MSBD5002 Project Report

Zhao, Cai

[redacted]

Abstract

This report is about predicting concentration levels of PM2.5, PM10, O3 between May 1 to May 2, 2018 (once an hour, 48 times for one station in total) for 35 stations in Beijing, China, using air quality data and meteorological(weather) data from January 2017 to April 30 (including), 2018. As weather data and air quality data are provided, the prediction process mainly consists of four parts: data preprocessing, feature engineering, modelling and Predicting.

1. Introduction

PM2.5 refers to atmospheric particulate matter (PM) that have a diameter of less than 2.5 micrometers, which is about 3% the diameter of a human hair [2]. PM10 describes inhalable particles, with diameters that are generally 10 micrometers and smaller [3]. O3 is the molecular formula for ozone [4].

The task is to predict concentration levels of PM2.5, PM10, O3 between May 1 to May 2, 2018 (once an hour, 48 times for one station in total) for 35 stations in Beijing, China, using air quality data and meteorological(weather) data from January 2017 to April 30 (including), 2018.

The reminder of this report is organized as follows. Section 2 describes the overview of system architecture. In Section 3, weather data selection is presented. In section 4, data preprocessing and feature engineering are performed. Data analysis on training data is put in section 5. Model selection can be seen in Section 6. The Training Details are shown in section 7. The results of prediction are demonstrated in section 8.

2. Overview of System Architecture

The system architecture is shown in Figure 1. Only grid weather data and air quality data are proprocessed by merging data separately, filtering data by station name, deleting duplicate data, filling missing data, joining data by time index for each station and merging data of all stations. Feature engineering includes original air quality features, statistics of air quality feature, station features, holiday and week features and weather features. At last LightGBM is chosen to be trained.

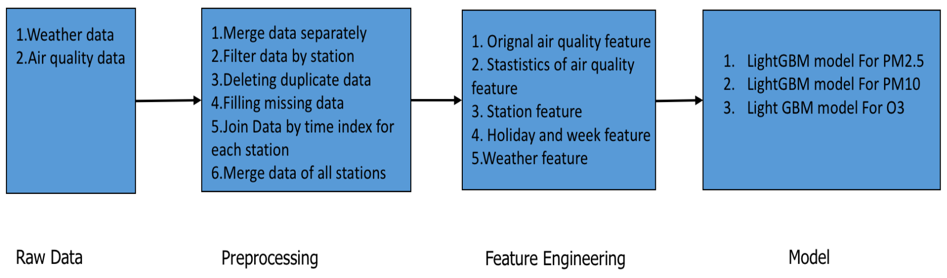

1.Weather data
2.Air quality data
1.Merge data separately
2.Filter data by station
3.Deleting duplicate data
4.Filling missing data
5.Join Data by time index for each station
6.Merge data of all stations
1. Orignal air quality feature
2. Stastistics of air quality feature
3. Station feature
4. Holiday and week feature
5.Weather feature
1. LightGBM model For PM2.5
2. LightGBM model For PM10
3. Light GBM model For O3
Raw Data
Preprocessing
Feature Engineering
Model

Figure 1: Overview of System Architecture

3. Weather Data Selection

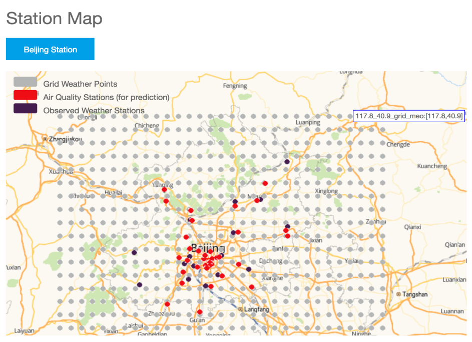
I only use the grid weather data, and do not extract any records from the observed weather data. This is because, on the one hand, the grid weather data provides enough weather information the prediction needs, on the other hand, the grid weather data is cleaner than the observed weather Data, what is more important, there are 35 air quality stations for prediction, but only 18 weather stations and some of there weather stations are not near the air quality stations (shown in Map 1). So it is more reasonable and convenient to use the grid weather data which covers all the weather data using for prediction.

