MSBD 5002 Final Project Written Report

Beijing Pollutant Concentration Levels Prediction

Group 53

Qingshuang PANG 20527602

Brian LAM 20576299

Abstract

Air Quality is not only crucial for the healthy quality of living, accurate forecasting on air quality index is also are very important for some group, such as people with respiratory disease. In this project, we are making an hourly prediction on pollutant concentration level for the next 48 hours in Beijing, China. Our goal is to develop an optimized model that utilized the combination of time series features and weather features to make such forecast. To explicit the accuracy of the model, we train the embedded features and achieved an robust result with ***LightGBM*** and compare the performance to the baseline model of ***XGBoost***.

1. Introduction

According to the World Health Organization, outdoor air pollution is one of the causes of respiratory disease and other severe health impacts, which increase the risk of premature mortality. The health concerns of the poor air quality have risen in many developing countries, including China. The air quality index of China is the metric used by Chinese government agencies to communicate to the public how polluted the air currently is or how polluted it will be by forecasting. The AQI level is based on the level of six atmospheric pollutants, $PM_{2.5}$, PM_{10}, Ozone(O_3), Sulfur Dioxide(SO_2), nitrogen dioxide(NO_2), and carbon monoxide(CO), measured at the monitoring stations throughout the city.

The infamous air quality in Beijing is the biggest health concern for the local residents and the intermittent smog and sandstorm may cause the AQI soared to a hazardous level. An accurate air quality forecasting can provide warnings to the public when air pollution level is in harmful levels and the government agencies can use the information to estimate the economic impact by the hazard air quality and establish regulation accordingly to achieve an optimized balance of economic growth and health-related quality of life

The goal of this project is to predict 48 hours of the 3 atmospheric pollutants concentration level of the 35 air quality monitoring station, $PM_{2.5}$, PM_{10}, and O_3. $PM_{2.5}$ refers to the fine particulate matter with aerodynamic smaller than 2.5 μm in diameter, which is the most problematic out of the 6 pollutants because they are small and light and tend to stay longer in the air than heavier particles. O_3 is the ground-level ozone which is the result of chemical reaction between NO_2, volatile organic compound (VOC) and the presence of sunlight. O_3 usually reach an unhealthy level on hot sunny days, low humidity and low wind condition.

2. Objective

- To understand the relationship between $PM_{2.5}$, PM_{10}, and O_3 and the weather elements, such as temperature, pressure, humidity, wind direction and wind speed.
- To predict the concentration level of $PM_{2.5}$, PM_{10}, and O_3 between May 1st, 2018 12:00 am to May 2nd, 2018 11:00 pm for 35 Air Quality station in Beijing, China.

3. Datasets and Features

3.1 Datasets

The project consists of 4 datasets, 3 datasets are provided by the faculty of the Hong Kong University of Science and Technology and we acquired an extra dataset from the Beijing Municipal Environmental Monitoring Center:

- The historical hourly data of the 6 atmospheric pollutants concentration level, $PM_{2.5}$, PM_{10}, O_3, SO_2, NO_2, and CO from 35 air quality stations.
- The historical hourly observed weather data by instruments from the 18 observatories including weather, temperature, pressure, humidity, wind direction, and wind speed.
- The historical hourly weather data by the combination of observation data, satellite image and other observed meteorology data from the intersection of latitude (39° to 41°) and longitude (115° to 118°) in minutes, including weather, temperature, pressure, humidity, wind direction, and wind speed.
- The historical hourly data for the air quality index from 35 air quality stations

Please note that all the pollutant concentration level data started from Jan 1st, 2017 22:00 to Apr 30th, 2018 23:00, while observed weather data start from late 01/2017 to 02/05/2018.

