Air quality prediction

Ger Inberg

Machine Learning Engineer Nanodegree - Capstone Project

Ger Inberg April 25th, 2017

I. Definition

Project Overview

While travelling in South East Asia, I noticed the air quality issues in some bigger cities. It affects peoples lives directly because they might get breathing problems, will stay only inside buildings and/or are wearing masks. When people will know that at a certain time the air quality is bad, they can take measures to prevent possible (health) problems. Because I am curious about the current air quality prediction systems and if it can be improved I have chosen this as my subject.

In the "Human Health Effects on Air Pollution" study (Marilena Kampa, Elias Castanas 2007) the relation between air quality and the health of the people having to deal with that air have been shown. This has led to the introduction of the (Airnow 2017). The AQI is an index for reporting daily air quality. It tells how clean or polluted the air is, and what might be the associated health effects.

Air Quality Index (AQI) Values	Levels of Health Concern	Colors
When the AQI is in this range:	*..air quality conditions are:*	*...as symbolized by this color:*
0-50	Good	Green
51-100	Moderate	Yellow
101-150	Unhealthy for Sensitive Groups	Orange
151 to 200	Unhealthy	Red
201 to 300	Very Unhealthy	Purple
301 to 500	Hazardous	Maroon

Figure 1: Air Quality Index table

The EPA's Air Quality Index is used daily by people suffering from asthma and other respiratory diseases to avoid dangerous levels of outdoor air pollutants, which can trigger attacks. There are already some systems that can predict air quality, however I would like to see if a more accurate model can be build.

The model I build could be used as the basis for an early warning system that is capable of accurately predicting dangerous levels of air pollutants on an hourly basis. In the Air Quality Prediction Kaggle competition (Kaggle 2012) this has been done already. However since this competition is already 5 years old, I want to use new techniques (for example XGBoost) to see if I can improve upon this.

Problem Statement

Certain health problems are related to the air quality index. In order to prevent health issues due to bad air quality it is important to have an accurate estimate of it. When dangerous levels are reached, certain preventive measures can be taken, like to stay inside in the house.

The best solution would be to make the air cleaner for example by less pollution. However that is not a solution that can be reached within in short time frame. However what we can do is to create awareness about the air quality and to signal when the level gets dangerous. People can act upon this signal by taking preventive measures and so some health problems can be prevented.

For every pollutant in the training data (see next section), the air quality level is expressed by a number, where a higher number means more pollution / worse air quality. The algorithm has to predict the numeric value of the air quality for each pollutant. Thefore the algorithm has to solve a regression task.

There are multiple machine learning algorithms that can solve a regression task. The eXtreme Gradient Boosting (xgboost) is nowadays a very popular algorithm and wins many competitions on Kaggle (Github 2016). Therefore I will be using xgboost.

Metrics

To measure the performance of a regression model, multiple metrics can be used, such as Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE).

The MAE is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. It is defined by the average of the absolute differences. The RMSE also takes into account the difference between predictions and eventual outcomes, however it severely punishes large errors. The metrics can be calculated in R as follow:

```
MAE <- sum(abs(y-y_pred)) / length(y)
RMSE <- sqrt(mean((y-y_pred)^2))
```

To determine which metric to use, it is important to think about what a good model is. First, a good model should have predictions that are close to the eventual outcomes. Second, it should take into account the distance between the predicted values and the outcomes. Both of these metrics do this. If we have 2 predictions and the 2nd one is twice as big as the first one, the contribution of the 2nd one to MAE is just twice as big as the first one. For RMSE, the contribution of the 2nd forecast will be four time as much as the 1st forecast, because of the square. It is not needed to severly punish for large errors, that's why I will use the MAE.

II. Analysis

Data Exploration

The datasets that I will use are provided by Kaggle, a description of the data can be found on [their data page] (https://www.kaggle.com/c/dsg-hackathon/data). The datasets consists of SiteLocation data, TrainingData and a sample submission file. The sample submission file contains the test data.

The training data consists of 37821 rows while the test data consists of 2100 rows. So, the test data is only about 5.5% of the training data, which means there is relatively a large amount of training data.

