPROCESSING

For preprocessing the weather data, we first converted the temperature column. We converted it from Kelvin to Celsius to standardize the temperature data into a more commonly used metric, and to improve our own interpretibilty. We also removed unnecessary columns such as,'timezone', 'sea_level', 'grnd_level', and 'weather_icon'. This reduction decluttered the dataset. We then imputed missing values with zeros for the rain and snow amount columns. These values were missing since there was no rain and snow at the time of the measurement, and adding zeroes provides a consistent baseline for days without these weather events.

We also noticed that the visibility column had many missing values. To rectify this, we filled the lacking visibility data with the last known value, a method known as forward filling.

Finally, we converted the date column so that it matched the date format of the AQI data.

The 'dt_iso' column was split into 'year', 'month', 'day', and 'time' columns, and then we dropped that original column from the dataset.

Now that both datasets were properly cleaned, we merged both sets into one, so that we could use the resultant sets in our models.

The data joining process involved matching timestamps across common columns ('year', 'month', 'day', 'time') to produce a comprehensive dataset for each city. Some additional cleanup was required, including removing entries with missing date-time information. Finally, any accidental duplicate rows were removed from the dataset.

The final dataset contains weather and air quality data for the previously mentioned cities. The columns include various weather metrics such as temperature (temp), visibility, dew point, feels like temperature, pressure, humidity, wind speed, wind degree, wind gust, rainfall in the last hour, snowfall in the last hour, cloud cover percentage, weather ID, weather main category, and weather description. Additionally, there are columns indicating the year, month, day, and time of the recorded data. The dataset also includes air quality information such as state and county codes, latitude, longitude, air pollutant parameters, sample measurements, units of measurement, sample duration, detection limit, uncertainty, qualifier, method type, method, method code, state, county, AQI (Air Quality Index), and AQI category.

E. CLIMATE CHANGE DATA COLLECTION

We collected our climate change data from global-warming.org. This is a resource and host of what would otherwise be terabytes of satellite data compiled into much more digestible CSV’s and JSON’s through API’s and a pip library. Their data is sources from organizations such as EPA, NOAA, NASA, and other reliable sources. Using a script we called their API and appended the data retrieved into a CSV. We ran into issues with some of their API’s returning JSON’s, CSV’s, or XML’s. Some of the API’s needed special parsing in our script because of this.

One of the datasets that the website listed was sea level, which required the use of a pip library called cdsapi. This library handled and formatted the API calls which simplified the script written for this dataset.

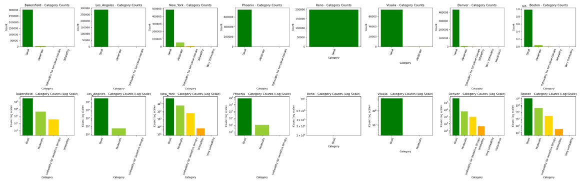
F. CLIMATE CHANGE PREPROCESSING

After collecting data on land temperatures, sea temperatures, methane levels, nitrous oxide levels and other metrics, we found that the resolution of the data was very sporadic. Some datasets had gaps weeks long and others had hour long intervals. Due to our AQI data being daily and ranging from 1980 to 2023, we moved forward only with the data that fit this time range and had at least 95% daily coverage. Of the seven datasets that we wanted to utilize, we moved forward with three, land surface temperature,sea ice extent, and sea surface temperature.

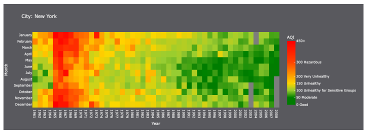
Some of the datasets had different date time formats. Keeping this universal with our AQI datasets, we needed to convert epochs, the percentage through the year, and variations of mm-dd-yyyy, to yyyy-mm-dd. We deleted data that extended beyond 2023 and before 1980. Duplicate and null rows were found and deleted. To handle the missing days, we imputed the avergae of the two nearest data points.

