Assignment 2 Report

Luke Emmins – 24021124

Air Quality Data in India (2015-2020)

Contents

[Introduction 3](#_Toc181484806)

[Literature Review 5](#_Toc181484807)

[Air Pollution Prediction with Machine Learning: A Case Study of Indian Cities 5](#_Toc181484808)

[Air Quality Prediction by Machine Learning Models: A Predictive Study on the Indian Coastal City of Visakhapatnam 7](#_Toc181484809)

[AQI Prediction Based on CEEMDAN-ARMA-LSTM 8](#_Toc181484810)

[Methodology 9](#_Toc181484811)

[Data Collection 9](#_Toc181484812)

[Data Preprocessing 10](#_Toc181484813)

[Missing Values 10](#_Toc181484814)

[Outliers 12](#_Toc181484815)

[Feature Engineering 13](#_Toc181484816)

[Data Transformations 13](#_Toc181484817)

[Feature Selection 14](#_Toc181484818)

[Ridge Regression 14](#_Toc181484819)

[Lasso Regression 15](#_Toc181484820)

[Principal Component Analysis 16](#_Toc181484821)

[Feature Removal 17](#_Toc181484822)

[Model Selection 17](#_Toc181484823)

[Ordinary Least Squares 17](#_Toc181484824)

[Random Forest Regressor 18](#_Toc181484825)

[Support Vector Regression 18](#_Toc181484826)

[XGBoost 18](#_Toc181484827)

[Neural Network + XGBoost 18](#_Toc181484828)

[Model Training 18](#_Toc181484829)

[Dataset Partitioning 18](#_Toc181484830)

[Parameter Tuning 19](#_Toc181484831)

[Cross Validation 19](#_Toc181484832)

[Model Evaluation 19](#_Toc181484833)

[Metrics 19](#_Toc181484834)

[Implementation & Assumptions 20](#_Toc181484835)

[Software & Packages 20](#_Toc181484836)

[Data Imbalance 20](#_Toc181484837)

[Results 22](#_Toc181484838)

[Model Performances 22](#_Toc181484839)

[SMOTER 23](#_Toc181484840)

[Optimal Model Parameters 24](#_Toc181484841)

[Model Comparisons 26](#_Toc181484842)

[Additional Tuning 27](#_Toc181484843)

[Discussion 28](#_Toc181484844)

[Evaluation 28](#_Toc181484845)

[Comparison to other Work 28](#_Toc181484846)

[Limitation 28](#_Toc181484847)

[Impact 28](#_Toc181484848)

[References 29](#_Toc181484849)

# Introduction

The Air Quality Index, referred to as AQI, indicates how polluted the air is at a given time. Government agencies will take measurements of chemicals and categorise the air based on their concentrations, among other factors. Common pollutants measured include carbon monoxide, nitrogen dioxide, ozone, and sulfur dioxide, as well as particles known as PM 10 and PM 2.5 (Wikipedia). The PM particles are known as Airborne Particulate Matter and comprise a mixture of chemical species.

The AQI scale used is not universal and varies between countries. This is common practice as countries develop the standards they wish to compare to, as seen in the Water Quality Index (WQI), Consumer Price Index (CPI) and several other sectors. Figures 1, 2, and 3 highlight the differences in the approach to Air Quality between Australia, India, and Singapore.

A screenshot of a medical advice

Description automatically generated

Figure 1. Air Quality Index (Australia)

A screenshot of a medical report

Description automatically generated

Figure 2. Air Quality Index (India)

A screenshot of a computer

Description automatically generated

Figure 3. Pollutant Standards Index (Singapore)

Each of the countries approach the problem differently. Australia and India use similar categories with different indexes, whereas Singapore uses a different scale entirely called the Pollutant Standards Index, which also tracks the pollutants.

In this report, I will look closely at the air quality in select regions in India between 2015 and 2020 and begin to explore how regional factors, specifically meteorological, can affect the prediction of AQI using various models, specifically looking at industrial and coastal areas. The dataset provided by Kaggle provides daily data across various cities, highlighted in figure 4, in India for a range of pollutants. The table below shows the list of pollutants included in the Kaggle dataset, along with their corresponding measurements.

|  |  |  |
| --- | --- | --- |
| Pollutant | Formulae | Measurement |
| Particulate Matter 2.5 | PM2.5 |  |
| Particulate Matter 10 | PM10 |  |
| Nitric Oxide | NO |  |
| Nitrogen Dioxide | NO2 |  |
| Nitric Oxide | NOx |  |
| Ammonia | NH3 |  |
| Carbon Monoxide | CO |  |
| Sulphur Dioxide | SO2 |  |
| Ozone | O3 |  |
| Benzene | N/A |  |
| Toluene | N/A |  |
| Xylene | N/A |  |

A map of the world with red points

Description automatically generated

Figure 4. Locations of Indian Cities in Kaggle Dataset

# Literature Review

With an ever-growing volume of data on air pollutants, the exploration of AQI forecasting has become more prominent. Accurate AQI predictions are vital for mitigating health risks associated with pollution, especially in highly populated areas. Traditional models and regression techniques have been used to predict pollutant levels and subsequent AQI levels, but limitations in handling larger, complex datasets have shifted focus towards Machine Learning and hybrid models.

## Air Pollution Prediction with Machine Learning: A Case Study of Indian Cities

India is a prime example of a rapidly growing country with increasing levels of pollutants in the air, with over 2 million deaths per year due to fossil fuels (BMJ, 2020). The dataset examined in the study by Kumar & Pande et al. (2023), is identical to the one that was analysed by myself. This dataset, containing key pollutants such as PM2.5, NO2, CO, and SO2, allows for a direct comparison of methodologies and results, further validating the findings and enabling deeper insights into AQI prediction trends.

