5002 Project: Air Quality Prediction

Group 48 : Huang Qiuyu   20548333

1. Introduction

This is 5002 Data Mining Project, in this project, I need to predict the 35 stations' pollution level of PM2.5, PM10, O3 from 2018-5-1 to 2018-5-2 in Beijing, China, by the air quality data and weather data from 2017-1-1 to 2018-4-30. There are total 48 hours for each station. In this project, I use Python and code on Spyder of Anaconda. And the code file is named as 5002_project.py. The result of prediction is stored in the submission.csv file.

2. Data analysis

In this project, I use grid weather data and air quality data to predict the result, as the following data file (.csv):

Table 2.1 The description of each data file

Name	Number of data	Content
airQuality_201701-201801	1155050	air quality data from 2017-1 to 2018-1
airQuality_201802-201803	247100	air quality data from 2018-2 to 2018-3
airQuality_201804	116550	air quality data in 2018-4
gridWeather_201701-201803	703470	grid weather data from 2017-1 to 2018-3
gridWeather_201804	463476	grid weather data in 2018-4
gridWeather_20180501-20180502	1680	grid weather data from 2018-5-1 to 2018-5-2
Beijing_grid_weather_station	651	Details of grid station, include longitude and latitude
Beijing_AirQuality_Stations_en	35	Details of aq_station which need to be predicted and the pollutant species

In the air quality data, it contains 'station_id', 'utc_time', 'PM2.5', 'PM10', 'NO2', 'CO', 'SO2' and 'O3', total 8 columns. 'station_id' is the aq_station name, 'utc_time' is the standard time zone, except air quality data in 201804 is use the Beijing time zone. There are total 1918700 data in air quality dataset.

In the grid weather data, it contains some weather data, include 'id', 'station_id', 'utc_time', 'weather', 'temperature', 'pressure', 'humidity', 'wind_direction' and ‘wind_speed’, among them, station means the grid station id, and grid weather data in 2018-4 is also Beijing time zone. The are total 74498182 data in grid weather dataset.

There are also 35 aq_stations with their location label and 651 grid stations.

However, there are some missing data in air quality data, I count the missing value of PM2.5, PM10 and O3 in the air quality data, and plot their proportion in the total dataset, as Figure 2.1 shows:

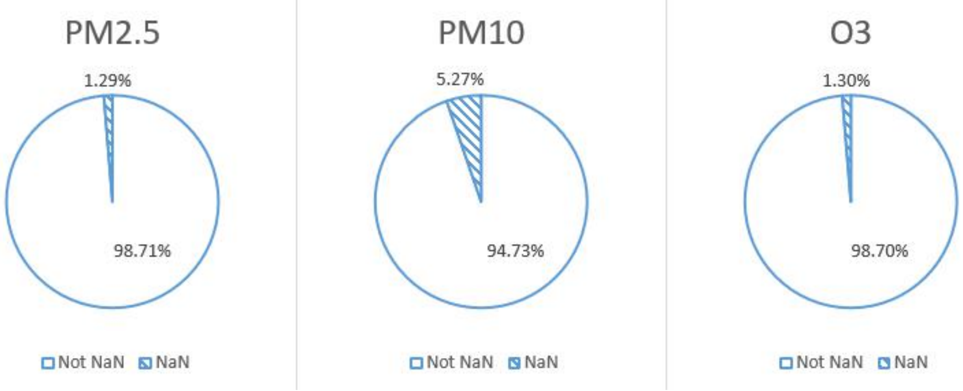

Figure 2.1 The proportion of missing values for PM2.5, PM10, O3

3. Data Preprocessing

The data we used is real-world data which is collected by testing department and scientific research department , because of acquisition way cracks, human error factors, the data cannot be used directly, I need prior to do certain preprocessing to make the data standard, even use the certain strategies to check missing data.

Through the general observation of the data, the following processing can be split to two part: data preprocessing for training data and testing data.

3.1 Data Preprocessing for Training Data

3.1.1 Import Data and Basic Handle

After import the necessary data processing lib, such as pandas and numpy, I import all the data file shows in Part 2 and do some basic process.

a. unified feature name

Some of the table with same data have different label in different .csv file. For example, PM2.5 is named ‘PM2.5’ in airQuality_201701-201801.csv and airQuality_201802-

201803, but ‘PM25_Concentration’ in airQuality_201804.csv. In that case, I rename the label and make them identical.

b. unified time type
In different data file, the expression of time has different kinds of format, some of them are ‘dd/mm/YYYY HH:MM:SS’, while others are ‘YYYY-mm-dd HH:MM:SS’. Aim to this, I have transfer them into identical datetime object.