The training data consists of the following features:

- rowID
- chunkID
- position_within_chunk (starts at 1 for each chunk of data, increments by hour)

- month_most_common (most common month within chunk of data–a number from 1 to 12) weekday (day of the week, as a string)
- hour (a number from 0 to 23, local time)
- Solar.radiation_64
- WindDirection..Resultant_1 (direction the wind is blowing from given as an angle, e.g. a wind from the east is "90")
- WindDirection..Resultant_1018 (direction the wind is blowing from given as an angle, e.g. a wind from the east is "90")
- WindSpeed..Resultant_1 ("1" is site number)
- WindSpeed..Resultant_1018 ("1018" is site number)
- Ambient.Max.Temperature_(site number)
- Ambient.Min.Temperature_(site number)
- Sample.Baro.Pressure_(site number)
- Sample.Max.Baro.Pressure_(site number)
- Sample.Min.Baro.Pressure_(site number)
- (39 response variables of the form): target_(target number)_(site number)

The definition of the response variables:
1 target_8 Carbon monoxide 2 target_4 Sulfur dioxide 3 target_3 SO2 max 5-min avg 4 target_10 Nitric oxide (NO) 5 target_14 Nitrogen dioxide (NO2) 6 target_9 Oxides of nitrogen (NOx) 7 target_11 Ozone 11 target_5 PM10 Total 0-10um STP
12 target_15 OC CSN Unadjusted PM2.5 LC TOT 13 target_2 Total Nitrate PM2.5 LC 14 target_1 EC CSN PM2.5 LC TOT 15 target_7 Total Carbon PM2.5 LC TOT 16 target_8 Sulfate PM2.5 LC 17 target_4 PM2.5 Raw Data 18 target_3 Acceptable PM2.5 AQI & Speciation Mass

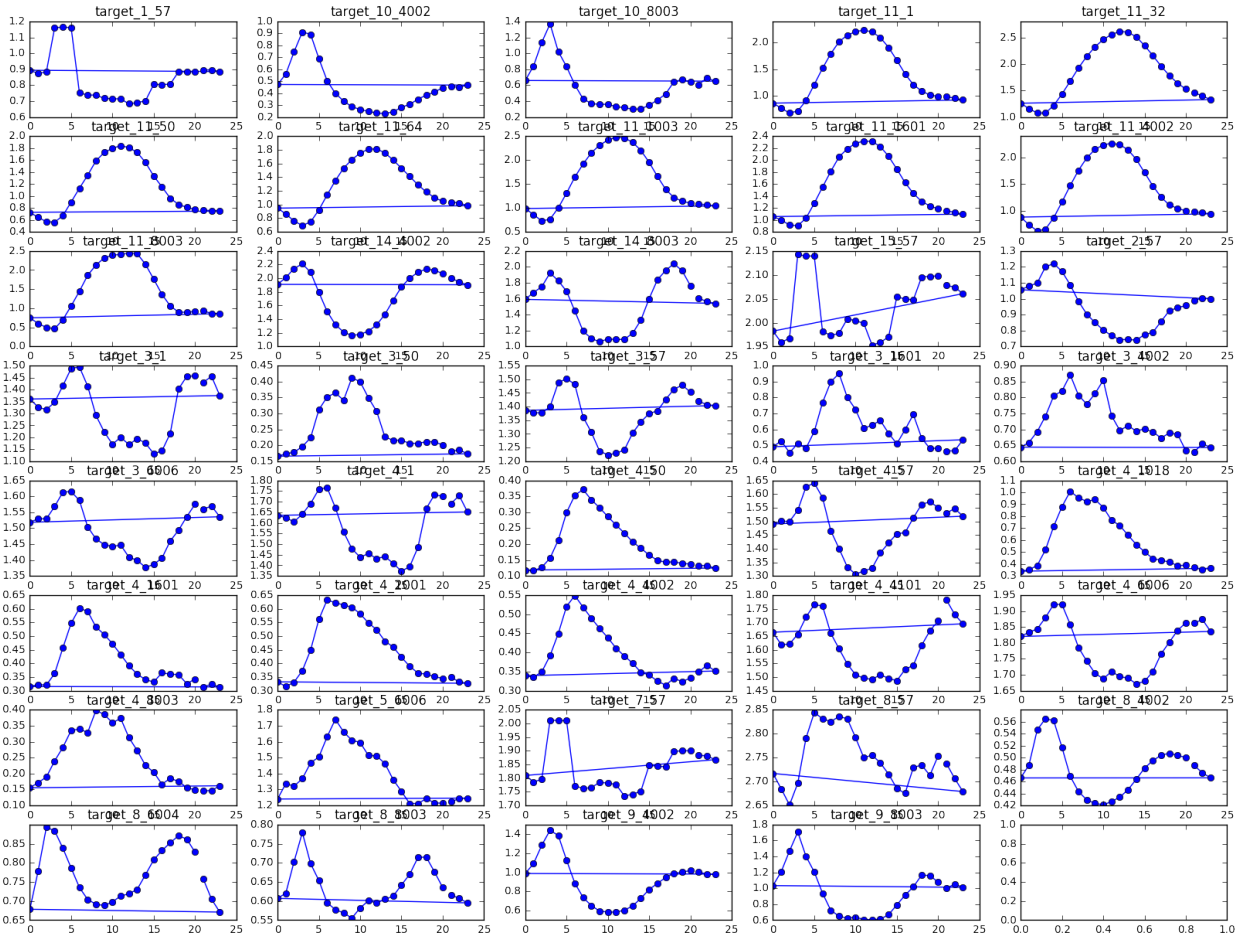
As can be seen, there are 39 output/response variables that have to be predicted. The other features are input features. However the testing dataset only contains the first 5 features, so only these features should be used in training the algorithm. It doesn't make sense to use the other input features since, they cannot be used when evaluating the algorithm. These 5 features (rowId, chunkID, position_within_chunk, month_most_common and hour) are all categorical features.

