Prediction of Nitrogen Dioxide Levels Based on Multivariate Linear Regression and Linear Support Vector Machines ECS784P

Abstract—This project report will present a prediction model on Nitrogen Dioxide levels based on some numerical concentrations of pollutants and greenhouse gasses. Measurements were taken from the roadside and background atmosphere of London. The two machine learning methodologies used are the linear regression model and support vector regression. Both were implemented using scikit-learn default parameters.

The chosen dataset has one feature of the type object and fourteen numerical features of type float64. The feature's correlation values are obtained using the Pearson correlation function. The report will go into detail on the pre-data processing steps taken and justify any feature reductions steps. A brief introduction to the learning machine methodologies used will be provided alongside the analysis and testing carried out. A literature review will be attached as well to observe any work in this field with machine learning methodologies. To conclude the findings and conclusion will be provided in the final section.

Keywords—Linear Regression, Support Vector Regression, Air quality, Nitrogen Dioxide

I. INTRODUCTION

The project aims to estimate a continuous value for Nitrogen Dioxide in (ug/m3) units when given data on the other main pollutants present in the atmosphere of London.

Air Pollution

Air pollution refers to the contamination of the atmosphere (indoor and outdoor) with the release of particles and noxious gasses. Most of these emissions end up being manmade and a few portions being natural. Natural emissions can be found in plants, soil and the ocean, manmade emissions are usually caused by the combustion of fossil fuels such as coal in factories or petrol/diesel in cars [6].

The main concern brought by these emissions is their effect on human health. Understanding the effect of these pollutants is a very complex problem and hard to monitor or test. Some guidelines have been created on the acceptable levels of these pollutants in the air. Due to weather, the numbers tend to fluctuate a lot given that wet or windy conditions are very good at blowing away and removing pollutants concentrations away. The concentration of these pollutants rises to dangerous levels in towns/cities when the weather tends to be stagnant [6].

There is a group of pollutants that are constantly being monitored in London. These include carbon monoxide (CO), nitrogen dioxide (NO2), ground-level ozone (O3), particles PM10 and PM2.5 and sulphur dioxide (SO2). In the 2010 Londoner Survey [1], air quality was the top concern among Londoners. Poor air quality leads to premature deaths in vulnerable people and damage to the lungs. This damage is pronounced to people suffering from asthma. It is estimated that there are 40,000 premature deaths caused by air pollution every year in the UK.

Nitrogen dioxide (NO2) belongs to a group of pollutants referred to as nitrogen oxides. They are the greatest urban air pollution component and the values of nitrogen dioxide in London have not been reduced to the target values set by the Mayor of London, despite the many initiatives in place. Some of these initiatives include: promoting the shift of cleaner forms of transport (electric cars/buses) and introducing ULEZ and LEZ zones in areas where nitrogen dioxide levels are high.

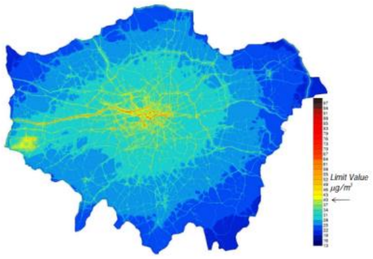

Figure 1: NO_2 annual mean concentrations in London (µg/m3)

High concentrations of NO2 irritate and inflame the lining of the lungs and throat, increase the chance of asthma attacks and cause breathing difficulties. This model could be used to predict NO2 concentrations without having to take recordings specifically for this pollutant and could be a way to obtain an estimate for the NO2 concentration at a moment's notice.

OBJECTIVES

- Identify which features highly correlate to the concentration of N02 in the atmosphere.
- Use of two machine learning methodologies to produce an accurate estimation model.
- Conduct exploratory analysis on the dataset features and entries. and
- Use common python libraries such as pandas, to visualize and present the data so it's easy to understand.
- Analyse the strengths and weaknesses of both regression models and evaluate their accuracy.

II. LITERATURE REVIEW

Air pollution has been widely researched and models have been created to predict pollution levels given a dataset or in other cases real-time data from meteorological devices. NO2 is a popular parameter that is widely associated with man-made air pollution as most of it is produced from the exhaust gases of cars.