They looked at a variety of Machine Learning models, including XGBoost, Gaussian Naïve Bayes, Support Vector Machines (SVM), Random Forest, and K-Nearest Neighbour, to predict air quality.

Due to the multi-dimensional dataset, Kumar & Pande discussed the need for feature selection that best represents that data. They opted to use a correlation-based approach for identifying key variables that directly impacted the AQI and found that pollutants such as PM2.5 and NO2 were among the key contributors to AQI values. Whilst this approach yields promising results, feature selection using only highly correlated variables has its limitations. A correlation-based approach ignores non-linear relationships, does not account for feature interactions, is sensitive to outliers, and can result in feature redundancy. Additionally, it may overlook the issue of multicollinearity, where two or more predictor variables are highly correlated with each other. Multicollinearity can inflate the variance of the estimated coefficients in predictive models, leading to less reliable estimates and reducing the model’s overall interpretability. Kumar & Pande do not explicitly address these issues, as their focus remains on identifying the most impactful features without delving into potential correlations between the selected features themselves or how multicollinearity might affect the model's stability and predictive power.

One area they discussed was the seasonality of the pollutants. Using EDA, they examined how the pollutants acted across the year, as this was something that an effective model would need to implement. They noticed a data imbalance that arose because of the uneven distribution of air pollution levels across the dataset; this was due to the rarer occurrences of high pollution levels. To address this, they applied SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic data points for the minority classes. Without the addition of SMOTE, the models can become biased towards underpredicting extreme pollution events.

The results of their study examined the models with and without SMOTE applied, and it showed that the XGBoost model outperformed other models, exhibiting the highest degree of linearity between the predicted and actual AQI values. On the other hand, the Gaussian Naïve Bayes model also showed relatively high accuracy, whereas the SVM model yielded the lowest accuracy among all tested models. The metrics that were evaluated for each included Mean Absolute Error, Root Mean Squared Error, Root Mean Squared Logarithmic Error, and R-squared, with XGBoost providing the optimal values across both the training and testing sets.

To improve their analysis, implementation of hybrid models such as a Neural Network combined with XGBoost or ARIMA could help capture the underlying patterns singular models cannot capture.

## Air Quality Prediction by Machine Learning Models: A Predictive Study on the Indian Coastal City of Visakhapatnam

Pollutants in the air can be released by various sources, both by humans and natural processes. Industrial areas are the leading contributors to pollutant emissions, with household emissions contributing heavily and over 2 billion people being exposed to dangerous levels of household air pollution regularly (WHO). Pollutant levels and types can differ significantly across regions because of many factors, such as population density, industrial activity and local meteorological conditions. Simple Machine Learning models will struggle to implement regional variability into predictions as they typically assume uniform patterns unless the data suggests otherwise. Ravindiran et al. (2023) sought to address the gap in the topic by focusing on the coastal city of Visakhapatnam, where local meteorological conditions and industrial emissions interact differently compared to more inland cities.

Like the previous study, Ravindiran applied five different Machine Learning models to evaluate their performance on AQI predictions. He looked at the same pollutants as the previous study but filtered out all data corresponding to Visakhapatnam and implemented additional meteorological data such as temperature, wind speed, wind direction, relative humidity, solar radiation, and rainfall. The aim was to investigate how external factors affected pollution levels and if they had any significance in predicting AQI. The models used for testing were LightGBM (Light Gradient Boosting Machine), Random Forest, CatBoost, AdaBoost and XGBoost.

In his study, Ravindiran does not explicitly discuss the feature selection process but discusses the importance of certain pollutants using EDA. He found that certain pollutants, such as PM2.5, PM10, NO2 and CO, had the strongest correlation with AQI and suggested that these were the best features to use for the model. The EDA also explored the effects that the meteorological features had on the AQI and was found that their contribution to AQI was minimal outside of Barometric Pressure.

The study found that the CatBoost model was the best-performing, with an value of 0.9998 and the lowest RMSE of 0.76, predicting AQI with high accuracy. Both the XGBoost and LightGBM also performed well but were not as accurate, and the AdaBoost was the weakest-performing model. It was found that even though the models were weaker in comparison, they all yielded values greater than 0.97 on both the training and testing data sets. Ravindiran concluded that the ensemble learning models offered highly accurate AQI predictions for specific regions and highlighted the effectiveness of implementing certain external factors such as meteorological.

Whilst the study yielded interesting results, I believe that it lacked the exploration of the meteorological factors mentioned in the introduction. Ravindiran focused solely on one region of India but neglected to explore the influence of geographical variations that might affect AQI, such as proximity to the sea or the seasonality of the meteorological features. Implementation of hybrid model too would have allowed for a more beneficial analysis of these factors as it is an effective model to explore how weather patterns on the coast can affect pollution. Instead, the meteorological factors were dismissed after performing EDA.

## AQI Prediction Based on CEEMDAN-ARMA-LSTM

The use of hybrid models combines the strength of different models, enabling them to handle various complexities such as non-linear relationships, time-dependent data, and the issues of overfitting.

After reviewing several studies on the topic of AQI prediction, it was found that Machine Learning models work better when the number of external factors is reduced, specifically location. Ravindiran and Sun & Liu both yielded promising results in comparison to Kumar & Pande, and I believe that it is solely due to the area of pollutants measured and analysed being regional rather than countrywide. Sun and Liu et al. (2022) sought to implement a hybrid model consisting of a Complete Ensemble Empirical Mode Decomposition (CEEMDAN), Autoregressive Moving Average (ARMA), and Long Short-Term Memory (LSTM) to predict AQI in the Beijing-Tianjin-Hebei region using monthly pollution data.