So only following files about weather data are used:

gridWeather_201804.csv

gridWeather_201701-201803.csv

gridWeather_20180501-20180502.csv


Station Map
Beijing Station
Grid Weather Points
Air Quality Stations (for prediction)
Observed Weather Stations
Beijing

Map 1: Station map

4. Data Preprocessing and Feature Engineering

As grid weather data consists of three individual files, these files are merged into a file by pandas, which is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the python programming language. Each file has different column names, so column names must be uniformed. The concentration of CO, NO2 and SO2 is not considered, and are deleted from the files to reduce the complexity. After preprocessing data, for gird weather data, only features of station_id, time, temperature, pressure, humidity, wind_direction, and wind_speed are remained; for air quality data, only time, PM2.5, PM10, and O3 are kept.

After that, the air quality data for each air quality station is joined with its 1 nearest grid weather data, and generate an individual file for each air quality station. This means 35 files are generated as there are 35 air quality stations.

For each file adds the features of the station type, time_week, time_month, time_day, time_hour, holiday, work, work_first_day, rest_last_day and rest_first_day, and then fill the null value by its mean of each day for each column. After this operation, if there are still some rows which have null value, they are dropped from the file. Statistics features such max_PM25_all, mean_PM25_all, median_PM25_all, sum_PM25_all, min_PM25_all, var_PM25_all and std_PM25_all for each air quality station are added. These feature engineering are operated on both training data and testing data.

When feature engineering is done for these 35 files, they are merged again, and then saved in a new file, whose records are ordered by the time from earliest to latest time.

After data preprocessing and feature engineering, the overview information of final training data can be seen in figure 2; and the overview information of final testing data can be known from figure 3.

At this period, the feature named time is still remained, which will be removed before training the model, and the categorical feature name *station_type* is not converted into indicator variables which will be finished before the training model.

The main functions of data preproccesing and feature engineering in the file main.py are as follows:

1). prepreocessing_first()

2). generate_traning_data_csv()

3). generate_testing_data_csv()

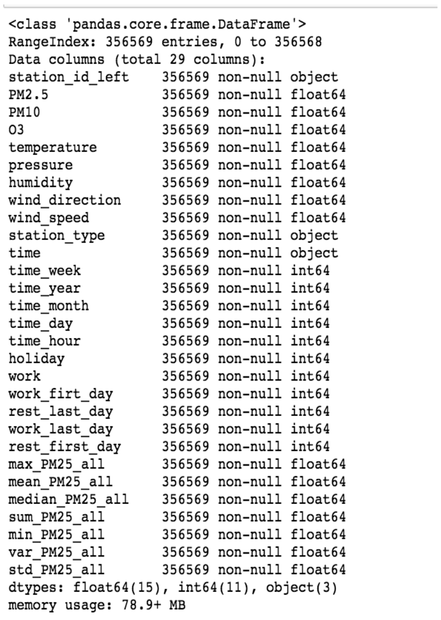
4). merge_train_data()

5). merge_test_data()

At the same time, the training data file with labels and the testing data file without labels are generated, named as follows:

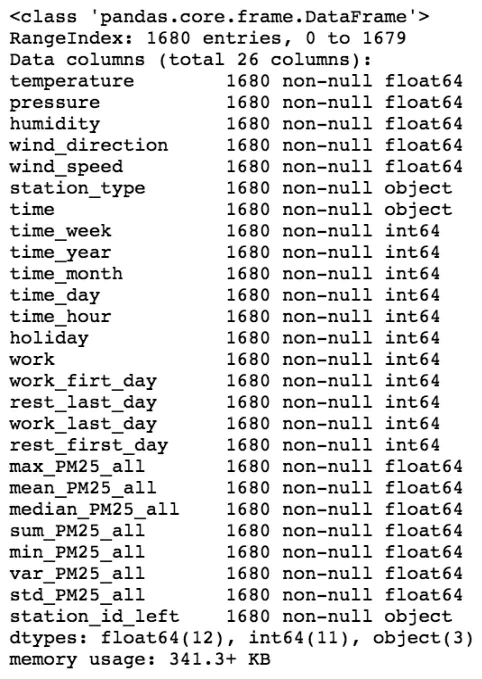
1). df_train_data_201701_201804_all.csv

2). df_test_data_20180501_20180502_all.csv



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 356569 entries, 0 to 356568
Data columns (total 29 columns):
station_id_left    356569 non-null object
PM2.5              356569 non-null float64
PM10               356569 non-null float64
O3                 356569 non-null float64
temperature        356569 non-null float64
pressure           356569 non-null float64
humidity           356569 non-null float64
wind_direction     356569 non-null float64
wind_speed         356569 non-null float64
station_type       356569 non-null object
time               356569 non-null object
time_week          356569 non-null int64
time_year          356569 non-null int64
time_month         356569 non-null int64
time_day           356569 non-null int64
time_hour          356569 non-null int64
holiday            356569 non-null int64
work               356569 non-null int64
work_firt_day      356569 non-null int64
rest_last_day      356569 non-null int64
work_last_day      356569 non-null int64
rest_first_day     356569 non-null int64
max_PM25_all       356569 non-null float64
mean_PM25_all      356569 non-null float64
median_PM25_all    356569 non-null float64
sum_PM25_all       356569 non-null float64
min_PM25_all       356569 non-null float64
var_PM25_all       356569 non-null float64
std_PM25_all       356569 non-null float64
dtypes: float64(15), int64(11), object(3)
memory usage: 78.9+ MB
```

Figure 2: information of final training data with labels

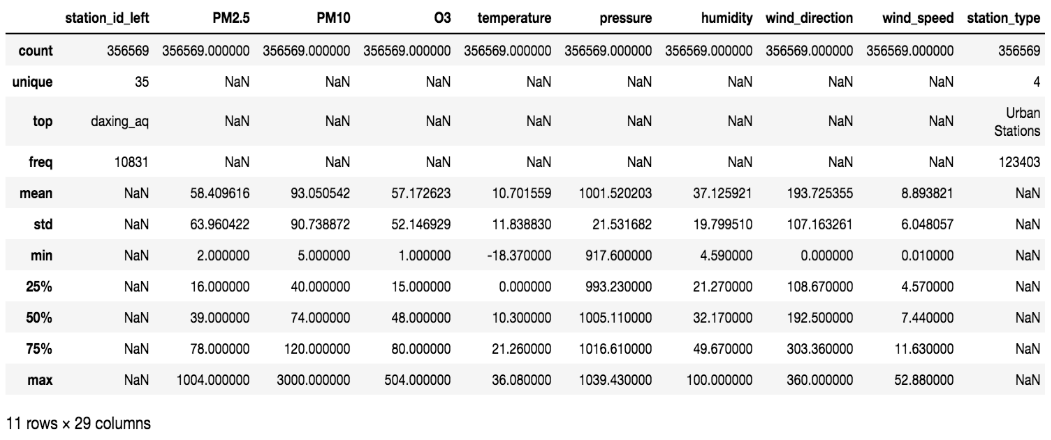


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1680 entries, 0 to 1679
Data columns (total 26 columns):
temperature        1680 non-null float64
pressure           1680 non-null float64
humidity           1680 non-null float64
wind_direction     1680 non-null float64
wind_speed         1680 non-null float64
station_type       1680 non-null object
time               1680 non-null object
time_week          1680 non-null int64
time_year          1680 non-null int64
time_month         1680 non-null int64
time_day           1680 non-null int64
time_hour          1680 non-null int64
holiday            1680 non-null int64
work               1680 non-null int64
work_firt_day      1680 non-null int64
rest_last_day      1680 non-null int64
work_last_day      1680 non-null int64
rest_first_day     1680 non-null int64
max_PM25_all       1680 non-null float64
mean_PM25_all      1680 non-null float64
median_PM25_all    1680 non-null float64
sum_PM25_all       1680 non-null float64
min_PM25_all       1680 non-null float64
var_PM25_all       1680 non-null float64
std_PM25_all       1680 non-null float64
station_id_left    1680 non-null object
dtypes: float64(12), int64(11), object(3)
memory usage: 341.3+ KB
```