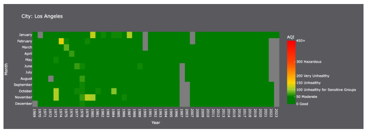
G. EXAMINING THE DATA

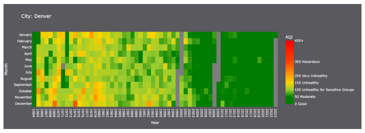
The number of rows in the AQI city data ranged anywhere from 200,000 for Reno to over 1,000,000 for Boston. Before we ran any models on the data, we examined it first with the help of visualizations. We first looked into the amount of AQI values in each category. We plotted a bar graph using the original values and then using a log scale on the y-axis. As shown in the figure, cities like Reno and Visalia have most if not all their AQI values falling into the “Good” Category, whereas cities like New York, Denver, and Boston have more diversity between the categories. This is an important context to keep in mind when analyzing the results of the models.

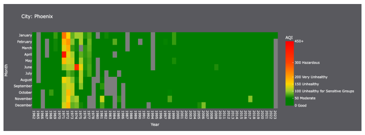


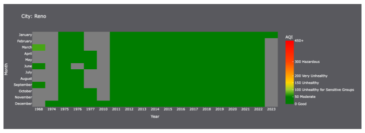
The date ranges of each city also vary greatly. Cities like Boston and New York have data going back to 1961, while Visalia only goes back until the year 1980. We also visualized the AQI values over time. As shown, the general trend is that AQI values get better over time. This is likely due to the increased efforts by the government to fight climate change and reduce greenhouse gas emissions.

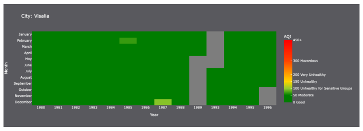


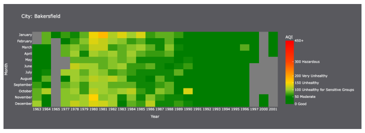


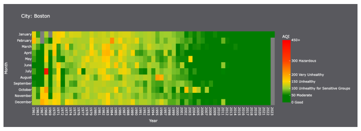




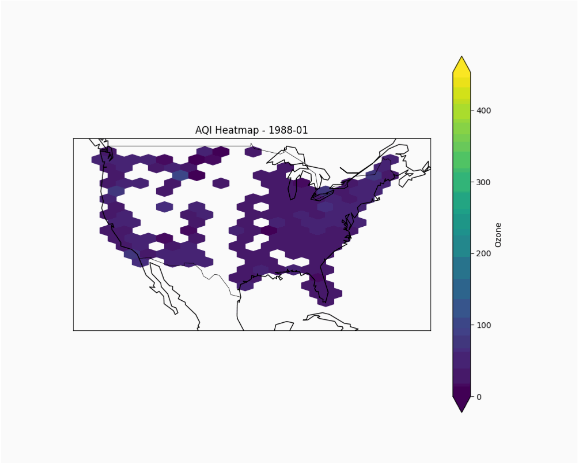




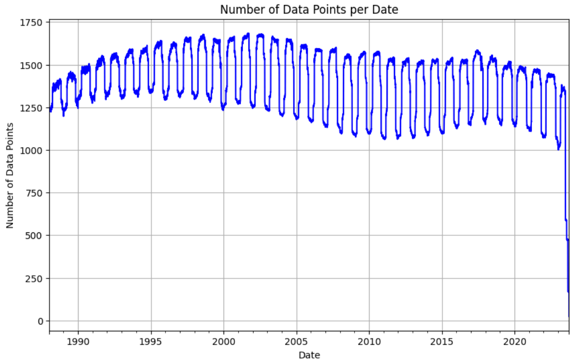




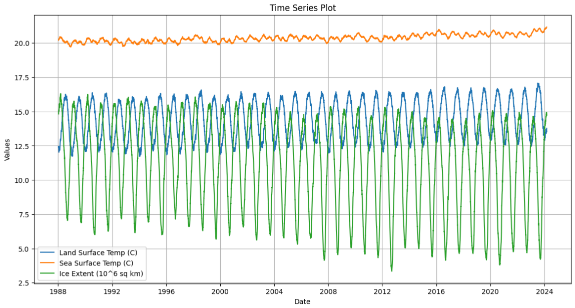
The figure below is an animated spatial heatmap of the climate change AQI dataset.



As you can see, the data is mostly occupies the CBSA areas. In this graph, we can see the frequency of AQI entries each day. It appears that there is a consistent seasonal drop in AQI entries. This will effect our models' ability to accurately predict AQI in some seasons.