3.1.2 Fill Missing Data

From data analysis part, we know that there are some missing value in air quality data, especially in PM10, I plot the value of PM2.5 in ‘aotizhongxin’ station from 2018-4-17 0:00 to 2018 4-7 23:00 as Figure 3.1 shows:

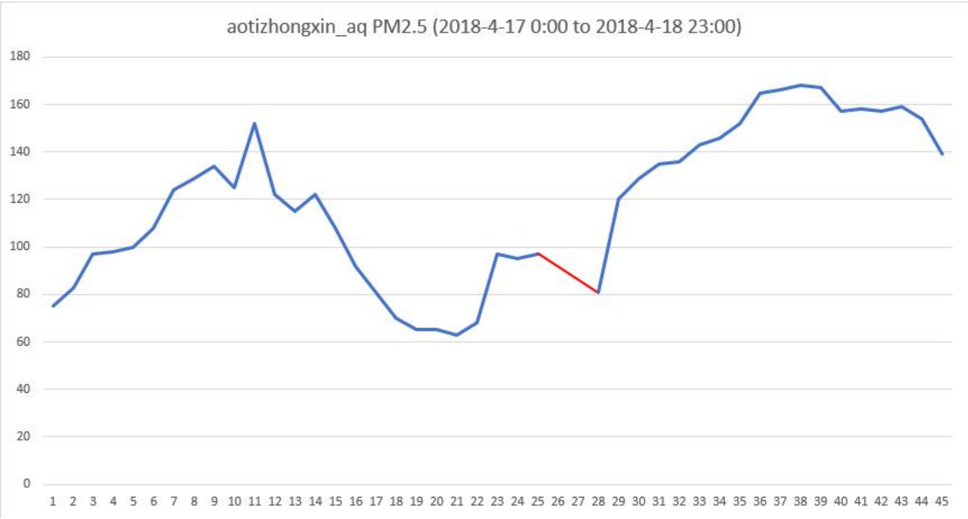

Figure 3.1: PM2.5 variation in ‘aotizhongxin’ station
from 2018-4-17 0:00 to 2018 4-7 23:00

From the Figure, we can see that, in the same space location, the value of the pollution data. At the same site, the pollutant value changes in the area are relatively continuous in time (the red line), rather than drastic and irregular changes. So we can fill in the missing values linearly. For example, if there are three consecutive missing values, and the values before and after the three values are 30 and 38 respectively, I will linearly fill the three values with 32,34,38 respectively.

3.1.3 Combine Air Quality Data with Weather Data

To predict the pollution, we need to have not only air quality data in this station, but also the corresponding weather data in this station and every hour. So, we need to find the corresponding weather data for each aq_station. And we have grid weather data and observed weather, I plot two contour maps in 2018-04-30 16:00:00 which show the temperature and wind speed of whole Beijing, as Figure 3.2 and Figure 3.3 show.