These datasets are relevant since they contain hourly data about locations and of various quantities including pollutants. With these features a model can be created that predicts the airquality for a given location and time of day. With this prediction, it can be determined if the level is dangerous or not (and thus if a warning should be triggered).

Exploratory Visualization

Since the test data contains only the first 5 features, I have decided to focus on the relation between these features and the targets. There are many observations in the data, so just plotting the feature data against the targets will probably not give a plot that gives much insight. However each feature contains less unique values, so if we group by a feature and average the targets, there will be less datapoints. For example, there are many chunks and most of them have common 'hour' values since hour ranges from 0-23.

When looking at the plot between hour and targets, there seems to be repetitive pattern (sinus function) for most targets. I would also expect this, since at certain times of the day there is more traffic and factories might produce more pollution which affects certain pollutants.



Below the plot is drawn between position_in_chunk and targets, there also seems to be some kind of relation for most targets.

I have also made these plots for the other features (see notebook), however there didn't seem to be a relation between those features and targets.

Algorithms and Techniques

As mentioned in my proposal I will use the eXtreme Gradient Boosting (XGBoost) library to create a model for my predictions. XGBoost is a very popular library that is used successfully in a lot of Kaggle competitions. The reasons why I want to use it for this project:

- it is fast, because it automatically parallels execution of the code
- it is accurate
- it can be used for a regression problem
- it has advanced features to tweak a model and get a better score

XGBoost requires that the input data is all numeric. Therefore, some preprocessing needs to be done, which I will disucss later on.

