Analyzing Air Quality: Insights into the Impact of Weather and Traffic in Big Cities

Project Methodology

Atreyo Das
das.at@northeastern.edu
Omer Seyfeddin Koc
koc.o@northeastern.edu
Shruti Suhas Kute
kute.s@northeastern.edu

Instructor: Dr. Fatema Nafa

Northeastern University, Boston MA
Master of Science in Data Science

Github Link

1 Purpose of the Methodology

The objective of our project is to analyze the relationship between air quality and external factors such as weather conditions and traffic patterns in major urban areas. To achieve this, we employ a combination of regression and time series modeling techniques, each chosen for its ability to address specific aspects of the problem.

The first task, which focuses on identifying predictors of pollutant levels (PM2.5, PM10, and NO_2), we utilize regression models to evaluate the predictive power of weather parameters (e.g., temperature, precipitation, wind speed, humidity) and traffic data. Two models—Random Forest and XGBoost—were implemented and compared. This comparison not only highlights the strengths of each model but also provides insights into the relative importance of different features in predicting air quality.

The next stage of our project involves forecasting future pollutant levels. To achieve this, we employ time series modeling techniques such as ARIMA and LSTM models. Comparing these models allow us to properly understand the nature of our time series data and help us understand which models are better suited to provide accurate results.

By employing these methodologies, we aim to provide actionable insights into the factors influencing air quality and develop reliable tools for predicting future trends, ultimately contributing to better-informed environmental policies and public health strategies.

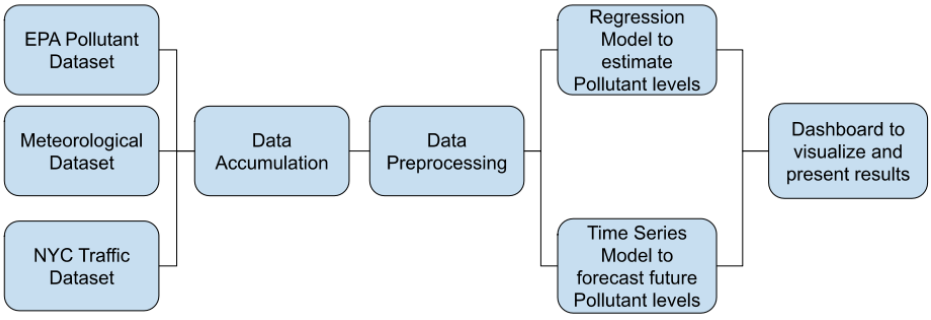

Figure 1: Project Workflow

2 Problem Statement

Urban air pollution is a pressing public health and environmental issue, particularly in densely populated cities where vehicular traffic and varying weather conditions significantly impact air quality. Pollutants such as PM2.5, PM10, and

NO2 pose severe health risks, including respiratory and cardiovascular diseases, making it crucial to understand the underlying factors driving their fluctuations.This project aims to investigate the relationship between air pollution, traffic density, and meteorological conditions in three major U.S. cities: New York, Chicago, and Los Angeles. By integrating historical air quality, weather, and traffic datasets, we seek to identify key predictors of pollution levels and develop accurate forecasting models.Through advanced machine learning techniques such as Random Forest and LSTM models, along with time series forecasting methods like ARIMA, we will assess the impact of traffic congestion and weather patterns on urban air quality. The insights gained from this research can help policymakers design targeted interventions to mitigate pollution and improve public health in metropolitan areas.

3 Data Collection and Preparation

3.1 Data Acquisition

To analyze the relationship between air quality, weather, and traffic in major metropolitan areas, we gathered data from multiple reliable sources. Each dataset provides valuable insights into different aspects of the urban environment. The data collected for this project was taken from three primary sources due to the project's multifaceted goal.

3.1.1 Air Quality

The United States Environmental Protection Agency (EPA) monitors pollutants across the country and offers comprehensive pre-generated datasets, in their official website, detailing various pollutants dating back to 1980. The website organizes the datasets into categories such as Average Yearly Summaries, Daily Levels, and Hourly Levels. They are also separated based on the different pollutants. For the purpose of the project, we took fifteen different datasets focusing on three different pollutants, PM2.5 (non FRM/FEM Mass), PM10, NO_2 collected in an hourly basis for the years 2020-2024. The datasets are collected in a compact CSV file for easier access. Each dataset has about 1-4 million columns and about 24 different columns. These columns included the county name and number, the latitude and longitude of the sites, paramater collected and it's unit, the time it was collected at, etc. This dataset would give people access to the pollutants of the site at any hour of most counties in the United States.