The climate change data is plotted against time below. As we can see, the land surface temperature and the sea surface temperature are both slowly increasing over time. The ice extent on the other hand is decreasing. This data reflects popular concerns of global warming and it's effects on our climate. All three climate change features have a seasonal fluctuation which will need to be compromised for during the model tuning process.



IV. MACHINE LEARNING MODELS

The following variables are used in our weather models.

temp	visibility	dew_point	feels_like
pressure	'humidity'	'wind_speed'	,'wind_deg'
'wind_gust'	'rain_1h'	'snow_1h'	'clouds_all'
‘weather_id',	‘year’	‘month’	‘day’

The target variable is the ‘aqi’ column. We used five different models, including, linear regression, decision trees, random forest regressors, KNN classifier and regression. We also wanted to use support vector machines however since our dataset was over a million entries long, its time complexity of Order n^3 was unsuitable for this problem.

In our climate change models, we used the following features.

Land_Surface_Temp(C)	Sea_Ice_Extent(10^6 sq km)
Sea_Surface_Temp(C)	Longitude
Latitude	Cosine
Sine	Year

The target variable is the Ozone Level. We used univariate linear regression, multivariate linear regression, decision tree regression, KNN regression, and neural networks.

A. Multivariate Linear Regression:

In this project, linear regression models were developed to predict air quality indices (AQI) for multiple U.S. cities. Linear regression is a statistical method that models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the data. The goal is to create a linear formula that predicts the dependent variable as accurately as possible as a function of the independent variables. This equation represents a line of best fit that minimizes the difference between predicted and actual values in the dataset, providing a simple and powerful way to forecast outcomes and understand relationships between variables.

The model training involved splitting the data into training and test sets, with the latter accounting for 20% of the dataset. This 20% would be used to validate the model's performance against unseen data. Performance metrics such as mean squared error, mean absolute error, and the R-squared value were then calculated to evaluate each model's accuracy and effectiveness, and the results were documented per city. The trained models were then saved using the python library Joblib for potential future use or further analysis.

B. Decision Tree

A Decision Tree regression model was also employed to analyze and predict the Air Quality Index (AQI) using the same numerical features. A decision tree model is a supervised learning algorithm that uses a tree-like graph of decisions and their possible consequences to make predictions. It works by repeatedly splitting the training data into smaller subsets based on specific conditions or thresholds, which are determined from the features in the data. This process creates a tree structure where each node represents a decision point and the leaves represent the final outcomes or predictions. Same as the regression model, performance metrics such as mean absolute error, mean squared error, root mean squared error, R-squared, and explained variance score were used here. The same 80-20 train test split was used as well.

C. Random Forest

Another machine learning model we used was Random Forest Regression. This ensemble learning method involved training multiple decision trees on different subsets of the dataset and averaging their predictions to improve accuracy and control over-fitting. For our data, we set it so 100 different decision trees were generated. Data preparation steps included standardizing features to normalize the data scale. Again, performance metrics such as mean squared error, mean absolute error, root mean squared error, and R-squared were computed to evaluate the model's predictive reliability.

D. Neural Network

We also ran our data through a Neural Network model. Neural Networks are computational models inspired by the human brain, consisting of layers of interconnected nodes, or neurons, that process data. They learn to perform tasks by considering examples, generally without being programmed with task-specific rules. Through a process called backpropagation, they adjust their internal parameters to minimize the difference between their predicted output and the actual data, improving accuracy over time as they learn from their errors. In our code, we also used Keras Tuner to automate the optimization of hyperparameters for our model. It systematically tests a range of hyperparameter values to determine the most effective combinations that minimize the validation loss.

The hyperparameters we tested were

Number of Layers (num_layers):

Range: 1 to 5 layers

Units in Each Layer (units_i):

Range for each layer: 32 to 512 neurons, in increments of 32

Activation Function for Each Layer (activation_i):

Options: 'relu', 'tanh', 'sigmoid'

Optimizer (optimizer):

Options: 'adam', 'sgd', 'rmsprop'

Batch Size (batch_size):

Options: 32, 64, 128, 256

E. Elastic Net Regression

We also used the Elastic Net Regression model. Elastic Net Regression is a powerful linear modeling technique that combines the penalties of both Ridge regression and Lasso regression. It incorporates both L1 and L2 regularization, which helps to control overfitting by penalizing large coefficients. The blend of these regularization methods makes Elastic Net particularly useful when dealing with highly correlated data, like our weather data. Again we used the 80-20 train test split with the same features as before.