CEEMDAN is a single decomposition technique that breaks down complex non-linear relationships by representing the high and low-frequency variations in the data. ARMA is a model used for time series forecasting by combining the autoregressive part (AR) and the moving average (MA) part, modelling the relationship between current and past values along with the current and past forecast errors, respectively. LSTM is a neural network that is designed to capture long-term dependencies and can learn from past data to make predictions. By combining these models, Sun & Liu were able to capture simple, linear patterns as well as complex, non-linear patterns.

The authors compared various combinations of this model, including CEEMDAN-LSTM, negating the ARMA model, LSTM as a standalone neural network, and ARMA-GARCH. Their investigations found that the CEEMDAN-ARMA-LSTM outperformed all other models in several performance metrics, such as MAE, MSE, RMSE, and MAPE. The margins in which the CEEMDAN-ARMA-LSTM outperforms other models are noticeable, with the authors claiming a relatively high accuracy and good prospects of application. It is also worth noting that the authors highlight the benefits of the other models by themselves, highlighting their strengths and weaknesses and reinforcing the motivation for combining various models to predict AQI.

The work conducted by Sun & Liu provides useful insights into AQI prediction in the Beijing-Tianjin-Hebei region. This insight is region-specific, and the lack of application to any other locations will always prompt the question of overfitting. Without the models being tested on different areas with varying pollution levels, there will be doubt about their effectiveness and scalability of them. Furthermore, with the increasing complexity of the models used, such as CEEMDAN-ARMA-LSTM, it creates the issue of interpretability.

Expanding the study to include other regions of China, such as Shanghai, would provide a valuable opportunity to test the model’s capabilities and address the risk of overfitting. Furthermore, implementing interpretability techniques such as SHAP (Shapley Additive Explanations) or LIME (Local Interpretable Model-Agnostic Explanations) into the models would offer significant value as these methods help speak to the model’s predictions by revealing the impacts of certain input features, contributing to the real-world application of these models.

# Methodology

During my investigation, I looked to implement a SEMMA methodology approach. This approach focuses more on the data preparation and modelling of the data in comparison to similar models such as CRISP-DM. The phases of SEMMA include sampling the data, exploration, modification, modelling, and then assessment of the results. This approach is highly beneficial to problems where data exploration is key. Figure 5 highlights a flow chart for a typical implementation of a SEMMA methodology.

A diagram of a model

Description automatically generated

Figure 5. SEMMA Model

## Data Collection

The dataset used in the modelling step of this study composed of 4 datasets. The datasets were both sourced from Kaggle, Air Quality Data in India (2015-2020) (2020). <https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india>, and Indian Cities weather 2010-2024: Dive in! (2024). <https://www.kaggle.com/datasets/mukeshdevrath007/indian-5000-cities-weather-data>. The data provided by both authors comprised of CSV files containing data on the weather of Indian cities as well as the pollution data mentioned previously.

## Data Preprocessing

To mitigate the variance in the dataset caused by geographical variation, I reduced the original dataset to 3 cities in India: Amritsar, Amaravati, and Jaipur. The cities were chosen so that the meteorological features and pollutant data would vary enough to notice the impacts of each figure whilst keeping the dataset manageable and reducing regional inconsistencies.

Amritsar represents a northern city with varying climate conditions due to its proximity to the Himalayas. Amaravati, located in southern India, is characterised by a warmer and more humid climate, and Jaipur, situated in the western part of India, experiences dryer conditions comparatively. These three cities will also experience differing pollutant levels due to differences in industrialisation, populations and one being a coastal city.

### Missing Values

The raw dataset on air pollution contained large amounts of missing data. I dropped all rows not containing values for AQI as these could not be used in the testing of a model, along with the column labelled ‘AQI\_Bucket’ as this was used for AQI classification. To find any underlying patterns in the data, I found the number of missing data per feature, highlighted in Figure 6.

|  |  |
| --- | --- |
| Feature | Missing Data |
| PM2.5 | 90 |
| PM10 | 30 |
| NO | 55 |
| NO2 | 12 |
| NOx | 334 |
| NH3 | 16 |
| CO | 82 |
| SO2 | 140 |
| O3 | 85 |
| Benzene | 189 |
| Toluene | 206 |
| Xylene | 1444 |

Figure 6. Table for Missing Values

Elevated numbers for Benzene, Toluene, and Xylene prompted further investigation, revealing that Benzene and Xylene did not strongly contribute to the prediction of AQI and were subsequently dropped from the data, correlation shown by the values in the correlation matrix in Figure 8. Toluene, however, provided a value of 0.430127, signifying a higher correlation with AQI. The feature was subsequently removed due to the number of points that needed to be synthesised.

A graph of different colored lines

Description automatically generated

Figure 7. Benzene, Toluene, Xylene, AQI (Normalised) over time

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AQI | Benzene | Toluene | Xylene |
| AQI | 1.000000 | 0.161793 | 0.430127 | 0.220132 |
| Benzene | 0.161793 | 1.000000 | 0.332796 | 0.487772 |
| Toluene | 0.430127 | 0.332796 | 1.000000 | 0.350304 |
| Xylene | 0.220132 | 0.487772 | 0.350304 | 1.000000 |

Figure 8. Correlation Matrix

Both SO2 and NOx had large amounts of data missing. Figures 9, 10 and 11 highlight how the missing data appeared for each feature over time for each city in the dataset. What is interesting here is that most NOx points missing are in 2017 and have now been resolved, whilst missing data for SO2 is focused in one city. As a result, both features remained in the dataset.