The first source for our literature review is a report that shows an example of using Big Data Analytics in an air pollution problem, titled "RAQ–A Random Forest Approach for Predicting Air Quality in Urban Sensing Systems" and written by Ruiyun Yu and company [2]. The report aimed to predict concentrations of PM2.5 particles using meteorological data, real-time traffic data from Google Maps and POI (point of interest distribution). The data was being generated by 11 weather stations across Shenyang city, China. The machine learning methodology used was a random forest model and the RAQ algorithm was created to extract relevant data for the model through feature selection. The dataset created from the RAQ algorithm was then fed into a random forest classifier that then predicted the PM2.5 levels. The dataset created was tested with different machine learning algorithms like Naïve Bayes, Logistic, Decision Trees, ANN and RAQ. RAQ model showed the best performance outperforming the Naïve Bayes algorithm with a prediction accuracy of over 80% [2]. This report highlights that there is a need for urban monitoring systems in busy cities to employ machine learning algorithms to predict pollution levels so that the government and other bodies can make better decisions on what initiatives to put in place to counter air pollution. It also highlights how easy it is to obtain the data required to train these models.

The second source is a research paper is titled "APPLYING MACHINE LEARNING TECHNIQUES IN AIR QUALITY PREDICTION" written by Elias Kalapanidas and Nikolaos Avouris [3]. In this report, they produce an air quality prediction system for the air quality monitoring station Athens AQOC. The main interest is NO2 concentrations. The system obtains data from database records to train the model. When new air pollution records are obtained, it is sent to the model alongside the current local meteorological data to predict if there will be a spike in pollutant concentration. The machine learning methodology used is Artificial Neural Network and it predicts if there will be an all-time high in the concentration of NO2 (a classification problem). As discussed earlier, local weather has a great effect on pollutant levels and this model looks to predict if due to the weather there will be a spike in the NO2 pollutant concentration that could be dangerous for people's health in that specific urban area. The accuracy of the model after ten-cross fold validation was around 60-80% [3].

The third source is a report was titled "Forecasting daily ambient air pollution based on least squares support vector machines" and was written by W.F. Ip and company [4]. The aim was to show Least squares support vector machines (LS-SVM) algorithms performed better than the pollutant level predictive models using multi-layer perceptrons (MLP). The datasets used were from the Macau peninsula, China and the features of the dataset consisted of common pollutants such as SO2 and NO2 and weather data such as temperature, humidity and wind speed. The predicted results had very low levels of error (7% - 15% for NO2) [4]. This report shows that support vector machines can perform regression problems related to air quality and NO2 levels which will be carried out for this project.

The final source is a report titled "Deep learning architecture for air quality predictions" written by Xiang Li and company [5]. In this paper a novel deep learning approach is used that looks to predict air quality by using a stacked autoencoder model to extract relevant features to air quality and is trained in a greedy-layer wise manner to predict air quality in different locations and not be affected by seasons or weather. A comparison is also done with other machine learning algorithms like SVR which will be used in this project. The pollutant they focused on was PM 2.5 particles and the dataset used was the hourly records of air pollutants obtained from twelve different stations. The model presented in the paper achieved better performance compared to SVR which might imply that deep learning neural networks can be trained to be a robust model for air quality predictions [5].

III. DATA MANAGEMENT

External Libraries

NumPy: A python library that's fundamental for scientific operations. Provides fast mathematical operations on multidimensional arrays and matrices by using high-level functions that are easy to use and implement.

Pandas: Very useful package widely used for data science/data analytics and machine learning. It uses NumPy and makes it simple and easy to carry out repetitive tasks on datasets. Some of the functions include data cleansing, data visualization and data filling.

Scikit-learn: A machine learning library that was used for this project to easily implement the machine learning algorithms for linear regression and support vector regression. It also supports other popular algorithms and can be used for classification problems too.

Matplotlib: A library to visualize data, different types of graphs can be produced from the data frame and developed. Graph axis and scales can be customized easily and also allow for colour customization.

Data Source and Description

The data was found on Kaggle in the following reference link [11].

The dataset has fifteen features, some of these features are useful for our model and some have been removed. The shape of our dataset is 15x132, with fifteen columns and 132 rows. The feature "month" is of type object and not very useful for our model, it acts as an ID. The rest of the features are continuous variables of type float64, as there are decimals present. The value represents the concentration of an air pollutant and is given in micrograms (ug/m^3). This unit refers to one-millionth of a gram and can be converted into PPM (parts per million), where 1PPM = 1000ug/m3.