3.2 Features

3.2.1 Air Quality Station Features:

Station_id: name of the 35 AQ station
Time: All time is in UTC time
$PM_{2.5}$: Atmospheric particulate matter smaller less than 2.5 μm in diameter.
PM_{10}: Atmospheric particulate matter smaller less than 10 μm in diameter.
NO_2: Nitrogen dioxide, gaseous air pollutants from road traffic.
SO_2: Sulfur Dioxide, from burning coal or from copper smelting
O_3: Ground-level ozone, from the chemical reaction of NO_2 and reactive substance
CO: Carbon monoxide
AQI: Air pollution level consists of the 6 pollutants above; index range from 1 to 151

Data Overview:

time*:** Although the name of the name column for timestamp is mixed with ***utc time and ***time***, we noticed the highest daily temperature is offset by 8 hours. Therefore we can infer that all times are in utc time. We may add 8 hours to adjust to local time.

***missing data*:** We noticed that there are a total of 22,890 missing timestamps, the percentage of the total missing value for the 6 pollutants are range from 11.21% to 30.48%, where PM_{10} contain the most missing values.

3.2.2 Observatory Station Features

Station_id: name of 18 observatory
Longitude: longitude location
Latitude: latitude location
Utc_time: times in UTC time
temperature: temperature in Celsius
Pressure: pressure in Pascal
Humidity: relative humidity in %
Wind_direction:[0 to 360], 999017
wind_speed: wind speed in m/s
Weather: 15 types of weathers (cloudy, dust, fog, etc…)

Data Overview:

missing data: There are sporadic missing data and we noticed some numbers are invalid and recorded as 999999.

time: Although the name of the time column is mixed with ***utc time*** and ***time***, we noticed the highest daily temperature is offset by 8 hours. Therefore we can infer that all times are in ***utc time***. We may add 8 hours to adjust to local time

***weather*:** weather may provide information to predict the pollutant, such as hot sunny weather has a higher ozone level and sand day should have a higher Air Quality Index.

3.2.3 Latitude Longitude based Features

Grid_name: name of 651 grid intersection.
Longitude: longitude location for the grid intersection (39° - 41° in min)
Latitude: latitude location for the grid intersection (115° - 118° in ‘)
Utc_time: All in UTC time
Temperature: temperature in Celsius
Pressure: pressure in Pascal
Humidity: relative humidity in %
Wind_direction: [0 to 360]
Wind_speed: wind speed in km/h
Weather 9 types of weathers, such as haze, rain, wind, etc

Data Overview:

time: Although the name of the time column is mixed with ***utc time*** and ***time***, we noticed the highest daily temperature is offset by 8 hours. Therefore we can infer that all times are in ***utc time***. We may add 8 hours to adjust the time to local.
***wind speed*:** the wind speed is in km per hour, and we may convert to m/s to keep it consistent with observed data.
weather: the first year of the weather is missing. r

4. Feature Engineering

The air quality station does not contain any weather feature and the two sets of weather features are not collected directly from the air quality, Therefore we generate the weather feature for the air quality station with the following approaches:

- To acquire the weather features from observatory station, we get the two neares stations and take a weighted average. If a null value exists, we take the weather features from the other station. If both stations are null, we consider the weather feature of the air quality station as null. We assume that the weather features of the air quality station are closed to the 2 nearest observatory.
- To acquire the weather features from grid intersections, we take the 4 corners from the adjacent air quality station and take a weighted average. Since there are not many missing data from the grid weather data. We can easily match the timestamp with the air quality data.

Table of Features from data

	AQ	OW	GW	Comment
time	X	X	X	UTC time
station_id	X	X	X	
longitude	X	X	X	Longitude of the location
latitude	X	X	X	Latitude of the location
$PM_{2.5}$	X			Fine particular matter smaller than 2.5 μm
PM_{10}	X			Fine particular matter smaller than 10 μm
NO_2	X			pollution from road traffic
SO_2	X			from burning coal
O_3	X			from the chemical reaction of NO2
CO	X			Carbon monoxide
AQI	X			Index from 1 to 510
temper.		X	X	
pressure		X	X	
humidity		X	X	
wind_dir		X	X	
wind_sp		X	X	
weather		X	X	Categorical value

- We do not need the weather feature from both observatory and grid, therefore we use the average of the 2 sources. If the observatory data is missing, we obtain from the grid data. If both are missing, then return null.
- We dropped the wind direction feature because we assume the direction is not related to any of the concentration indexes.
- We group the similar weather condition into the same category, such as cloudy day and cloudy night and we did a label encoding to change it into integers.
- We convert the utc to local time and convert km per hour to meters per second. and for some of Air Quality Station, the nearest observatory station is almost 30 km away.