Figure 3: information of final testing data without labels

5. Data Analysis On Training data

Figure 4 shows some statistics features of training file

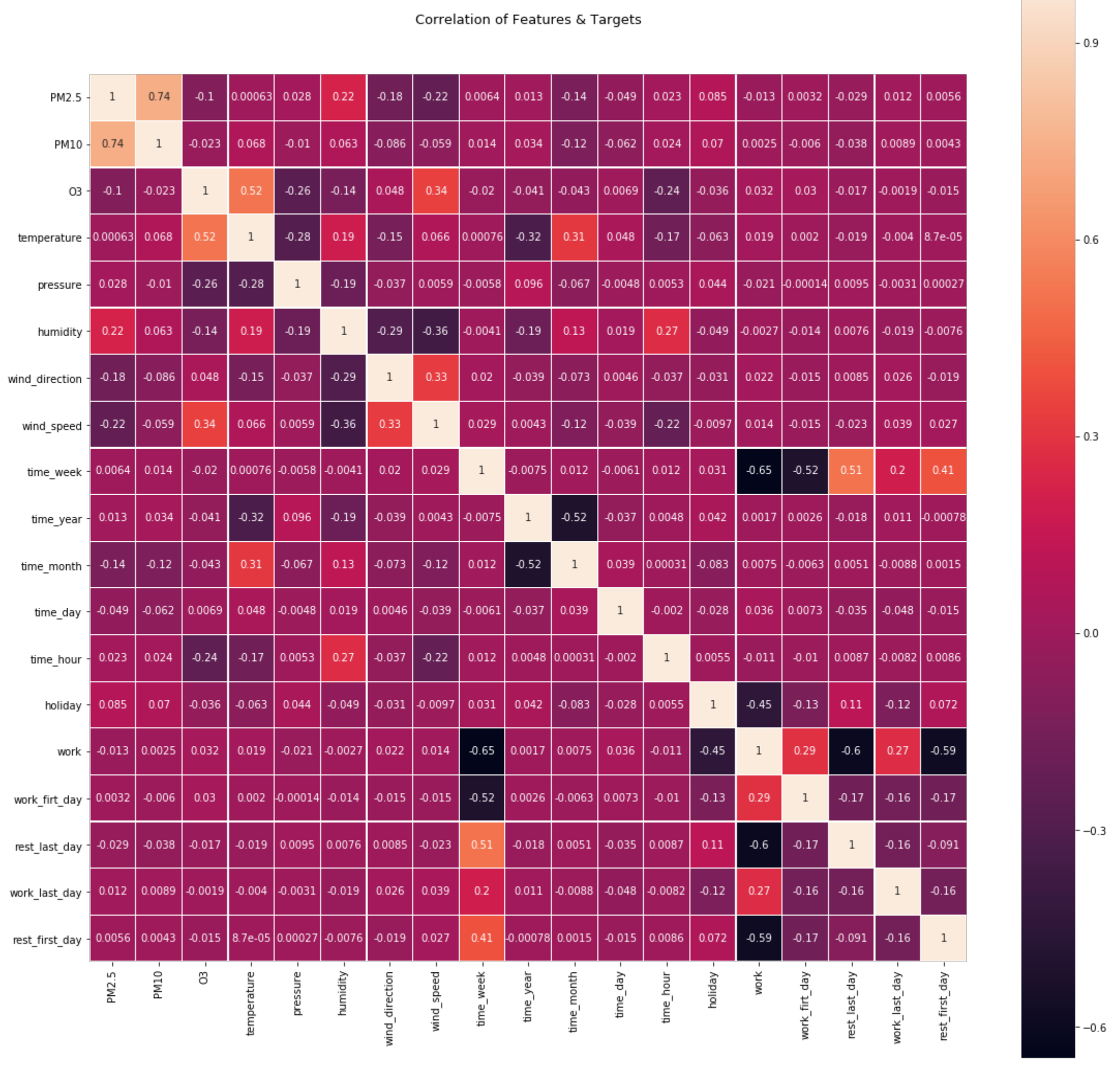


	station_id_left	PM2.5	PM10	O3	temperature	pressure	humidity	wind_direction	wind_speed	station_type
count	356569	356569.000000	356569.000000	356569.000000	356569.000000	356569.000000	356569.000000	356569.000000	356569.000000	356569
unique	35	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4
top	daxing_aq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Urban Stations
freq	10831	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	123403
mean	NaN	58.409616	93.050542	57.172623	10.701559	1001.520203	37.125921	193.725355	8.893821	NaN
std	NaN	63.960422	90.738872	52.146929	11.838830	21.531682	19.799510	107.163261	6.048057	NaN
min	NaN	2.000000	5.000000	1.000000	-18.370000	917.600000	4.590000	0.000000	0.010000	NaN
25%	NaN	16.000000	40.000000	15.000000	0.000000	993.230000	21.270000	108.670000	4.570000	NaN
50%	NaN	39.000000	74.000000	48.000000	10.300000	1005.110000	32.170000	192.500000	7.440000	NaN
75%	NaN	78.000000	120.000000	80.000000	21.260000	1016.610000	49.670000	303.360000	11.630000	NaN
max	NaN	1004.000000	3000.000000	504.000000	36.080000	1039.430000	100.000000	360.000000	52.880000	NaN

11 rows × 29 columns

Figure 4: some statistics features of training file

Figure 5 shows Correlation of some features and targets



	PM2.5	PM10	O3	temperature	pressure	humidity	wind_direction	wind_speed	time_week	time_year	time_month	time_day	time_hour	holiday	work	work_firt_day	rest_last_day	work_last_day	rest_first_day
PM2.5	1	0.74	-0.1	0.00063	0.028	0.22	-0.18	-0.22	0.0064	0.013	-0.14	-0.049	0.023	0.085	-0.013	0.0032	-0.029	0.012	0.0056
PM10	0.74	1	-0.023	0.068	-0.01	0.063	-0.086	-0.059	0.014	0.034	-0.12	-0.062	0.024	0.07	0.0025	-0.006	-0.038	0.0089	0.0043
O3	-0.1	-0.023	1	0.52	-0.26	-0.14	0.048	0.34	-0.02	-0.041	-0.043	0.0069	-0.24	-0.036	0.032	0.03	-0.017	-0.0019	-0.015