A purple and yellow graph

Description automatically generated

Figure 9. Missing Data for Jaipur

To complete the dataset, I proceeded to use a model called Iterative Imputer to synthesise the missing points. It works by modelling each missing value as a function of the other features in the dataset, allowing it to make predictions based on available data. This is a beneficial approach for this problem due to the models’ efficiencies in working with correlated data. We can make impactful predictions on the missing points without employing more basic, error-prone methods.

A purple and yellow graph

Description automatically generated

Figure 10. Missing Data for Amaravati

A purple and yellow graph

Description automatically generated

Figure 11. Missing Data for Amritsar

### Outliers

Addressing the issue of outliers for AQI predictions was a complicated task. Unless the captured values for a specific pollutant were recorded by faulty equipment, disregarding them undermines the problem being addressed, predicting AQI.

A graph with a graph and numbers

Description automatically generated with medium confidence

Figure 12. Box Plot for outliers

Figure 12 revealed that all features included large amounts of outliers, reinforced by Figure 13. Due to the higher volumes of outliers, I could not emit them from the dataset due to their importance and instead explored resampling techniques, discussed in a later section. The priority was to represent all cases for AQI to ensure our model provided interpretability and could be applied to the problem.

|  |  |
| --- | --- |
| Feature | Number of Outliers |
| PM2.5 | 107 |
| PM10 | 84 |
| NO | 181 |
| NO2 | 124 |
| NOx | 136 |
| SO2 | 120 |
| CO | 120 |
| O3 | 78 |
| NH3 | 161 |

Figure 13. Number of outliers for all features

### Feature Engineering

My investigation included the implementation of meteorological conditions that can affect AQI, specifically the weather. The dataset, Indian Cities Weather, provided a range of information on different types of hourly weather. I engineered some features that could best represent the weather across each day, including averages and maximum values, including all possible features. The features I engineered can be found in Figure 14.

|  |
| --- |
| Temperature\_mean |
| Temperature\_max |
| Humidity\_mean |
| Humidity\_max |
| Dew\_point\_mean |
| Dew\_point\_max |
| Precipitation\_sum |
| Rain\_sum |
| Cloud\_cover\_sum |
| Wind\_speed\_10m\_mean |
| Wind\_speed\_10m\_max |
| Wind\_speed\_100m\_mean |
| Wind\_speed\_100m\_max |

Figure 14. Features Engineered

### Data Transformations

I found that PM2.5 and PM10 particles correlated heavily with AQI, so much so that they were leading features implemented in the papers I reviewed. However, using both pollutants in the modelling often led to multicollinearity and overfitting, yielding promising results on the test set but poor results when cross-validating. Due to the nature of both particles, PM2.5 particles comprise a portion of PM10 particles (arb.ca.gov).

To address this issue, I applied Principal Component Analysis (PCA) on the two features to combine them into one new feature called ‘PC1’ in the Jupyter Notebook. The aim of this is to create a new feature that can represent as much variance between the two features as possible, minimising multicollinearity and maintaining as much of the interpretability as possible. After applying PCA, the variance explained by the new component was 85.24%.

Standardisation was the only other transformation used as it was a preprocessing requirement for several models used within my research, including the SVR model and PCA.

## Feature Selection

To best fit the models, the dimension of the dataset needed to be reduced to ensure the models were not overfitted within the training stage. I opted to use both Ridge and Lasso Regression to analyse the effects of different algorithms on the original features to determine the weightings of each and if certain features needed to be removed.

### Ridge Regression

I implemented Ridge Regression to determine the effect of each feature on the AQI. Ridge regression will use the input features to model the data and produce coefficients for each feature, denoting the weighting of the features. An optimisation algorithm was used to find optimal parameters for the model to yield the best results. Figure 15 shows the weights for each feature, highlighting the correlation each feature has with AQI, and Figure 16 shows the performance metrics calculated for the model.

|  |  |
| --- | --- |
| Feature | Coefficient |
| PM2.5 | 19.4429 |
| PM10 | 40.3056 |
| NO | 2.6178 |
| NO2 | 3.9969 |
| NOx | -0.8001 |
| NH3 | -1.8427 |
| CO | 3.4398 |
| SO2 | 0.0591 |
| O3 | 6.3830 |
| Temperature\_mean | -0.1537 |
| Temperature\_max | -0.2777 |
| Humidity\_mean | 1.5425 |
| Humidity\_max | 1.7028 |
| Dew\_point\_mean | 1.5615 |
| Dew\_point\_max | -2.2894 |
| Precipitation\_sum | -0.1532 |
| Rain\_sum | -0.1532 |
| Cloud\_cover\_sum | 0.1124 |
| Wind\_speed\_10m\_mean | 1.1986 |
| Wind\_speed\_10m\_max | -1.1583 |
| Wind\_speed\_100m\_mean | 0.6510 |
| Wind\_speed\_100m\_max | 0.9381 |

Figure 15. Ridge Coefficients

|  |  |
| --- | --- |
| Metric | Value |
| Mean Square Error | 705.0775 |
| R-Squared | 0.8129 |

Figure 16. Ridge Performance Metrics

### Lasso Regression

Lasso Regression however, will penalise features that do not contribute to the prediction of the target variable. After the results of Ridge Regression yielded very low coefficients for some features, Lasso Regression was an algorithm that would further analyse their effects on AQI and reduce them to 0 if needed.

