# Air Quality Prediction

Zhu An Qi

## INTRODUCTION

Air pollution is one of the great killers of our time. It occurs when the surrounding air contains gases, dust, fumes or odors in high enough quantities to be harmful to the health of humans and animals or enough to cause damage to plants and materials. With the fast development of the economy and industrial construction, air pollution has always been a major problem in China, especially in cities like Beijing. The objective of this project using big data analytic techniques to predict the hourly concentration of air pollutant level over a period of 48 hours in Beijing. We are provided two types of historical datasets: meteorological data including variables such as humidity, wind direction and speed, temperature, and atmospheric pressure; air pollutants data, including pollutants such as PM2.5, PM10, O3, NO2, CO and SO2. I will use meteorological data as the training data and the pollutants data at the labels. When predicting air quality, there are many variables to take into account, some of which are quite unpredictable. Predicting air quality, therefore, involves many difficulties, which requires extra attention to the details. Throughout this project, I will be performing the overall steps of exploring the data, preparing the data, selecting the models for predictions, comparing and testing the models, and finally coming up with the best result.

## DATA EXPLORATION

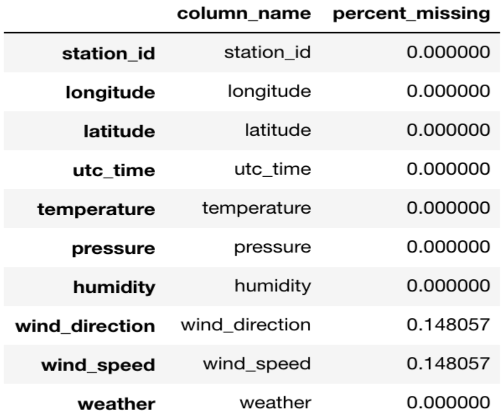
Due to the high volume of data and varied features, I began this project with data exploration to have a more structured understanding of the dataset I am working with. To effectively gain a deeper understanding of the data, I used a lot of graphs and tables to help me visualize the dataset better, and hopefully gain insights that are otherwise not available. My approach is mainly divided into three parts: missing or duplicated values detection, identification of important features and analysis of correlation of the features.

### 1. Missing or duplicated values detection

In the dataset, the first thing I noticed is the poor quality of the data including missing values, missing hours and duplicated values. The percentage of missing values for each pollutant is summarized in the table1-1 below. We can see all the pollutants contain missing values: most of them miss only miss 5% up to 6.7% of the values; PM10 reached a missing rate as high as 26%, which means over a quarter of PM10 information is unavailable. On the other hand, I found, in the same file, there are 35 missing hours and 6475 duplicated rows. Similar findings when I detected the other air quality data. Removing these outliers later will most likely improve results. The good news is, comparing with the pollutant data, the meteorological data is much cleaner, which can be shown in figure1-2.

| dataset<br>pollutant | 1701-1801 | 1802-1803 | 1804 |
|---|---|---|---|
| PM25 | 6.55 | 6.21 | 5.29 |
| PM10 | 26.77 | 26.12 | 20.95 |
| NO2 | 5.99 | 6.21 | 4.49 |
| CO | 13.76 | 6.70 | 4.48 |
| O3 | 6.56 | 6.74 | 5.11 |
| S02 | 5.96 | 6.30 | 4.40 |

*Figure1-1*



| | column_name | percent_missing |
|---|---|---|
| **station_id** | station_id | 0.000000 |
| **longitude** | longitude | 0.000000 |
| **latitude** | latitude | 0.000000 |
| **utc_time** | utc_time | 0.000000 |
| **temperature** | temperature | 0.000000 |
| **pressure** | pressure | 0.000000 |
| **humidity** | humidity | 0.000000 |
| **wind_direction** | wind_direction | 0.148057 |
| **wind_speed** | wind_speed | 0.148057 |
| **weather** | weather | 0.000000 |