3.1.2 Meteorological Information

The weather parameters was collected from the website Open-Meteo, which is an open-source weather API offering free access for non-commercial use. Its historical weather API collects data from weather station, aircraft, buoy, radar, and satellite observations to create a comprehensive record of past weather conditions. The website features a user-friendly interface that allows users to select a location by name or coordinates, specify a desired time range, and choose from a variety of hourly weather variables such as temperature, air pressure, humidity, and wind speed. It then generates a well-organized dataset in CSV or XLSX format. Using we generated three weather datasets for the cities Chicago, New York and Los Angeles from the year 2020 to 2024. We selected the temperature, relative humidity, precipitation levels (both rain and ice) and wind speed (100 meters above ground) for our task. This generated a dataset with over 43,000 rows and five columns, for the four weather parameters and the time it was collected.

3.1.3 Traffic Count

Although traffic datasets for U.S. cities are widely available, finding datasets with uniformly collected hourly traffic metrics across the cities studied in this project proved challenging. As a result, we focused on analyzing the effects of traffic specifically in New York City. This data was collected from the New York State's Official Website. The dataset, MTA Bridges and Tunnels Hourly Crossings, provides hourly bridge and tunnels crossings by facility, direction, vehicle class, and payment method, such as EZ-Passes or Tolls, in a CSV, RDF, XML or JSON file. The dataset is regularly updated and contained approximately 11 million rows of data at the time of collection. Its eleven columns provide detailed information such as the date, time, the facility where the data was recorded, vehicle types, traffic counts, and other relevant details.

3.2 Data Processing

To ensure the data was suitable for analysis, multiple processing steps were undertaken. This involved combining data from different sources, imputing missing values, and transforming features to better capture trends and patterns.

3.2.1 Data Accumulation

A total of nineteen datasets were collected for the task. To streamline the analysis, the necessary information was extracted from all datasets and consolidated into a single, unified dataset. To achieve this task, we took the datasets of the pollutants and arranged them in an orderly format where each row had all the pollutant levels for a particular hour in our selected regions. To narrow the scope of our analysis, we focused on three specific counties: New York (primarily encompassing the Manhattan borough of New York City), Cook (part of the Chicago Metropolitan Area), and Los Angeles (representing the Greater Los Angeles Area). Additionally, we restricted our analysis to the month of January for each year under consideration. Upon collecting the data, we identified that the dataset provided by the EPA lacked PM10 and NO_2 concentration levels for New York County. Consequently, we limited our analysis for this county to PM2.5 levels only.

Further, the weather data from Open-Meteo was merged by matching records based on the corresponding date, hour, and county name. For New York's traffic data, we examined the names of the facilities where the data was collected and filtered out those not located within or directly connected to the county. We then aggregated the traffic counts across the relevant facilities for the corresponding time period to compute the Average Traffic Count. This aggregated metric was subsequently merged into the dataset for New York entries.

As a result, we obtained a final aggregated dataset comprising 11,160 rows and 14 columns, encompassing all necessary air pollutant, weather, and traffic data for the analysis.

3.2.2 Data Transformation

Several preprocessing techniques were applied to refine the dataset and ensure its suitability for analysis.

- **Linear Interpolation**: Minor gaps in the datasets were filled using linear interpolation to maintain continuity in the data without introducing artificial spikes.
- **Scaling**: The data was scaled using Standard Scaling to ensure features followed a consistent distribution, improving model performance.
- **Feature Engineering**: Additional features were generated to capture relevant patterns such as rolling average of the pollutant levels to highlight short-term pollutant trends while smoothing out random fluctuations and cyclic encoding of the "Day" variable into sine and cosine components to account for cyclical trends in weekly patterns.

4 Selection of Machine Learning Model

To evaluate the relationship between air pollution, weather, and traffic conditions, we implemented multiple machine learning models, including **Random Forest, XGBoost, LSTM,** and **ARIMA**. The performance of these models was assessed using key evaluation metrics such as **Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) score.**

4.1 Comparison of Model Performance

The results from the different models across New York, Chicago, and Los Angeles are summarized below:

- **Random Forest:** While Random Forest provided reasonably good predictions, it struggled with capturing the temporal dependencies in the data. This resulted in higher MAE and MSE values, particularly for pollutants with strong seasonal and hourly variations.
- **XGBoost:** XGBoost consistently outperformed Random Forest in regression-based predictions, achieving higher R^2 values and lower error rates. For instance, XGBoost achieved an R^2 score of 0.9019 for NO2 prediction in Chicago, significantly better than the Random Forest model.
- **LSTM:** The deep learning-based LSTM model demonstrated superior performance for time-series forecasting, capturing long-term dependencies in pollutant trends. In Chicago, LSTM achieved an R^2 of 0.75 for PM2.5, outperforming other models for sequential data. However, its performance varied across locations, and it required extensive tuning.