The approach for applying Lasso Regression followed the same steps as Ridge Regression, with a GridSearch being used to find the optimal alpha value. The intention behind this is to find a finer-tuned model with selected features being omitted by the Lasso Regression Model. Figure 17 highlights the features removed by the algorithm, represented with a 0.

|  |  |
| --- | --- |
| Feature | Coefficient |
| PM2.5 | 18.2334 |
| PM10 | 42.3753 |
| NO | 0.4779 |
| NO2 | 2.4616 |
| NOx | 0.0000 |
| NH3 | -0.0000 |
| CO | 2.6974 |
| SO2 | 0.0000 |
| O3 | 5.0142 |
| Temperature\_mean | -0.0000 |
| Temperature\_max | -1.1196 |
| Humidity\_mean | 0.0000 |
| Humidity\_max | 1.2598 |
| Dew\_point\_mean | -0.0000 |
| Dew\_point\_max | -0.0000 |
| Precipitation\_sum | 0.0000 |
| Rain\_sum | 0.0000 |
| Cloud\_cover\_sum | 0.0000 |
| Wind\_speed\_10m\_mean | 0.0000 |
| Wind\_speed\_10m\_max | 0.0000 |
| Wind\_speed\_100m\_mean | 0.2459 |
| Wind\_speed\_100m\_max | 0.0000 |

Figure 17. Lasso Coefficients

Most of the features that were engineered did not have much predictive power and were penalised in the Lasso Regression and were subsequently removed from the dataset. Figure 18 highlights that the negation of these features made the linear regression model worse but only by a negligible amount, as the value remains unchanged to 4 decimal places.

|  |  |
| --- | --- |
| Metric | Value |
| Mean Square Error | 705.1791 |
| R-Squared | 0.8129 |

Figure 18. Lasso Performance Metrics

### Principal Component Analysis

The coefficients tables for both the Ridge and Lasso Regression algorithms highlighted the importance of both PM2.5 and PM10 particles when predicting AQI. By implementing PCA, we saw an increase in the magnitude of coefficients in both the Ridge and Lasso Regressions, as shown in Figure 19.

|  |  |  |
| --- | --- | --- |
| Feature | Coefficient | |
| Ridge | Lasso |
| PC1 | 55.5651 | 55.7898 |
| NO | 3.4023 | 2.6533 |
| NO2 | 4.6578 | 3.8804 |
| NOx | -1.2953 | -0.0000 |
| NH3 | -2.5154 | -2.3886 |
| CO | 4.5225 | 4.1384 |
| SO2 | -1.3136 | -1.1143 |
| O3 | 7.1369 | 7.0569 |
| Temperature\_mean | -3.1066 | -0.0000 |
| Temperature\_max | 0.0819 | -0.4843 |
| Humidity\_mean | -0.2246 | 1.7635 |
| Humidity\_max | 1.4029 | 1.3564 |
| Dew\_point\_mean | 6.1278 | 0.4732 |
| Dew\_point\_max | -2.8415 | -0.0000 |
| Precipitation\_sum | 0.0417 | -0.0000 |
| Rain\_sum | 0.0417 | -0.0000 |
| Cloud\_cover\_sum | -0.9777 | -0.7294 |
| Wind\_speed\_10m\_mean | 2.4465 | 1.2949 |
| Wind\_speed\_10m\_max | -1.4796 | -0.2129 |
| Wind\_speed\_100m\_mean | -0.8209 | 0.1027 |
| Wind\_speed\_100m\_max | 2.3003 | 1.1195 |

Figure 19. Ridge & Lasso Regression Coefficients after PCA

By reducing the dimensions of the PM particles, the models could explore other features to determine their effects on AQI, which is why we see fewer features penalised in the Lasso Regression. As a result, both models saw performance increases, as shown in Figure 20.

|  |  |  |
| --- | --- | --- |
| Metrics | Value | |
| Ridge Regression | Lasso Regression |
| Mean Square Error | 624.0663 | 620.4562 |
| R-Squared | 0.8344 | 0.8354 |

Figure 20. Ridge & Lasso Regression Performance Metrics after PCA

### Feature Removal

Features were removed that either had no correlation or very little to AQI. It was important to include those that had coefficients large enough to have an impact on predicting AQI, both positive and negative, something that was disregarded in the literature reviews. The features that were removed from the data included: NOx, temperature\_mean, dew\_point\_max, precipitation\_sum, rain\_sum, wind\_speed\_100m\_mean, wind\_speed\_10m\_max, temperature\_max, dew\_point\_mean.

The final set of features encapsulated a range of air pollutants from the dataset whilst also including a smaller section of the engineered features related to the weather.

## Model Selection

### Ordinary Least Squares

The Ordinary Least Squares (OLS) model estimates the relationship between independent and dependent variables by minimising the sum of squared differences between observed and predicted values. Its interpretability and simplicity make it a good benchmark model.

OLS assumes that the relationships between the predictors and the target variables are linear, and their residuals are normally distributed. This model provides a good starting point by predicting AQI using linear models and comparing them to more complex models down the line to better determine the relationships between predictors and target variables.

### Random Forest Regressor

The Random Forest Regressor is an ensemble method that constructs multiple decision trees during the training phase and outputs the mean prediction of individual trees. It works by bootstrapping samples from the data and using a subset of features to reduce the chances of overfitting and improve generalisation.

Random Forest models can capture non-linear relationships and are a popular choice of model for regression problems.