	Month	London Mean Roadside Nitric Oxide (ug/m3)	London Mean Roadside Nitrogen Dioxide (ug/m3)	London Mean Roadside Oxides of Nitrogen (ug/m3)	London Mean Roadside Ozone (ug/m3)	London Mean Roadside PM10 Particulate (ug/m3)	London Mean Roadside PM2.5 Particulate (ug/m3)	London Mean Roadside Sulphur Dioxide (ug/m3)	London Mean Background Nitric Oxide (ug/m3)	London Mean Background Nitrogen Dioxide (ug/m3)	London Mean Background Oxides of Nitrogen (ug/m3)	London Mean Background Ozone (ug/m3)	London Mean Background PM10 Particulate (ug/m3)
0	2008-01-01	NaN	55.502688	NaN	29.512097	24.969086	14.678763	4.217742	NaN	42.338710	NaN	36.942204	18.817204
1	2008-02-01	NaN	75.922414	NaN	20.317529	39.477011	28.772989	7.553161	NaN	60.237069	NaN	26.425287	31.896552
2	2008-03-01	NaN	55.610215	NaN	40.103495	21.569892	12.300135	3.868280	NaN	39.801075	NaN	50.227151	15.477151
3	2008-04-01	NaN	61.756944	NaN	37.884722	28.740278	20.461111	4.475000	NaN	44.009722	NaN	50.133333	21.729167
4	2008-05-01	NaN	62.903226	NaN	46.266129	34.611559	27.508065	4.634409	NaN	44.141129	NaN	60.512097	29.545699
5	2008-06-01	NaN	49.161111	NaN	39.836111	23.198611	16.010057	3.593056	NaN	31.241667	NaN	51.326389	18.250000
6	2008-07-01	NaN	48.444892	NaN	34.982527	22.958333	14.240591	3.100806	NaN	31.216398	NaN	46.623656	17.204301
7	2008-08-01	NaN	41.072581	NaN	30.021505	20.693548	11.452957	2.155914	NaN	27.850806	NaN	37.094086	15.508065
8	2008-09-01	NaN	54.080556	NaN	22.375000	28.227778	17.979167	3.748611	NaN	41.215278	NaN	28.886111	22.244444
9	2008-10-01	NaN	56.658602	NaN	19.337366	23.002688	12.918011	4.305108	NaN	43.813172	NaN	25.427419	16.469086

Figure 2: Data sample from the original dataset

Dealing with Missing Data

Missing values in a dataset are common when dealing with real-life values. It can be caused by data corruption or a simple failure of recording the data. These missing entries must be handled during the pre-processing stage as machine learning algorithms do not support missing values. There are multiple ways of dealing with missing data such as:

- Replace missing values with the average for continuous variables.
- Replace missing values with the mode for discrete variables
- Remove the rows where there are missing values
- Input the data manually through background knowledge or experience on the subject.
- Ignore the missing entries through an algorithm like KNN-means which can ignore a feature when a value is missing or a K-means algorithm that clusters the data in groups.
- Replace missing entries with estimated predictions from the trained model on the rest of the features.

In the project dataset, there are missing 24 values and no duplicate values were found. I replaced the missing values with the mean value for that feature.

This method was chosen as it is easy to implement on small datasets. It also works well since we are dealing with numerical continuous variables. This will prevent the loss of information that comes from deleting the missing rows in our already very small dataset of 132 entries.

Feature selection

The dataset contains fifteen features, but after dropping the "month" column (it was not useful for the machine learning model) and merging the entire roadside and background readings into a single mean of the two for each pollutant. The final version of the dataset has only seven features ready to be trained using a machine learning algorithm. We can also visualize the overall pollutant levels in London, nitrogen dioxide being the most common pollutant and sulphur dioxide the least common.