- **ARIMA:** As a traditional time-series statistical model, ARIMA performed well in short-term forecasting but struggled to capture more complex relationships between air pollution, traffic, and weather conditions. While it provided interpretable predictions, its adaptability was limited in highly dynamic urban environments.

For example, XGBoost's R^2 score for PM2.5 prediction in Los Angeles was 0.9146, significantly outperforming Random Forest. Meanwhile, LSTM showed promising results in capturing time dependencies, making it a strong candidate for sequential predictions.

4.2 Justification for Selecting XGBoost and LSTM

Based on the comparative analysis, we identified **XGBoost** as the best model for regression-based predictions and **LSTM** as the most effective model for time-series forecasting. Their selection is justified by the following factors:

4.2.1 XGBoost - Best Regression Model

- **Superior Predictive Performance:** XGBoost consistently produced higher R^2 values and lower error rates, making it the most accurate choice for structured air pollution data.
- **Robust Feature Handling:** It effectively manages missing values and outliers, which is crucial for large-scale and diverse environmental datasets.
- **Computational Efficiency:** XGBoost is optimized for speed and scalability, allowing efficient model training even with large datasets.

4.2.2 LSTM - Best Time-Series Model

- **Effective Temporal Modeling:** LSTM excelled in capturing long-term dependencies and fluctuations in air pollution levels.
- **Deep Learning Flexibility:** It adapts well to complex sequential data, making it superior for time-series forecasting compared to ARIMA and traditional machine learning models.
- **Robust Sequential Predictions:** Unlike regression models, LSTM can predict future pollutant levels based on past trends, making it more suited for dynamic air quality forecasting.

In summary, both XGBoost and LSTM emerged as the most effective models, each excelling in different aspects. XGBoost proved to be the strongest model for regression-based tasks, providing high accuracy and computational efficiency. Meanwhile, LSTM was the best choice for time-series forecasting, capturing temporal dependencies in air quality trends. While Random Forest provided a balanced alternative, it did not outperform XGBoost in predictive accuracy. Similarly, ARIMA remained useful for short-term forecasts but lacked adaptability for rapidly changing urban air quality conditions. By combining XGBoost for regression-based predictions and LSTM for time-series forecasting, we can achieve a comprehensive and robust approach to air quality modeling.

5 Model Development and Training

5.1 Architecture and Configuration

5.1.1 Random Forest Model

- **Number of Features**: The Random Forest model used a variety of features, including temporal (year, month, hour, day of the week), meteorological (temperature, humidity, precipitation, wind speed), and engineered features (rolling average of temperature, cyclic encoding of the day feature).
- **Model Complexity**: The complexity of the Random Forest model was managed by tuning hyperparameters such as the number of estimators, maximum depth, and minimum samples per split.
- **Feature Selection**: The dataset was filtered based on counties (New York, Cook, Los Angeles), and different pollutant targets (PM2.5, PM10, NO2) were selected based on availability. New York included traffic density as an additional feature.

5.1.2 XGBoost Model

- **Number of Features**: The XGBoost model utilized the same meteorological and time-based features as the other models but also incorporated engineered features such as cyclic encoding of the day and a rolling average for PM2.5 to capture short-term trends.
- **Model Complexity**: XGBoost uses gradient boosting decision trees and was optimized by tuning key hyperparameters such as the number of estimators, learning rate, tree depth, and regularization parameters.
- **Feature Selection**: Data was filtered similarly to other models, with an additional focus on using hyperparameter tuning to optimize performance.

5.1.3 ARIMA Model

- **Number of Features**: The ARIMA model focused on pollutant data (PM2.5, PM10, NO2) recorded across selected counties (Cook, Los Angeles, and New York). Data was organized chronologically with features such as date, month, and hour to ensure temporal consistency.
- **Model Complexity**: The ARIMA model utilized a grid search approach to identify the best combination of parameters.
- **Feature Selection**: The dataset was filtered based on the pollutants (PM2.5, PM10, NO_2) levels on each year in a particular city (New York, Chicago, Los Angeles).