*Figure1-2*

## 2. Identification of important features

To investigate how different factors affect the pollutants level, I generated different graphs to visualize the interesting patterns. For instance, I plotted a graph of PM2.5 concentration versus time to reveal the variation trend of PM2.5 throughout the year 2017. As shown in figure1-3, PM2.5 concentration in the summer is the lowest, and in the winter, is the highest. This seasonal variation reflects that there is a strong negative correlation between the PM2.5 and the temperature. I then plotted two similar graphs, but with different time periods. The first one, as shown in figure1-4, demonstrates the PM2.5 trend change over a week from the end of April to the beginning of May in 2017. From the graph, it is interesting to see the sudden trend change at the beginning of the May. My guess is that because people travel during the common holidays, which produce more gas and pollutions. Therefore, having holidays can be considered as a sign of having high level of pollutions. The second one, as shown in figure1-5, shows the PM2.5 trend over a day. Again, we noticed some interesting patterns here. The pollution level seems higher during the night than it is during the day.

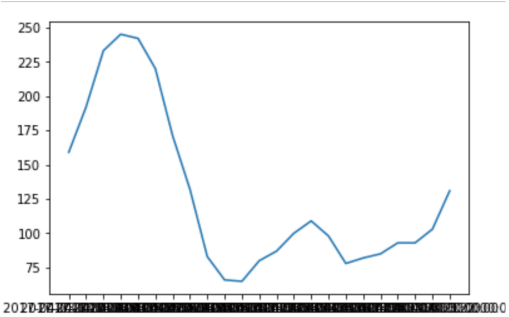


*Figure1-3*

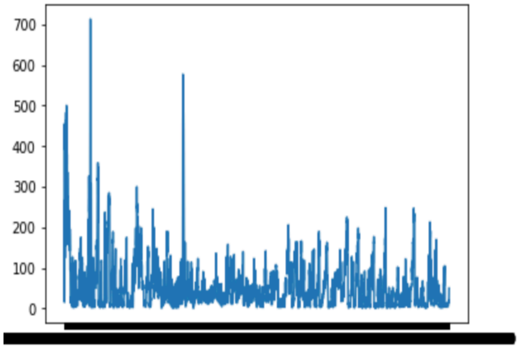


*Figure1-4*

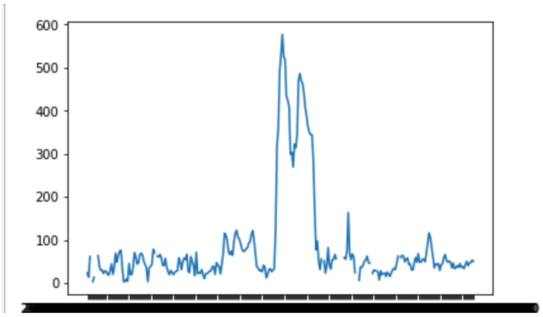


*Figure1-5*

## 3. Analysis of correlation of the features.

My last approach in exploring part is analyzing the correlations between the pollutants using the heat map. Based on the map, as shown in figure1-6, most of the pollutants are positively correlated, except O3, which is independent by itself. Of the related ones, PM10 and PM2.5 reached a correlation index of 0.85, which indicates there is a strong correlation between them. These valuable observations will need to be kept in mind when preparing the data later.

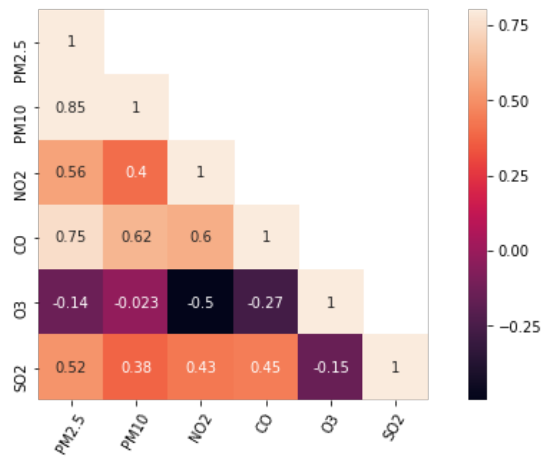


*Figure1-6*

## DATA PREPROCESSING

The quality and quantity of the data will have great influence on whether the model is good or not. Therefore, after analyzing the data details, I started preparing the data based on the observations I made previously, to improve the data quality and have them ready to achieve the best possible predictive performance.

