# Air quality prediction

*Ger Inberg*

## Machine Learning Engineer Nanodegree - Capstone Proposal

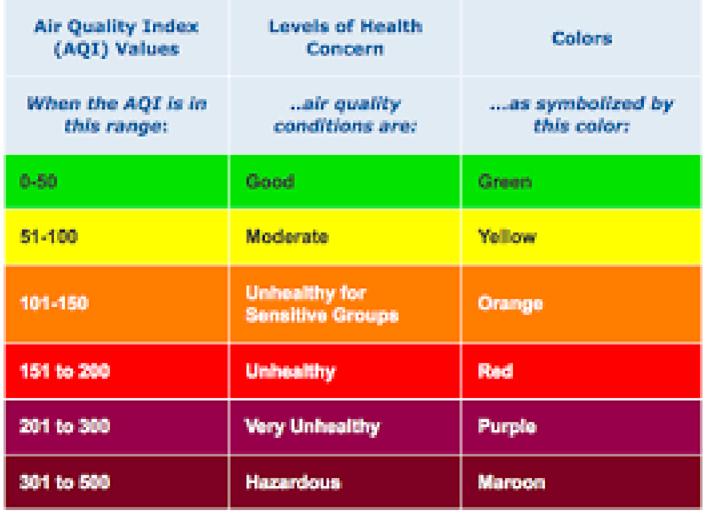
Ger Inberg April 25th, 2017

## Proposal

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.

### Domain Background

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 this 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.

### Modelling

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.

### Datasets and Inputs

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)

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).

### Solution Statement

Based upon the training data, I will build a model to predict air quality for a given location and time of day. Next, the model is applied upon the testing data, to generate the pollutant values. The result of the predictions on the testing data can be measured by the mean absolute error (MAE) across all values. The formula for the MAE is also given on [Kaggle] (https://www.kaggle.com/wiki/MeanAbsoluteError)

As mentioned before I will use the xgboost algorithm and use it as to solve the regression task. As xgboost only can process numerical values as input, some preprocessing has to be done. Since the input features that can be used are categorical features and doesn't match the numerical requirement. I will use one-hot encoding to transform the categorical features.

### Benchmark Model

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)

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.

### Evaluation Metrics

In statistics, the mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is defined by the average of the absolute differences. In R this metric can be calculated with the formula:

```r
MAE <- sum(abs(y-y_pred)) / length(y)
```

### Project Design

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

My project will consist of the next steps

- data analysis: analyse the given dataset and clean it if needed.

- design model: think of a model that can be used to predict air quality.
- implement model: implement the designed model.
- benchmark: compare my scores with the benchmark score.
- write report: state the steps I have done in a written paper. Next to describing the process and the steps, the outcomes will be discussed and compared to the benchmark model.

### References

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.