### Support Vector Regression

Support Vector Regression (SVR) uses the principles of Support Vector Machines and applies them to regression problems. SVR tries to fit the data within a specified margin, minimising the error for data points within this margin and penalising those outside of this margin.

SVR models provide more flexibility in the customisation, with the selection of different kernels: Linear, Polynomial, RBF.

### XGBoost

XGBoost is a more advanced ensemble algorithm that uses gradient-boosting techniques. It iteratively improves its predictions by fitting new models to the residuals of previous predictions, focusing on reducing the error. XGBoost is known for its speed and scalability, making it ideal for the prediction of AQI. It also includes regularisation to prevent overfitting, which is useful for noisy data – another useful implementation for the dataset.

### Neural Network + XGBoost

The implementation of a Neural Network (NN) alongside XGBoost allows for pattern recognition to be included in the model. Both models have been implemented in a stacked nature in such a way that the input of the NN is the output of the XGBoost model.

The NN can better capture potential seasonality or better predict extreme values of AQI, thus making it an optimal machine learning model. By leveraging the strengths of two ML models, I hope to provide as accurate results as possible.

## Model Training

### Dataset Partitioning

The data used for modelling was partitioned using an 80:20 training testing split. This allowed for enough training data to identify underlying patterns and relationships.

### Parameter Tuning

Each model used takes different input parameters. Choosing the optimal set for each model was done by looping through a list of different parameters to find the best-performing set. Baseline versions of each model use a random set of parameters, with little motivation for their reasoning, due to the implementation of a Random Search algorithm to find optimal parameters for each model.

For both the Ridge & Lasso Regression models, Grid Search was used to find optimal parameters as it provided a cost-effective way to find the best alpha value. However, as the complexity and number of parameters for the models increased, the computational cost of using Grid Search increased. As a result, the random search algorithm was implemented to find optimal parameters whilst balancing computational cost.

### Cross Validation

During the testing phase, it was important to check for overfitting. Whilst the models produced performance metrics such as r-squared or MSE, these values only applied to the data it was trained on. To better test a model’s performance, -fold validation can be applied. During this process, the same performance metrics can be calculated across each fold, and an average can be taken for the overall performance of the model. If certain models perform worse across certain folds, it suggests overfitting, something standard metrics may not show.

Performing -fold validation can also be computationally expensive with the more complex models. The solution was to lower the number of folds, but the computational cost was disregarded to ensure consistency across all tests.

## Model Evaluation

### Metrics

Mean Square Error (MSE), Root Mean Square Error (RMSE) and Coefficient of Determination () were calculated to evaluate the performance of the models. MSE is the average of the squared differences between the predicted values and actual values. The RMSE is the square root of MSE and is more useful for model comparison as it is in the same units as AQI, providing a more interpretable result. indicates the proportion of variance in AQI that is predictable for the input features, with a value between 0 and 1 providing another interpretable metric for evaluating a model’s performance.

The last method of evaluation is the use of the residuals. The residuals are the differences between predicted values and actual values. Plotting these residuals on a scatter plot allows for a visual representation of where each model struggles and where it excels. Figure 21 provides an example.

A graph with blue dots

Description automatically generated

Figure 21. Residual Scatter Plot

## Implementation & Assumptions

### Software & Packages

All the work was completed in a Jupyter Notebook using Python. The below list highlights the range of Python packages used within the notebooks available on the GitHub link:

* Sklearn
* SMOTN
* Pandas
* XGBoost
* NumPy
* Seaborn
* Matplotlib

### Data Imbalance

Data imbalance causes prediction bias, causing models to ignore the rare cases, extreme AQI in this case. In the context of this report, it is important for a model not to overlook the higher AQI values and to be able to implement them in their predictions.

To best view data imbalance is through the skew of data. Figure 22 highlights the skew of the AQI, emphasising a data imbalance within the class. The severity of the imbalance would mean that models would struggle with learning and predicting AQI values of over 200, especially if they are patterns that cannot be recognised within the training set.

A graph of a distribution of a number of data

Description automatically generated

Figure 22. AQI Histogram

To prevent the data imbalance, I also trained models of a new dataset that included synthetic data points. The method used was a variation of SMOTE, called SMOTER. This involved synthetic data points created to balance AQI, done using a kNN algorithm to create data points from existing, neighbouring data. The Python package included parameters that allowed me to customise the number of nearest neighbours to use. Figure 23 shows how the skew of the AQI changed after resampling, with the higher values of AQI better represented.

A graph of a distribution of a number of data

Description automatically generated

Figure 23. AQI Class after resampling

To further improve the balance of the class, additional research will need to be conducted to include features outside of the ones engineered in this report. This is due to the features in this dataset not directly causing the higher AQI values, and thus, any resampling done on the dataset can potentially lead to overfitting as we do not understand the cause for the extreme AQI values.