	London Nitric Oxide	London Nitrogen Dioxide	London Oxides of Nitrogen	London Ozone	London PM10 Particulate	London PM2.5 Particulate	London Sulphur Dioxide
0	50.231125	48.920699	97.936703	33.227151	21.893145	13.985745	3.895161
1	50.231125	68.079741	97.936703	23.371408	35.686782	21.032857	7.143678
2	50.231125	47.705645	97.936703	45.165323	18.523522	12.796430	3.077285
3	50.231125	52.883333	97.936703	44.009028	25.234722	16.876918	3.855556
4	50.231125	53.522177	97.936703	53.389113	32.078629	22.042445	4.442204
5	50.231125	40.201389	97.936703	45.581250	20.724306	14.311911	3.070139
6	50.231125	39.830645	97.936703	40.803091	20.081317	13.069892	2.797043
7	50.231125	34.461694	97.936703	33.557796	18.100806	11.327168	2.122312
8	50.231125	47.647917	97.936703	25.630556	25.236111	16.651840	3.402778
9	50.231125	50.235887	97.936703	22.382392	19.735887	12.269489	3.571909

Figure 3: Dataset after feature selection

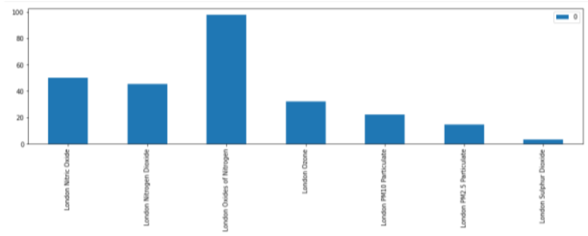

Figure 4: Average air pollutant levels in London

Normality Testing

It is used to identify if a variable is normally distributed. Normal distribution will have a bell-shaped curve that is symmetric around the mean. A variable that is normally distributed will perform better in linear regression models as the algorithm assumes errors/residuals follow a normal distribution themselves. Normal distribution problems tend to be more easily solvable [7].

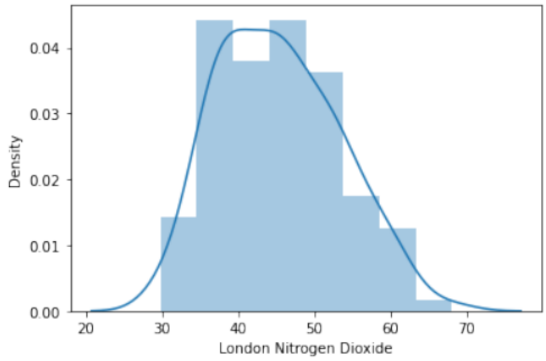

Figure 5: Histogram of Nitrogen Dioxide variable

We can conclude that our dependent variable has a normal distribution thanks to the bell-shaped curve. It also tells us it is slightly skewed towards the left; this is likely caused by the high outlier values caused by stagnant weather. We do not want to get rid of these outliers as they should also be part of the model.

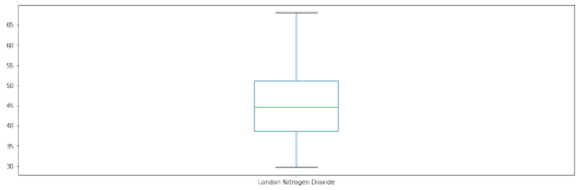

Figure 6: Box plot of Nitrogen Dioxide variable

Box plots are also a useful way to see if a variable follows a normal distribution. If there is a normal distribution then the median line (green line) will be in the centre of the mean (blue box).

Other statistical tests can be carried out, such as the Chi-Square Normality test that will produce a p-value. If the p-value obtained is less than 0.05 then we assume the distribution is not gaussian/normal. If the value is greater than 0.05 then we assume it's a normal/gaussian distribution [7].

IV. Methodologies

The project will take a supervised learning approach as the dataset has labelled features. The dataset can be used to train an algorithm to predict a value/outcome accurately. Supervised learning algorithms adjust for the correct answer and in this case, we are trying to predict the value of nitrogen dioxide.

Supervised learning can be separated into two different problems: a regression problem or a classification problem. In simple terms, classification involves predicting a label for the data. There can be multiple classes or the data can be classified into a categorical target y with two outcomes. Regression on the other hand is used to predict a feature exact quantity (numerical target y) given some data or observation.

We have chosen to make a regression model and our model accuracy will be determined by how close our predicted value is to the actual value in the training set of data and test set of data. This accuracy is evaluated using root mean squared error and R2 score.