5. Exploratory Data Analysis (EDA)

In order to get a understand of the relationship between different pollutions and weather, we did some exploratory data analysis.

Some preliminary data description gave us insight into the concentration index, and we can make some assumption based on the facts. The followings are some of our premises:

- Air Quality Index is a measure based on the 6 pollutants in the air, the index should have a high correlation to all of the pollutant concentration level.
- $PM_{2.5}$ and PM_{10} should be correlated because the index is measured by the size of the particle.
- NO_2 is the air pollutants from road traffic, we should expect a seasonality pattern in peak vs. off-peak hours and weekdays vs. weekends, Studies found motor vehicle exhaust fumes produce 70% of NO_2
- O_3 is Ground-level ozone, which is the main photochemical oxidants, the chemical reaction is formed when sunlight hit on NO_2 and the reactive organic substances in the air. Therefore we can

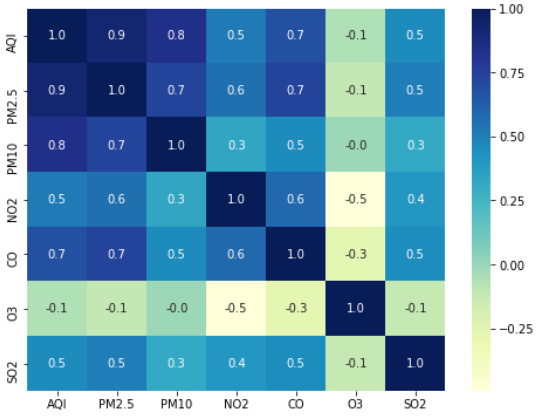


Fig. 5.1 Heatmap of the correlation between the features, AQI has a very high correlation to $PM_{2.5}$ (r = 0.9) and PM_{10} (r =0.8)

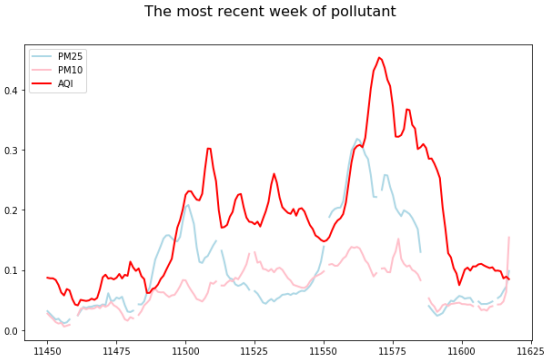


Fig 5.2 $PM_{2.5}$, PM_{10}, and AQI followed a very similar pattern of movement.

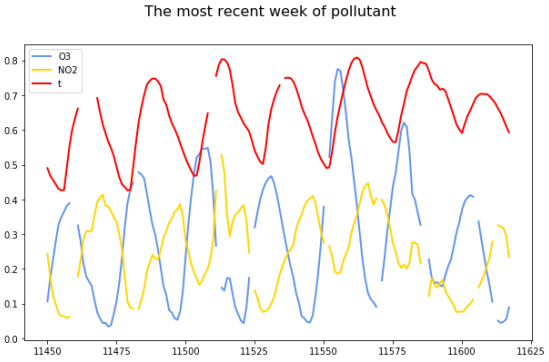


Fig 5.3 The correction of O_3 & NO_2 is -0.5; as a result, we are not surprised that the movement of the two concentration levels is moving in the opposite direction. From the graph above, temperature and NO_2 follow the same cyclical pattern

infer there is some correlation between NO_2 and O_3. Studies found the motor vehicle exhaust fumes produce 50% of O_3.

- Both sets of weather features are not directly collected from the Air Quality Station. For some Air Quality Station, the nearest observatory station is within 1 km.

6. Data Processing

We noticed the missing and error values as we discussed in part III, due to human errors or incorrect data entry. If we try to delete all the records contain missing data, the timestamp maybe mismatched - probably the worth that can be happened in Time Series Model. Our approach to the missing values is:

- First, we separate the dataset by station and fill the null value by calling pandas function, ***fillna*** with the method ***ffill*** (forward fill) and ***bfill*** (backward fill) where the gap is within 3 hours.
- Second, we separate the data by time. In the same time, if there are missing value, fill them with the average of the other stations.