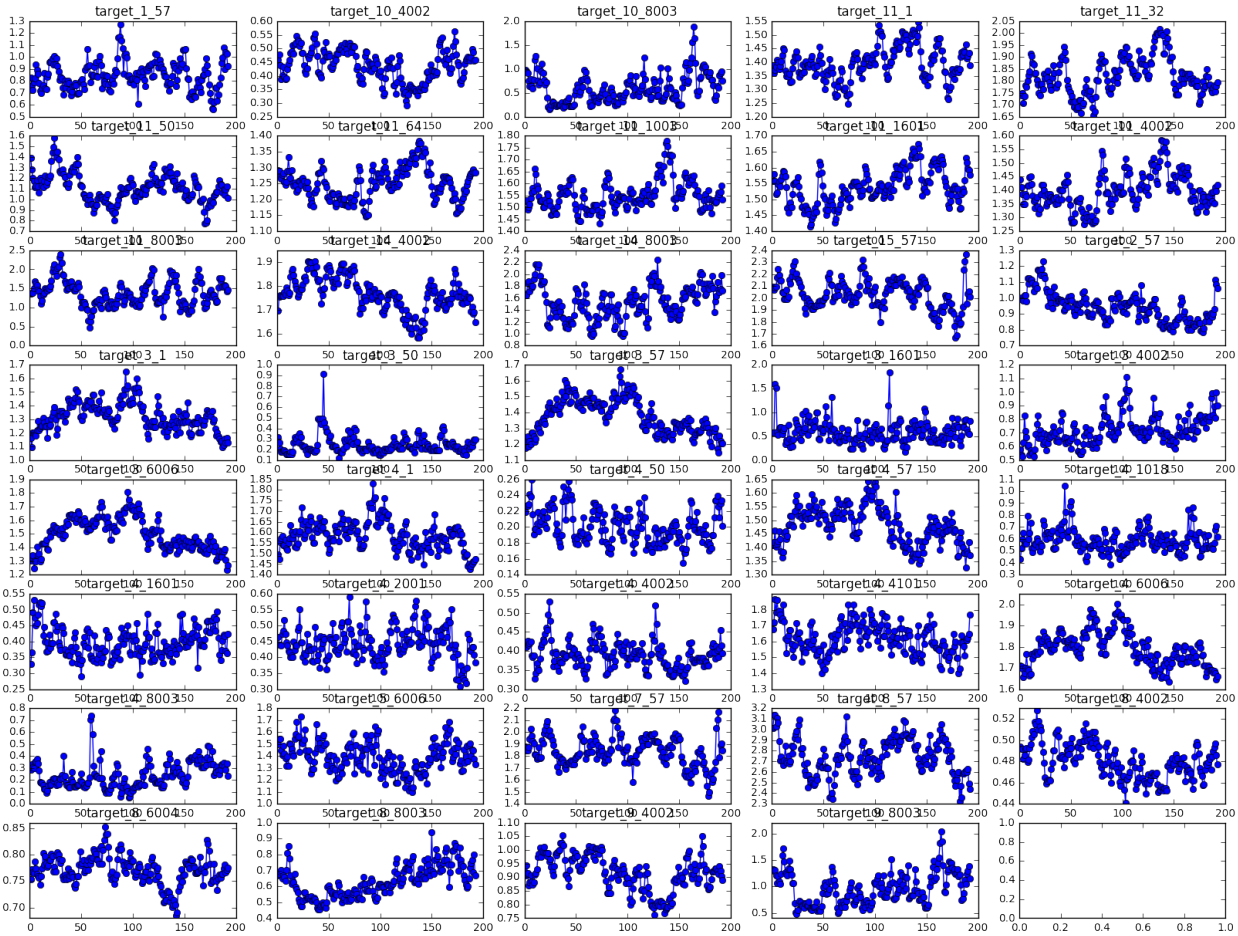


Figure 2: Average pollution per hour

Benchmark

Each kaggle competition contains a leaderbord with scores of the participants. Next to these scores, some benchamrk scores are provided. For this competition the next benchmarks and scores are provided

- predicting using average by hour over chunk (0.27532)
- predict using hourly averages (0.29362)
- SubmissionZerosExceptNAs.csv (0.53541)
- SubmissionAllZerosEvenNAsVeryBadScore.csv (517253.56661)

Because the score is the Mean Absolute Error (MAE), obviously a lower score is better. The last 2 benchmarks are clearly too simple (given their name) so it's not a serious benchmark to consider. The first one seems of a moderate complexity given it's name and it's position on the leaderboard. Therefore I will use this as my reference model.

III. Methodology

Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section: - *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented? - Based on the* ***Data Exploration*** *section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected? - If no preprocessing is needed, has it been made clear why?*

Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section: - *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data? - Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution? - Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section - *Has an initial solution been found and clearly reported? - Is the process of improvement clearly documented, such as what techniques were used? - Are intermediate and final solutions clearly reported as the process is improved?*

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section: - *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate? - Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data? - Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results? - Can results found from the model be trusted?*

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section: - *Are the final results found stronger than the benchmark result reported earlier? - Have you thoroughly analyzed and discussed the final solution? - Is the final solution significant enough to have solved the problem?*

My final result test score is 0.2386, which is better than then the benchmark score of 0.27532.

V. Conclusion

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section: - *Have you visualized a relevant or important quality about the problem, dataset, input data, or results? - Is the visualization thoroughly analyzed and discussed? - If a plot is provided, are the axes, title, and datum clearly defined?*

Feature importances? Testing scores for each target?

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section: - *Have you thoroughly summarized the entire process you used for this project? - Were there any interesting aspects of the project? - Were there any difficult aspects of the project? - Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

It took me quite a while to find a subject for this project. Since there are so many different domains where machine learning can be applied and I have a broad interest. However, I started thinking about air pollution when I was travelling in SE Asia and noticed it myself. When I saw that people are wearing masks and I read about health issues related to the air pollution, I realized it is a big problem in certain areas. I found out there has been a Kaggle competition related to it, which was very helpful. The capstone project was quite overwhelming at first, because you have to do a lot more than in the other projects of the Nanodegree.