Linear Regression Algorithm

Linear regression is a supervised learning algorithm that finds the line of best fit between an independent variable and a dependent variable (a linear relationship). It is one of the simpler models for prediction and serves as the foundation for more complex statistical and machine learning models such as neural networks. Linear regression can be performed with two variables or more.

$$y = b_o + b_1x_1 + b_2x_2 + b_3x_3 \;....+b_nx_n$$

Figure 7: Equation for Multiple linear regression

In this equation y is the prediction of the dependent variable, b0 is the intercept and b1, b2, b3…bn are the coefficients/slopes of the independent variables x1,x2,x3...xn [8].

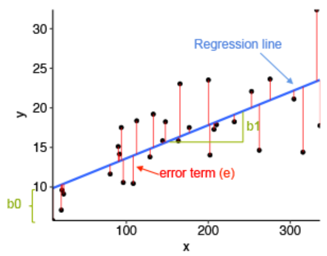

Figure 8: Illustration of a Linear Regression Algorithm

The linear regression model aims to find the line of best fit, intercept value and coefficients such that the error term is minimized. The black dots are the observations of the independent variables (x).

Support Vector Regression Algorithm

Support vector machines are well known to solve supervised learning classification problems but their use in regression is less documented. SVR is an algorithm that is very similar to linear regression algorithms, it looks to produce a line of best fit, calculate coefficients and the right intercept to minimize error. It will however give the model more flexibility because instead of forcing the model to go for the lowest error value possible, it will be allowed to operate in a certain error margin [9].

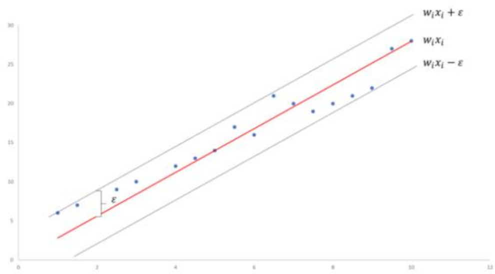

Figure 9: Illustration of a Support Vector Regression Algorithm

V. ANALYSIS, TESTING AND RESULTS

After performing pre-processing steps to the dataset and merging our features using dimensionality reduction methods, I tested the two regression algorithms on the dataset. The algorithms were implemented using the default hyperparameters from the Scikit-learn library. The chosen dependent variable (y) was nitrogen dioxide and the other six features were in the variable (X).

The data was split into 70% training and 30% test data and six-fold cross-validation was used to determine the model accuracy. The training accuracy obtained was 41% and the cross-validation accuracy value was 20% roughly for both models.

I decided to proceed with reducing the number of features and perform data normalization. In theory, regression is insensitive to standardization but it is shown to improve the numerical stability of the model and speed up the training process. Standardization gives the same consideration for each variable and this might not always be useful [10]. The normalization method implemented was log transformation.

```
data['London Nitrogen Dioxide'] = np.log(data['London Nitrogen Dioxide'])
```

Figure 10: Example of data normalization using log transformation

To reduce the number of features further I decided to use Pearson correlation Matrix. Using Pearson Correlation (corr () function) we obtain all the correlation values for each feature. A positive correlation is between (0.5-1) and a negative correlation is between (-0.5 and -1). There is no correlation between those values.

Figure 11: Pearson Correlation Matrix

We looked at what features correlated weakly with nitrogen dioxide, and features that correlated too highly with each other (duplicates). In this case, I decided to drop the variables: ozone and sulphur dioxide concentrations as they were weakly correlated to nitrogen dioxide concentrations.

This left me with four independent variables: London Nitric Oxide, London Oxides of Nitrogen, PM 10 particulate and PM 2.5 particulate. My dependent variable is London Nitrogen Dioxide.

VI. RESULTS

I trained the new models using the new set of independent variables. The model should produce four coefficient values. Cross-validation is used to test the model's performance on unseen data. If the model performs very well in the training set but poorly in the test set, then we can consider the case of overfitting.