F. Gradient Boosting Machine

Gradient Boosting Machines (GBM) is a type of ensemble learning technique that builds models sequentially, with each new model focusing on correcting the errors made by the previous ones. This approach combines numerous weak predictive models, typically decision trees, into a strong predictor by optimizing a loss function, making GBMs highly effective. We ran our data through the GBM Regressor with a learning rate of 0.1 and the same features and 80-20 test split.

G. K-Nearest Neighbors

The final continuous predictive model we used is the K-Nearest Neighbors (KNN) regressor. Unlike most traditional models that attempt to learn a direct mapping from input features to output predictions, KNN works by memorizing the entire training dataset. When a prediction is needed, the KNN regressor identifies the 'k' closest instances (or neighbors) in the training data to the new instance, based on a chosen distance metric such as Euclidean distance. The final output is calculated as the average of the dependent variable values of these nearest neighbors. Again, the same 80-20 training split and features were used in the model. The same metrics were also calculated.

Along with continuous predictive models, we also utilized the K-Nearest Neighbors (KNN) classifier to predict categorical Air Quality Index (AQI) levels. This differs from the KNN regressor where it looks to the AQI category of its neighbors and chooses the majority vote, instead of taking the average neighbors' AQI values. The target variable here is the AQI_cat column we imputed in the preprocessing step. The categories are the same as those shown in the above figure. Since this is a categorical modal, its performance was assessed using different metrics from before. Instead, we used a confusion matrix, classification report, and accuracy score to evaluate the models accuracy.

V. CONCLUSION

We decided to use the MSE as our main metric for evaluating the models performance. MSE is a common statistical metric used to measure the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is particularly useful in quantitative output models like AQI prediction because it heavily penalizes larger errors, ensuring that models with lower MSE values are more precise in their forecasts. This characteristic makes MSE an excellent choice for our AQI project, as it emphasizes the accuracy of predictions, which is important for reliable air quality assessments that can impact public health and environmental policy decisions.

Of the models, the KNN Classifier performed the worst. This wasn’t very surprising as we initially suspected that a continuous model is a better fit for this problem. When dealing with AQI data where most observations fall into the "good" category, continuous models generally provide better performance than categorical models. This skew towards one category can make it challenging for a categorical model to effectively learn and predict across a a broader range of AQI levels, as it tends to overfit the dominant class and under-represents less frequent categories. Continuous regression models, on the other hand, treat AQI as a numeric scale that reflects more subtle variations in air quality, which is crucial for accurately predicting AQI values that might not necessarily align well with categorical thresholds. These models capture the inherent ordinal nature of the data, allowing for more nuanced predictions that can inform more precise environmental health assessments and policy-making decisions.

The best performing weather model was the neural network with an average MSE of 58. As for the climate change models, the best performing model was also the neural network, having a MSE of 348. The second best model was our decision tree with a MSE of 466. This is likely due to the LSTM layers that the neural network was able to take advantage of.

VI. FUTURE WORK

The results of this capstone project have provided insights into the complex interrelationships between air quality, weather conditions, and climate change. However, there are numerous opportunities for further research.

We only utilized Ozone Levels and Sulfur Dioxide to measure air quality. By using the other metrics such as carbon monoxide, nitrogen dioxide, PM2.5, and PM10, we could gather more insights on air quality trends.

Linking air quality data directly with health outcomes data would enable a more detailed analysis of the public health implications of air pollution. This could involve collaboration with healthcare providers and epidemiologists to track pollution exposure and associated health impacts over time.

By using the models and data provided from each source of understanding air quality, they could be very powerful in conjunction. Through ensemble learning, these different avenues that we can continue studying, can create more accurate models.

VII. REFERENCES

https://www.ibm.com/topics/neural-networks

https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems

https://scikit-learn.org/stable/modules/tree.html

https://www.geeksforgeeks.org/k-nearest-neighbours/

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

https://machinelearningcompass.com/machine_learning_models/elastic_net_regression/