After the two steps, the only missing values in the dataset are that for a certain time, all station's data is missing.

We also extract features in this step, besides generating features from weather stations.

- Day time feature. Noticing that the air pollution in a day has a periodic property, we add the time in a day as a feature.
- Statistical feature. The mean of the past 48h air pollutions. In the test data set, we use the mean in April 29th, 30th to estimate it.
- Air pollution feature: $PM_{2.5}$ and PM_{10} show a strong correlation, so we first predict $PM_{2.5}$, then use it as a feature to predict PM_{10}.

Table of missing data

	Before Data Processing		After Data Processing	
AQI	46744	11.5%	12495	3.07%
PM2.5	58858	14.47%	21315	5.24%
PM10	133485	32.8%	37860	9.31%
NO2	56926	11.22%	21315	5.24%
CO	81368	20%	40845	10.04%
O3	59079	14.53%	21315	5.24%
SO2	56868	11.22%	21315	5.24%
Weat.	27254	6.7%	24325	6%
Humi.	175	0%	0	0%
Press.	175	0%	0	0%
Temp.	175	0%	0	0%
Wind s	175	0%	0	0%

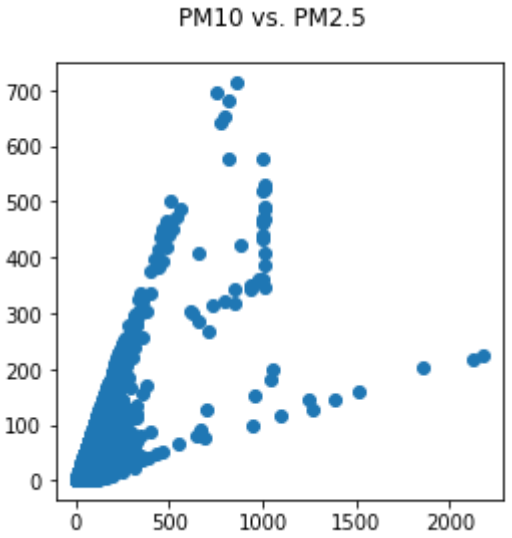


Fig 6.1 Scatter plot of the correlation of PM_{10} and $PM_{2.5}$

7. Methods

7.1 XGBoost

Our baseline model for this project is XGBoost. The loss function is defined as

$$L = \sum_{i=0}^{n} loss(y_{res},\ h(x)) + \tfrac{1}{2}\lambda \sum_{j=1}^{T} w_j{}^2 + \alpha \sum_{j=1}^{T} |wj|$$

XGBoost minimizes $L1$ and $L2$ objective function combine a loss convex loss function and a penalty term for model complexity, where $L1$ and $L2$ regularization terms above have different effects on the weights. $L2$ regularization terms encourages the weights to be small, whereas $L1$ regularization terms encourages sparsity (zero weight). Boosting is an ensembling method to generate a ***strong model*** based on ***weak predictors***. In this context, weak and strong are the measure of correlation between the model and target. The training proceeds by adding new trees that predicts the errors of prior trees, then combined with previous trees to make the final prediction, until the training data is accurately predicted by the model.

In traditional gradient boosted trees, training is inefficient because it consider **all the potential loss** and splits to create a new branch. XGBoost reduce the search by **only** consider the distribution of features in the leaf.

For this project, we add day time feature for prediction. We train all 35 air quality stations simultaneously and predict the next 48 hours concentration index for each pollutant. Because of the high correlation of $PM_{2.5}$ and PM_{10} with a high missing data for $PM_{10,}$ we use $PM_{2.5}$ as the feature to predict PM_{10} once we obtain the the estimate of $PM_{2.5}$. We obtained the best parameters by using Grid Search.