The mean square root measures the average of squares of the prediction error, it is better when it's closer to 0. The R-squared: coefficient of determination measures the proportion of variance in the dependent variable that is predictable by the independent variables. A value of 1 is close to a perfect regression model.

```
from sklearn.linear_model import LinearRegression
LR_model = LinearRegression() #Performing linear regression
classify(LR_model, X,y)

Accuracy is:  81.7502041788675
Cross validation Accuaracy:  64.84631542614872

Coefficients:
 [0.00060568 0.43801988 0.1374621  0.1591829 ]
Mean squared error: 0.01
Coefficient of determination: 0.82
```

Figure 12: Linear Regression Results

```
from sklearn.svm import SVR
SVR_model = SVR() # performing Support Vector Regression
classify(SVR_model, X,y)

Accuracy is:  83.57651124570415
Cross validation Accuaracy:  62.306997935854206

Coefficients:
 [[0.04684706 0.25387815 0.09797599 0.27994807]]
Mean squared error: 0.01
Coefficient of determination: 0.83
```

Figure 13: Support Vector Regression Results

VII. CONCLUSION

This project aimed to create a model that could predict nitrogen dioxide levels using linear regression and support vector regression algorithms. Both models produced, have an accuracy greater than 80%. Through feature selection and observing weight coefficients we can determine that concentration values for Oxides of Nitrogen and PM 2.5 particulates are very useful in predicting nitrogen dioxide values compared to the other air pollutants. This would allow air monitoring systems to predict a value for nitrogen dioxide concentration by just measuring the concentration of oxides of nitrogen and PM 2.5 particulates.

Limitations of the project include the lack of weather features in the dataset. The weather would have been a useful feature for the prediction of nitrogen dioxide levels, spikes in the concentrations, in particular. The dataset also had very few entries and a better approach would have been to use the hourly dataset available.

This project can be switched from a regression problem to a classification problem, where we can predict if the concentration of nitrogen dioxide is safe or alerts and precautions need to be taken. Using real-time data would be interesting to implement as new challenges would be introduced such as taking into account training speed and prediction speed. Data would arrive from different sources and have to be cleansed and prepared using other algorithms.

More complex algorithms such as deep learning and neural networks could be used to create these models. They are seen to perform faster and learn key features more accurately. Consulting with professionals and checking if any other important missing features could be used to improve our model's accuracy.

VIII. REFERENCES

Report/Journal references

[1] 'Clearing the Air the Mayor's Air Quality Strategy' Mayor of London - 2010
URL:https://www.london.gov.uk/sites/default/files/Air_Quality_Strategy_v3.pdf

[2] 'RAQ-A Random Forest approach for Predicting Air Quality in Urban Sensing Systems' Ruiyun Yu, Yu Yang, Leyou Yang, Guangjie Han and Oguti Ann Move-2016
URL:https://www.mdpi.com/1424-8220/16/1/86/htm

[3] 'Applying Machine Learning Techniques in Air Quality Prediction' Elias Kalapanidas and Nikolaos M. Avouris – 1999
URL:https://www.researchgate.net/publication/2611442_Applying_Machine_Learning_Techniques_in_Air_Quality_Prediction

[4] 'Forecasting Daily Ambient Air Pollution Based on Least Squares Support Vector Machines' CM. Vong, and PK Wong – 2010
URL:https://ieeexplore.ieee.org/document/5512401

[5] 'Deep Learning Architecture for Air Quality Predictions' Xiang Li, Liang Peng, Yuan Hu, Jing Shao and Tianhe Chi – 2016
URL:https://pubmed.ncbi.nlm.nih.gov/27734318/

Web References

[6] 'London Air Quality Network Guide'
URL:https://www.londonair.org.uk/londonair/guide/WhatIsPollution.aspx

[7] '10 Normality Tests in Python' Sivasai Yadav Mudugandla – 2020
URL:https://towardsdatascience.com/normality-tests-in-python-31e04aa4f411

[8] 'All you need to know about your first Machine Learning Model – Linear Regression' Deepanshi – 2021
URL:https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/

[9] 'An Introduction to Support Vector Regression (SVR)' Tom Sharp – 2020
URL:https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2

[10] 'Understand Data Normalization in Machine Learning' Zixuan Zhang – 2019
URL:https://towardsdatascience.com/understand-data-normalization-in-machine-learning-8ff3062101f0

[11]'Kaggle - London Air Quality'
URL:https://www.kaggle.com/zsn6034/london-air-quality?select=data