temperature	0.00063	0.068	0.52	1	-0.28	0.19	-0.15	0.066	0.00076	-0.32	0.31	0.048	-0.17	-0.063	0.019	0.002	-0.019	-0.004	8.7e-05
pressure	0.028	-0.01	-0.26	-0.28	1	-0.19	-0.037	0.0059	-0.0058	0.096	-0.067	-0.0048	0.0053	0.044	-0.021	-0.00014	0.0095	-0.0031	0.00027
humidity	0.22	0.063	-0.14	0.19	-0.19	1	-0.29	-0.36	-0.0041	-0.19	0.13	0.019	0.27	-0.049	-0.0027	-0.014	0.0076	-0.019	-0.0076
wind_direction	-0.18	-0.086	0.048	-0.15	-0.037	-0.29	1	0.33	0.02	-0.039	-0.073	0.0046	-0.037	-0.031	0.022	-0.015	0.0085	0.026	-0.019
wind_speed	-0.22	-0.059	0.34	0.066	0.0059	-0.36	0.33	1	0.029	0.0043	-0.12	-0.039	-0.22	-0.0097	0.014	-0.015	-0.023	0.039	0.027
time_week	0.0064	0.014	-0.02	0.00076	-0.0058	-0.0041	0.02	0.029	1	-0.0075	0.012	-0.0061	0.012	0.031	-0.65	-0.52	0.51	0.2	0.41
time_year	0.013	0.034	-0.041	-0.32	0.096	-0.19	-0.039	0.0043	-0.0075	1	-0.52	-0.037	0.0048	0.042	0.0017	0.0026	-0.018	0.011	-0.00078
time_month	-0.14	-0.12	-0.043	0.31	-0.067	0.13	-0.073	-0.12	0.012	-0.52	1	0.039	0.00031	-0.083	0.0075	-0.0063	0.0051	-0.0088	0.0015
time_day	-0.049	-0.062	0.0069	0.048	-0.0048	0.019	0.0046	-0.039	-0.0061	-0.037	0.039	1	-0.002	-0.028	0.036	0.0073	-0.035	-0.048	-0.015
time_hour	0.023	0.024	-0.24	-0.17	0.0053	0.27	-0.037	-0.22	0.012	0.0048	0.00031	-0.002	1	0.0055	-0.011	-0.01	0.0087	-0.0082	0.0086
holiday	0.085	0.07	-0.036	-0.063	0.044	-0.049	-0.031	-0.0097	0.031	0.042	-0.083	-0.028	0.0055	1	-0.45	-0.13	0.11	-0.12	0.072
work	-0.013	0.0025	0.032	0.019	-0.021	-0.0027	0.022	0.014	-0.65	0.0017	0.0075	0.036	-0.011	-0.45	1	0.29	-0.6	0.27	-0.59
work_firt_day	0.0032	-0.006	0.03	0.002	-0.00014	-0.014	-0.015	-0.015	-0.52	0.0026	-0.0063	0.0073	-0.01	-0.13	0.29	1	-0.17	-0.16	-0.17
rest_last_day	-0.029	-0.038	-0.017	-0.019	0.0095	0.0076	0.0085	-0.023	0.51	-0.018	0.0051	-0.035	0.0087	0.11	-0.6	-0.17	1	-0.16	-0.091
work_last_day	0.012	0.0089	-0.0019	-0.004	-0.0031	-0.019	0.026	0.039	0.2	0.011	-0.0088	-0.048	-0.0082	-0.12	0.27	-0.16	-0.16	1	-0.16
rest_first_day	0.0056	0.0043	-0.015	8.7e-05	0.00027	-0.0076	-0.019	0.027	0.41	-0.00078	0.0015	-0.015	0.0086	0.072	-0.59	-0.17	-0.091	-0.16	1