### 1.Selection of data

Considering the big amount of dataset and high rate of missing values, I decided to only use part of the data for training and validating to avoid too many uncertain fillings for the empty values and make the training process more efficient on my laptop. At the first, I considered to use the data from 2018 as the training dataset because they are fresher. However, since we are missing too many values in 2018, I decided to only keep the data of April for training LSTM model. Next, I decided to keep the data from April to June in 2017 for training the ensemble model. Based on the seasonal variation of the pollutants trend I observed before, these three months belong to the same season as which we need to predict so there is a big possibility that the pollutants during this time follow the similar trend. Moreover, comparing to 2018 data, they are cleaner and easier to fix. Lastly, I decided to use the last two days of April 2018 as the validation set, again, because the date is close to which we need to predict.

### 2. Data cleaning

After selecting the useful data, I started cleaning them. I firstly removed the duplicated rows and then removed the outliers such as extremely large number like 99999. Meanwhile, I inserted the missing hours.

### 3. Missing value completion

When handling the missing values, I divided them into two cases: first, PM10, which has high percentage of missing values; second, the rest variables, which have lower percentage of missing values. In regards of the second case, I tried both filling them using the data from the previous line and filling them using the mean of the rest of data in the same column. The two methods gave me similar accuracy results at the end, I thus simply kept the first one. In regards of PM10, inspired by the observation from the heat map, I decided to interpolate the missing values by the strong relationship between pm2.5 and pm10. I trained the linear regression model using PM2.5 as the training data to predict the missing values for PM10. To test the accuracy of my prediction, I used the K-fold cross validation and received a RMSE of 29, which is not perfect but acceptable. I also thought about filling missing PM10 using available data from other closed stations. However, due to the time, I did not have the chance to try.

### 4. Data matching

Since multiple datasets and time series are involved, another challenge is how to match the air quality station dataset with the weather station dataset. We know that the closer the distances are between the stations, the more relevant their data are. Hence, I calculated the distance between each station using Euclidean formula, as shown in figure2-1. Then, I paired air pollutant station data with its closest weather station data on the basis of time to obtain the required data format for applying the machine learning methods later.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

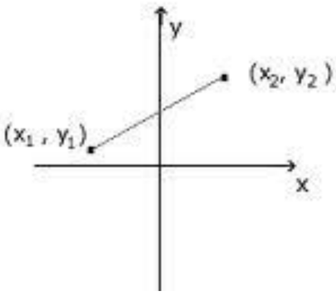


*Figure2-1*

### 5. Data transformation: unify the unit of measure

Another detail that need to pay attention is that the units of the feature are not unified in our dataset. For example, I caught, in one of the dataset, the unit of pressure is in kph but it is in m/s in the other files. I therefore converted it by multiplying the original values by 5/18.

### 6. Creating new features

In the dataset, although it seems we are already given enough meteorological features, there is still some potentially important features that are not provided but can be interpolated. For instance, as mentioned in the exploration part, I noticed the interesting relationship between the pollutions and the time. The pollutant levels vary according to different hour, day or month. In this case, I believe it is necessary to use time as features to improve the predictive quality of the models. To do so, I split up the given time into three parts including month, day and time and encoded them.

## MODEL SELECTION

In this section, I describe the two models that I applied for predicting the ambient concentration of air pollutant.

### 1.XGBoost Model

Model Details:
XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way, as shown in figure3-1.

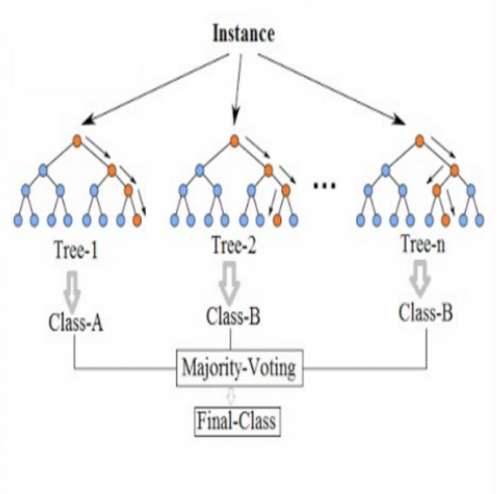


*Figure3-1*

Hyper-parameters:
I left most of the parameters as the default values. I set the n_estimators equal 300 and the max_depth equals 5.