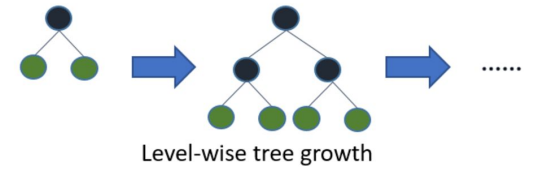
Parameters for XGBoost and LightGBM

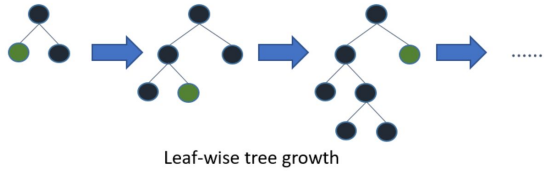
		$PM_{2.5}$	PM_{10}	O_3
XG Boost	**Column sample**	0.7	0.9	0.9
	Gamma	0.8	0.8	0.8
	Learning rate	0.01	0.01	0.01
	Max depth	7	10	10
	Min child weight	2	2	2
	N estimators	1000	1000	800
	Reg alpha	0.001	0.001	0.001
	subsample	0.8	0.8	0.8
LightGBM	**Learning Rate**	0.01	0.01	0.01
	Max Depth	10	10	10
	Num Leaves	400	450	400
	Subsamples	0.8	0.8	0.8
	Seed	3	3	3
	Num iterations	1000	900	1200

7.2 LightGBM

A major reason is that for each feature, they need to scan all the data instances to estimate the information gain of all possible split points, which is very time consuming. Light GBM is proposed recently to solve this question: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS can obtain quite accurate estimation of the information gain with a much smaller data size and EFB can effectively reduce the number of features without hurting the accuracy of split point determination by

much. LightGBM grows tree vertically while XGBoost grows trees horizontally. Usually we call the first algorithm grows tree leaf-wise and other algorithm grows level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, leaf-wise algorithm can reduce more loss than a level-wise algorithm. Because of this property, it can handle the large size of data with a faster rate of convergence and taking a lower memory to run.





The disadvantages of leaf-wise is that it may cause overfitting. Thus a cautious choice of parameter is important. We use grid search to find the best parameters.

7.3 Grid search

Grid search is a common method to find the best parameters. It use cross validation and scoring each potential combination on the validation set. Then get the average score as the final score. The model with the highest score has the best parameter we need. We use negative SMAPE as the score function. During the first trial, because of the huge amount of the training dataset, it will take about 12 minutes for every model. And for once grid search we usually need to try 800-900 fits. To reduce time consuming, we only use 0.6 of the dataset for every potential model to train. This sub-training set is pick randomly in from the whole training dataset randomly. And it reduced the time to 2-3 minutes for every models. The best parameters are shown in the table of "Parameters of XGBoost and Light GBM".

8. Results and Discussion

We use the data from 2017/1/30 22:00 as training set and the last 48 hours as validation test. However, the performance of XGBoost is not accurate as expected. We plot the predicted value and true value from the validation set, XGBoost can catch the trend of the air pollution but may not be able to catch the prediction value.

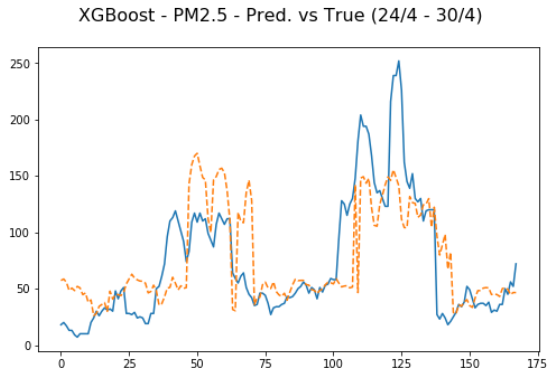
We use the symmetric mean absolute percent error (SMAPE) to evaluate our model. The SMAPE is given by

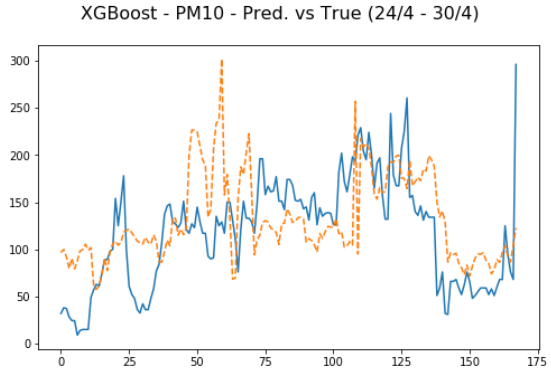
$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{F_t + A_t}$$