5.1.4 LSTM Model

- **Number of Features**: The LSTM model used time series-based features, including temperature, humidity, precipitation, wind speed, and traffic density (for New York).
- **Model Complexity**: The model consisted of two LSTM layers (with 50 units each), a dense layer of 25 neurons, and a final output layer.
- **Feature Selection**: Data was preprocessed by normalizing relevant features using MinMaxScaler, and sequences of 24-hour intervals were created for time series forecasting.

5.2 Training Process

5.2.1 Data Splitting

- All the models split the dataset into **80% training and 20% testing**.
- ARIMA and LSTM used sequential data splitting to maintain temporal integrity.

5.2.2 Overfitting Handling

- **Random Forest**: Used cross-validation during hyperparameter tuning.
- **XGBoost**: Employed L1 (Lasso) and L2 (Ridge) regularization to prevent overfitting, alongside cross-validation for hyperparameter selection.
- **ARIMA**: Monitored model performance on a separate test set to prevent overfitting on the training data, along avoiding excessive hyperparameter ranges to limit model complexity.
- **LSTM**: Utilized a dropout layer within the LSTM network and early stopping based on validation loss.

5.3 Hyperparameter Tuning

5.3.1 Random Forest

- Utilized **GridSearchCV** with cross-validation to optimize hyperparameters, tuning:
  - Number of trees (`n_estimators`)
  - Tree depth (`max_depth`)

- Minimum samples per split (`min_samples_split`)
- Minimum samples per leaf (`min_samples_leaf`)
- Bootstrapping strategy (`bootstrap`)

5.3.2 XGBoost

- Performed **GridSearchCV** with cross-validation to tune hyperparameters, optimizing:
  - Number of estimators (`n_estimators`)
  - Learning rate (`learning_rate`)
  - Maximum tree depth (`max_depth`)
  - Subsample ratio (`subsample`)
  - Column sampling for features (`colsample_bytree`)
  - L1 regularization (`reg_alpha`)
  - L2 regularization (`reg_lambda`)

5.3.3 ARIMA

- Utilized a custom grid search strategy to optimize hyperparameters, tuning the order parameters:
  - p (autoregressive terms) — controls lagged values in the model.
  - d (differencing order) — ensures stationarity by differencing the data.
  - q (moving average terms) — controls lagged forecast errors

5.3.4 LSTM

- Used manual tuning for:
  - Number of LSTM units (50 per layer)
  - Number of layers (2 LSTM layers)
  - Batch size (32)
  - Number of epochs (20)
  - Learning rate (default Adam optimizer settings)

6 Evaluation and Comparison

When evaluating different air quality prediction models, we compared four approaches: Random Forest, XGBoost, LSTM, and ARIMA. Each model has its strengths and challenges, with certain approaches performing better in specific scenarios. Random Forest proved to be a strong model when working with structured data. In Chicago's NO2 prediction, it achieved an R^2 score of 0.787, demonstrating solid accuracy. However, in a more variable environment like Los Angeles, PM10 prediction yielded an R^2 of just 0.357, indicating that the model struggled to capture complex time-series relationships.

On the other hand, XGBoost stood out as the best-performing model overall. It reached an impressive R^2 of 0.9019 for NO2 prediction in Chicago, while also performing well for PM10 with an R^2 of 0.8557. However, for PM2.5 in Los Angeles, it scored 0.6084, suggesting that while it is highly accurate, it may be sensitive to city-specific variations. The LSTM model, a deep learning-based time-series approach, performed well in capturing long-term dependencies but exhibited more variation in accuracy. In Chicago, it achieved an R^2 of 0.75 for PM2.5, showing strong predictive power. However, in Los Angeles, its PM10 prediction had an R^2 of only 0.19, highlighting the model's sensitivity to data availability and hyperparameter tuning.

Lastly, the ARIMA model, a traditional statistical time-series approach, showed consistency in short-term predictions. For PM2.5 in Chicago, the best ARIMA model, with the order parameters (1,2,1), resulted in a test RMSE of 4.71, but for PM10, the RMSE increased to 9.71, indicating higher prediction errors compared to other models. While ARIMA worked well with relatively stable data, its flexibility was limited when dealing with more complex and dynamic air quality patterns.

Overall, both XGBoost and LSTM emerged as the best models, each excelling in different aspects. XGBoost proved to be the most accurate model for regression tasks, offering high predictive power and robustness across different pollutants and cities. Meanwhile, LSTM was the strongest model for time-series forecasting, effectively capturing temporal dependencies and long-term trends.