Figure 5: Correlation of some features and targets

6. Model Selection

Considering the score of predict results via Symmetric mean absolute percentage error and the training time, LightGBM [5] is chosen as the final model. LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages [1]:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel and GPU learning.
- Capable of handling large-scale data.

According to my test result, LightGBM does have faster training speed, better accuracy than other models such as random forest, adboosting, GradientBoosting and xgboosting.

7. Model Training Details

I use different validate set, different number of features and different parameters to train the LightGBM model, and select the best parameter according the score of predict results via Symmetric mean absolute percentage error. This means that at last, there are three models used to predict the concentration levels of PM2.5, PM10, O3 separately.

As the task is to predict the concentration levels of PM2.5, PM10, O3 between May 1 to May 2, 2018 (once an hour, 48 times for one station in total) for 35 stations, this means the time window is 48 hours. When training the model, I have used different time windows of data as the validation dataset, from 24 hours (1 day) to 420 hours (30 day).

At last, the model for predicting the concentration levels of PM2.5 is trained by use by the last 10 days' dataset as validation dataset and the features named *wind_direction*, and *wind_speed* are removed from both the training dataset and validation dataset.

The model for predicting the concentration levels of PM10 is trained by use by the last 2 days' dataset as validation dataset and the features named *wind_direction*, and *wind_speed* are removed from both the training dataset and validation dataset.

The model for predicting the concentration levels of O3 is trained by use by the last 1 days' dataset as validation dataset and the feature named *wind_direction* is removed, and the feature of *wind_speed* is not removed from both the training dataset and validation dataset.

More details can be seen in the source code file **main.py**, which contains all the functions of preprocessing data, training the model, and predicting.

8. Results of prediction

The results of prediction are saved in the file named submission.csv.

References

[1] https://github.com/Microsoft/LightGBM

[2] https://blissair.com/what-is-pm-2-5.htm

[3] https://www.epa.gov/air-trends/particulate-matter-pm10-trends

[4] https://en.wikipedia.org/wiki/O3

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3149-3157.