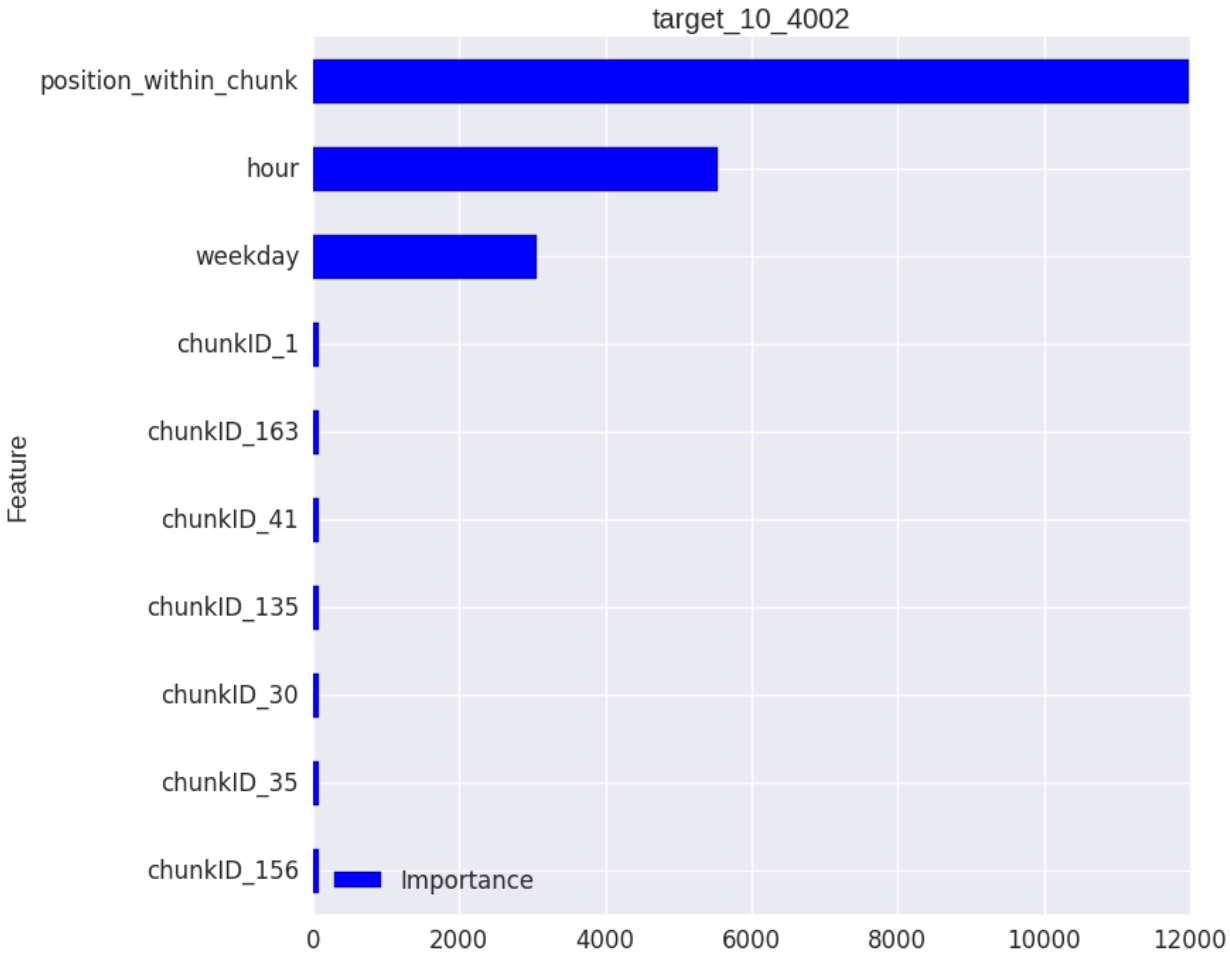


Figure 3: Average pollution per hour

Being able to use the Kaggle project was a big help, since I could use the dataset and there was some sample code to get started.

I found out that XGBoost does not support multi label classification at the moment. So, my main struggle was to sort that out.. Other struggle..testing score could not be determined because the competion was closed.

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section: - *Are there further improvements that could be made on the algorithms or techniques you used in this project? - Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how? - If you used your final solution as the new benchmark, do you think an even better solution exists?*

Improvement: use gridSearch for optimized parameters.

Airnow. 2017. "Air Quality Index." Website. https://airnow.gov/index.cfm?action=aqibasics.aqi.

Github. 2016. "XGBoost Winning Solutions." Website. https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions.

Kaggle. 2012. "EMC Data Science Global Hackathon (Air Quality Prediction)." Website. https://www.kaggle.com/c/dsg-hackathon#description.

Marilena Kampa, Elias Castanas. 2007. "Human Health Effects of Air Pollution." Journal Article. https://www.researchgate.net/publication/6192687_Human_health_effects_of_air_pollution.