# Results

### Model Performances

Figure 2 explains how the AQI index works for classifying air quality, with categories spanning 50-100 points. For a model to be remotely successful in predicting AQI, the error in the predictions must be sufficiently small to not predict the wrong level of AQI. Figure 24 shows the performances of all models on the test set, with the Random Forest Regressor performing the best. This model exceeded on the test set for both datasets, with and without SMOTER applied. The RMSE was small enough that AQI prediction can only be 1 class away from the true value, at worst. This conclusion applies to all models in the table, with all of them performing well during the testing phase.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Without SMOTER | | | With SMOTER | | |
| MSE | RMSE |  | MSE | RMSE |  |
| OLS | 622.1035 | 24.9420 | 0.8345 | 686.9436 | 26.2096 | 0.8178 |
| Random Forest | 559.1032 | 23.6454 | 0.8517 | 630.5646 | 25.1110 | 0.8327 |
| SVR | 630.6412 | 25.1126 | 0.8327 | 635.4782 | 25.2087 | 0.8314 |
| XGBoost | 655.7954 | 25.6085 | 0.8260 | 751.1734 | 27.4076 | 0.8007 |
| XGBoost + NN | 669.9045 | 25.8806 | 0.8223 | 691.7777 | 26.3017 | 0.8165 |

Figure 24. Model Performance Metrics

All models were cross validated to ensure they were not overfitting to the training set. Here, we see that the Random Forest is no longer the best-performing model, and the hybrid stacked model is now performing better, as shown in Figure 25, suggesting the stacked model is less prone to overfitting.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Without SMOTER | | | With SMOTER | | |
| MSE | RMSE |  | MSE | RMSE |  |
| OLS | 962.6065 | 30.1744 | 0.7781 | 962.6065 | 30.1744 | 0.7781 |
| Random Forest | 901.6064 | 29.1414 | 0.7924 | 1132.3372 | 32.4356 | 0.7400 |
| SVR | 887.8931 | 29.0915 | 0.7938 | 900.7386 | 29.2871 | 0.7896 |
| XGBoost | 935.7752 | 29.8375 | 0.7801 | 969.5623 | 30.4269 | 0.7698 |
| XGBoost + NN | 885.1948 | 29.0424 | 0.7948 | 910.8698 | 29.5478 | 0.7868 |

Figure 25. Cross Validation Model Performance Metrics

The residuals for the random forest and stacked models can be found in figures 26 and 27. The observed data aligns with performance metrics for both models, indicating that the stacked model performs better across the entire dataset. However, it encounters significant residual errors when predicting higher values of AQI, signifying room for improvement within the model.

A graph with blue dots

Description automatically generatedA graph with blue dots

Description automatically generated

Figure 26. Residuals for Random Forest Model. Figure 27. Residuals for Stacked Model

### SMOTER

With the emphasis on creating synthetic points for extreme AQI, models run the risk of overfitting the areas being synthesised. Whilst the performance metrics suggest all the models performed worse with the synthetic data, this does not necessarily mean the data balancing did not do its job.

Figures 28 and 29 show the residuals for the different datasets, showing that the residuals are closer to zero for increasing AQI values, thus yielding the results intended after SMOTER was applied. However, as the extreme cases were still misrepresented, the residuals are relatively high for those AQI values.

A graph with blue dots

Description automatically generated

Figure 28. Residuals for SVR Model Without SMOTER

A graph with blue dots

Description automatically generated

Figure 29. Residuals for SVR Model With SMOTER

### Optimal Model Parameters

Due to the emphasis on cross-validation, trail runs with different parameters were all done using the randomised search function in Python. By providing a list of different parameters to use, the algorithm can cross-validate across all combinations to provide the parameters that yield the best results.

However, using parameters that do not intuitively align with the model can suggest issues such as overfitting and a lack of generalisation, two issues that can severely impact the usability of a model.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Metric Evaluation | | | Cross Validation | | |
| MSE | RMSE |  | MSE | RMSE |  |
| OLS | 622.1035 | 24.9420 | 0.8345 | 962.6065 | 30.1744 | 0.7781 |
| Random Forest | 559.1032 | 23.6454 | 0.8517 | 901.6064 | 29.1414 | 0.7924 |
| SVR | 630.6412 | 25.1126 | 0.8327 | 887.8931 | 29.0915 | 0.7938 |
| XGBoost | 655.7954 | 25.6085 | 0.8260 | 935.7752 | 29.8375 | 0.7801 |
| XGBoost + NN | 669.9045 | 25.8806 | 0.8223 | 885.1948 | 29.0424 | 0.7948 |

Figure 30. Performance Metrics for Test Set vs Cross Validation

The below list of parameters was used to obtain the results found in Figure 30.

|  |  |
| --- | --- |
| Model | Features |
| Random Forest | * max\_depth: 19 * min\_sample\_leaf: 8 * n\_estimators: 84 |
| SVR | * C: 5.908361216819946 * Epsilon: 0.8761761457749352 * Gamma: ‘Auto’ * Kernel: Linear |
| XGBoost | * colsample\_bytree: 0.7361074625809747 * learning\_rate: 0.04587827378149051 * max\_depth: 8 * n\_estimators: 180 * reg\_alpha: 0.07607850486168975 * reg\_lambda: 0.6612771975694962 * subsample: 0.8854835899772805 |
| XGBoost + NN | * nn\_activation: ReLu * nn\_\_alpha: 0.002284404372168336 * nn\_\_hidden\_layer\_sizes: 100, * nn\_\_solver: adam * xgb\_\_colsample\_bytree: 0.9416401294594341 * xgb\_\_learning\_rate: 0.10730350630158218 * xgb\_\_max\_depth: 9 * xgb\_\_n\_estimators: 141 * xgb\_\_subsample: 0.6781489190384875 |

Within the list of parameters produced, only a few raise cause for concern. High values for C can result in overfitting whilst large can result in the SVR model ignoring important data points. Furthermore, the combination of the xgb\_\_n\_estimators in the stacked model and the learning rate together can potentially limit convergence.