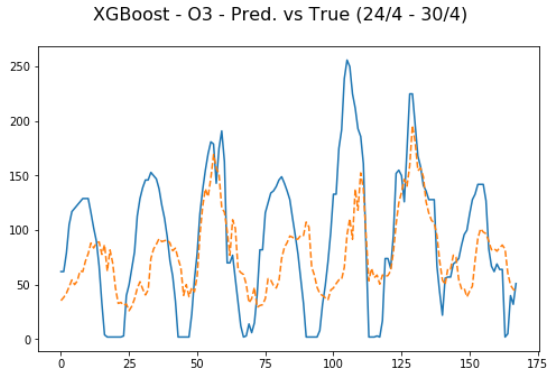
The performance of LightGBM is much accurate. The SMAPE of LightGBM is lower than XGBoost for all 3 concentration index with a better efficiency and faster computation time.

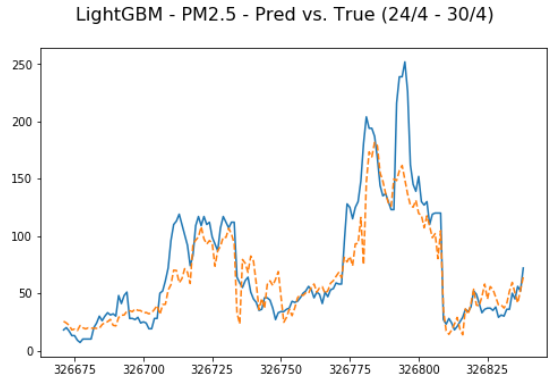
Summary of Training Error Rate

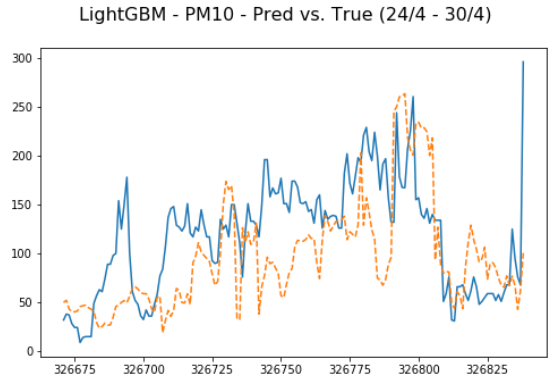
	XGBoost	LightGBM
$PM_{2.5}$	0.69760054	0.43303453
PM_{10}	0.58748142	0.40818243
O_3	0.58748143	0.56833336

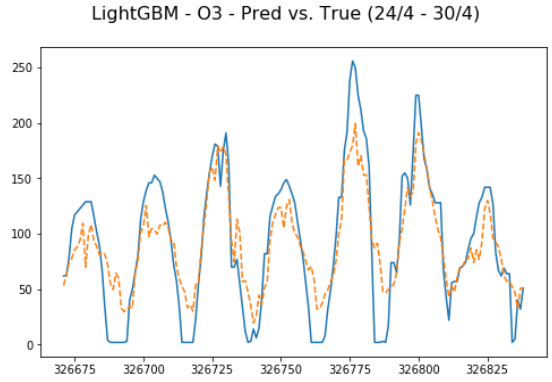












9. Conclusion and Future Work

XGBoost can catch the trend of the air pollution but is not accurate at the numerical value. LightGBM shows greater accuracy and faster speed in air pollution prediction.

In the future, we want to apply deep learning models such as LSTM because it can handle multiple output and remember the past condition, which is suitable for this problem because the air condition at this time is under a great influence of the air condition before.

10. Reference

- https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5
- https://arxiv.org/pdf/1603.02754.pdf
- https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc
- https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

11. Contribution

Brian Lam Research | Data Preprocessing | Feature Engineering | EDA | Report

QingShuang Pang Research | Methods | Data Cleansing | Report