Predictive results:
As mentioned before, I used the data from April to June in 2017 to train the model. Then I used two standards to test my results. I firstly used the K fold validation to test. They are summarized as the table in figure3-2 below.

| Pollutants | RMSLE |
|---|---|
| PM25 | 15.15 |
| PM10 | 27.97 |
| O3 | 17.82 |

*Figure3-2*

Then, I used the last two days in April 2018 as the validation set and test the result using SMAPE. They are summarized as the table in figure3-3 below. And the visualizations of PM2.5, O3 and PM10 are shown in figure3-4, 3-5 and 3-6, respectively.

| Pollutants | SMAPE |
|---|---|
| PM25 | 0.98 |
| PM10 | 0.52 |
| O3 | 0.72 |

*Figure3-3*

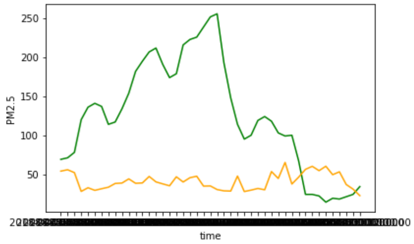


*Figure3-4*

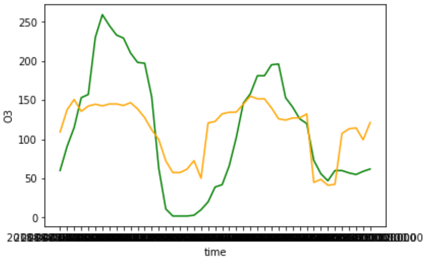


*Figure3-5*

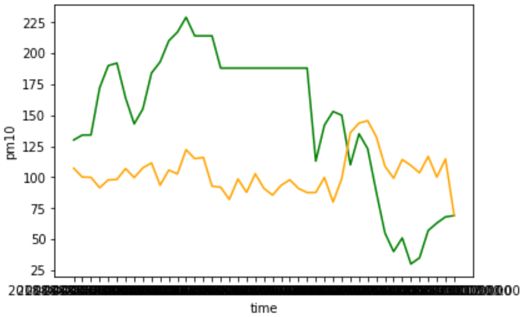


*Figure3-6*

From the result, we can see the performance is not consistent. The prediction of pm2.5 is under fitting.

## 2. LSTM Model

Model Details:
My next attempt is using Long Short-Term Memory (LSTM) networks, which can be directly imported from Keras package, to predict the data. LSTM are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. Since time series is involved in this project so the time must be continuous, I decided to use the data from April 2018 as the training set to predict pm2.5. This model works in a way that using the previous 48 hours to predict the air quality of the next hour.

Hyper-parameters:
Again, in this model, I left most of the parameters as the default values.

Predictive results:
It is surprised to see that the result I got from this model for predicting PM2.5 has a very low smape value, 0.03, which seems over fitted. The plotted graphs can be seen in figure 3-7 below. Due to the time, I did not have the chance to predict the other two pollutant.

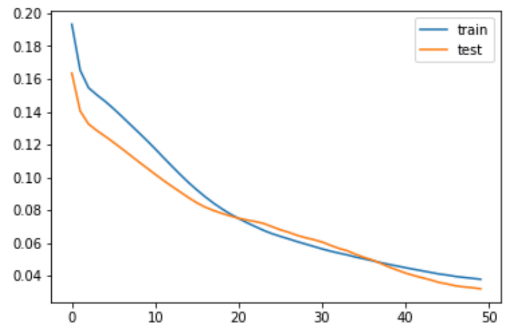


*Figure3-7*

**CONCULSION**

In conclusion, comparing the performance of the two models, I finally decided to use the XGboost to output the final results.
In the future, to improve the accuracy of my prediction, I would like to compare the performance of more models such as Lightgbm and other deep learning models, and use stacking to vote for the best performed one. In terms of feature engineering, as mentioned before, I am willing to try interpolating the missing values based on the geographical information. Moreover, tuning the parameters would likely to improve the performance of the models as well.

**Reference:**

https://xgboost.readthedocs.io/en/latest/tutorials/model.htm

https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

https://www.researchgate.net/publication/323012727_Air_Quality_Prediction_Big_Data_and_Machine_Learning_Approaches