During the RandomSearch cross-validation of parameters, certain parameters were influential in the model’s performance, specifically the mean square error. Figure 31 highlights the parameters for each model that were strong indicators for lower mean square errors during the testing phase, with additional information found in the Jupyter Notebook.

|  |  |
| --- | --- |
| Model | Features |
| Random Forest | Higher Depth  Medium-Higher n\_estimators  Square Root feature |
| SVR | Linear Kernel  Auto Gamma type  Larger values for C |
| XGBoost | Higher Depth  Lower Learning Rates  Positive Alpha/Lambda |
| XGBoost + NN | Higher Depth  Lower Learning Rates  Positive Alpha/Lambda  Moderate Alpha values |

Figure 31. Parameter influence for all models

For the stacked model, it is also worth noting that cross-validation on different NN solvers and activation methods was too computationally expensive to implement.

### Model Comparisons

An ANOVA test, followed by Tukey HSD analysis, was performed to directly compare each model to check if there were any statistical differences between them.

#### Without SMOTE

The ANOVA test yielded the below statistics, concluding that there was statistical differences between the models.

|  |  |
| --- | --- |
| F-Statistic | 24.05884983751123 |
| P-Value | 4.758283790238124e-11 |

The Tukey test found that the best models were the SVR and Random Forest models, with both outperforming the hybrid model but not enough statistical difference to suggest a difference between those two. The below table shows the full results of the Tukey test.

A screenshot of a graph

Description automatically generated

#### With SMOTE

The ANOVA test for the data with SMOTER provided the same outcome but with different values. The significantly higher F-statistics suggest there was a significantly large variance between the models, and this was supported by an extremely low P-value.

|  |  |
| --- | --- |
| F-Statistic | 59.26548983974521 |
| P-Value | 7.449718723380513e-20 |

The Tukey test results mirrored the non-SMOTE findings, with SVR and Random Forest again outperforming the hybrid model. However, the increase in F-statistic with SMOTE suggests that the models' performance spread became more hightened, potentially due to the synthetic data created by SMOTER, which might have increased the differences in how each model handles extreme values.

A table with numbers and letters

Description automatically generated

The interesting conclusion from this is that the Random Forest and SVR models were outperformed by the hybrid model on cross-validation, but they were statistically better. Further tuning to the models is needed, ones not motivated by performance metrics.

### Additional Tuning

During the data imbalance step of the investigation, I tried different iterations of the SMOTER to see what effects the parameters had on the data. I began by changing the sampling method to balanced, which yielded and MSE of 687.13 and of 0.8177 compared to 630.56 and 0.8327 for the Random Forest, respectively.

Furthermore, I also varied the value of but ran into computational errors when synthesising data points. Increasing the value of to 8, also yielded slightly worse values than the original data, with the MSE being 657.07 and 0.8257.

Parameter tuning using GridSearch rather than RandomisedSearch would have yielded better results, but due to the computational cost of implementation, it was not an option.

# Discussion

### Evaluation

The models all performed relatively well with the task of predicting AQI, with the features explaining over 75% of the AQI variance. Whilst the additional features engineered had minimal impact, they improved the model’s ability to generalise across different regional conditions. The unexplained variance stems from additional factors beyond the scope of this report, likely from less regulated pollutants. Additionally, data on socioeconomic factors like population density or household fuel use might explain further variance by linking human activities to pollution levels more directly.

I believe that the variation in location and weather data is sufficient to capture underlying patterns.

### Comparison to other Work

The implementation of the stacked model in this report included sufficient varying data to provide a justifiable reason to apply the model to different areas and climates. The models used in the stacked are less susceptible to overfitting, something Sun & Liu struggled with when implementing their hybrid model. Furthermore, the implementation of regional features added to the generalisation of all models, improving on the work conducted by the authors in the literature review.

### Limitation

The main two limitations in this report were computational cost and the complexity of the models. As the complexity of the models increased, so did the computation time, along with interpretability being sacrificed. A lack of interpretability results in less impactful real-life insights. This becomes an issue when a model predicts the higher AQI values, especially because the accuracy of these values is also low.

### Impact

The analysis holds significant value for various sectors. Government agencies can use insights on AQI to implement effective environmental policies and regulations, whilst industries can use the data on the pollutants to produce cleaner technologies, ultimately reducing their carbon footprint. Most importantly, the pollutants in the air can have serious health implications for all people, regardless of existing health conditions. Short-term implications include respiratory infections, aggravated asthma and other various internal issues, whilst long-term exposure can lead to certain types of cancer and Dementia. As a result, countries must understand the pollutants that have the largest impact on the air and how regional factors influence AQI predictions so they can look to improve the everyday lives of their citizens.

# References

Wikipedia (n.d.) Air quality index. Available at: https://en.wikipedia.org/wiki/Air\_quality\_index (Accessed: 05 October 2024).

California Air Resources Board (n.d.) Inhalable particulate matter and health. Available at: https://ww2.arb.ca.gov/resources/inhalable-particulate-matter-and-health (Accessed: 05 October 2024).

BMJ (2023) Impact of fossil fuels on health. British Medical Journal. Available at: https://www.bmj.com/content/383/bmj-2023-077784 (Accessed: 12 October 2024).

World Health Organization (n.d.) Air pollution. Available at: https://www.who.int/health-topics/air-pollution (Accessed: 12 October 2024).

Rao, R. (2020) Air quality data in India (2015-2020). Kaggle. Available at: https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india (Accessed: 19 October 2024).

Devrath, M. (2024) Indian Cities Weather Data (2010-2024). Kaggle. Available at: https://www.kaggle.com/datasets/mukeshdevrath007/indian-5000-cities-weather-data (Accessed: 19 October 2024).

Emmins, L. (2024) CSC 40080 code repository. GitHub. Available at: https://github.com/Luke1709/CSC-40080.