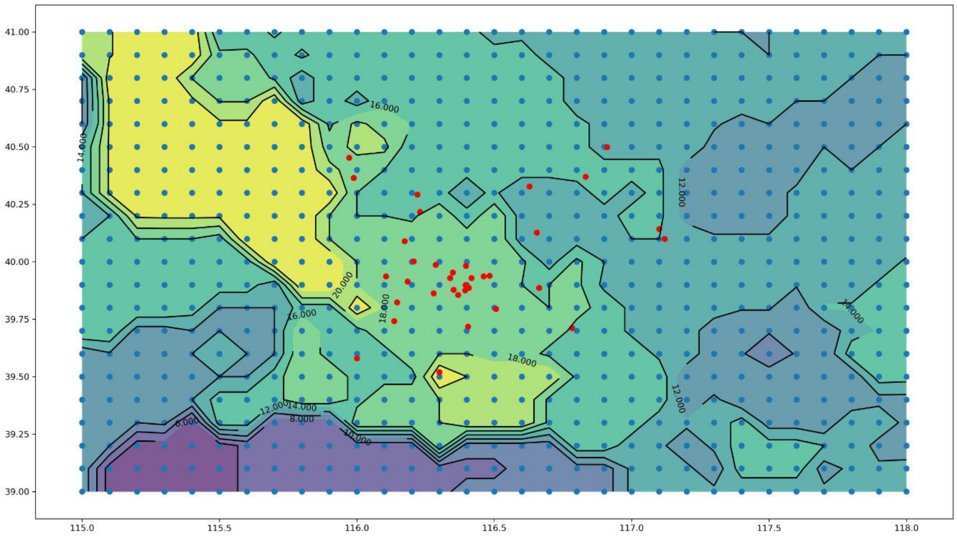

Figure 3.2: 2018-04-30 16-00-00-temperature

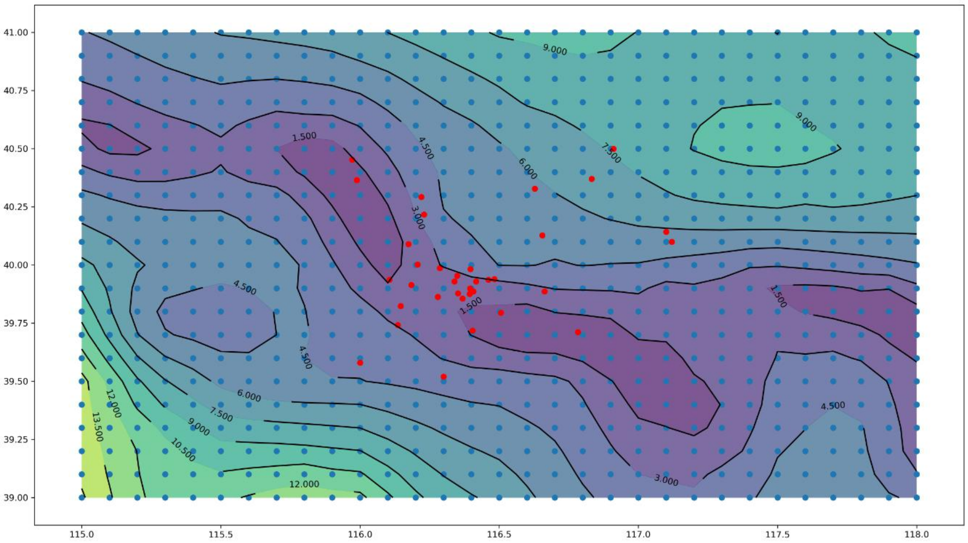

Figure 3.3: 2018-04-30 16-00-00-temperature

The two maps shows the space distribution of temperature and wind speed in some time. The red dot is the aq_station that we need predict, the blue dot is the 651 grid stations. The deeper the color, the lower the value. We can see that, in the same time, The distribution of weather data is spatially relatively continuous. So we can use the weather data of the nearest neighbor grid_station of aq_station as its weather data.

To find the nearest grid station, I use the longitude and latitude of the aq_station to calculate the index of the grid station, and add the grid station id in the table of aq_station.

To get the table which combine all the feature, we need to combine the table. At first, combine aq_station table with the air_quality table, to get the air quality data of each aq_station. After that, combine the new table with the grid weather table by grid_station id, and get the weather data of each aq_station.

3.1.4 Delete Noise

There are some duplicate grid weather data. For example, some of grid station have two data in same utc time, which may influence the accuracy of the prediction. So I delete this repetitive data.

3.2 Data Preprocessing for Testing Data

3.2.1 Basic processing

Here we read the grid weather data in 2018-5-1 to 2018-5-2, and also combine it with the table of aq_station by grid station id.

3.2.2 Fill Missing Data

Before feature engineering, we need to choose the air quality data from 2018-4-27 to 30, but there are some missing time. For example, the 'dongsi' station's air quality data in 2018-4-28 5:00 is lost, so I need to create a new table to add the whole time for each aq_station. After that, I also need to modified the time format, combine it with the initial table and fill the pollution value linearly. And filter the data with time between 2018-5-1 and 2018-5-2.

3.2.3 Sort the data

After getting the table, we need to sort it to the same as sample submission.I sort the table by the station id in station table and list by time. Here I new a table to store the data after sorting.

4. Feature Engineering

The feature mainly combined by three part: 5 weather features, 4 location features and 48 pollution data values.

4.1 Basic Weather Data

There I choose all the 5 weather features: 'temperature', 'humidity', 'pressure', ‘wind_speed’, ‘wind_direction’, which are getting from the grid weather data.

4.2 Location

From the air quality station table, we can get the location area of each aq_station, I add this attributes as location, it contains: ‘urban’, ‘suburban’, ‘others’, ‘traffic’. Because the pollution in urban and suburban may have big difference because of the population, green area, and industrial emission. And the station near traffic may also have higher pollution because of the traffic emission.

So I use this attribute as feature and one hot process it, get 4 features at last.

4.3 Pollution Value in 48 Hours

The pollution value is a continuously changing value. At a certain time, the weather and other characteristics will affect the trend and speed of the pollution value. For example, we can reasonably guess, When the wind speed is high, the pollution value will decrease rapidly. However, the change value alone cannot be predicted, and the base value should be referred to as the basis. Therefore, short-term historical data, such as the data of the first 72 hours, should be selected as its characteristics. Consider 2018-5-2 cannot use the data of 5-1 as feature, I can only choose the first 72 to 24 hour as totally 48 features. For example, data of 28.29 are taken from 5.1, and data of 29 and 30 are taken from 5.2. In the training data set, I delete the data before 2017.1.4, because they cannot get the whole 48 features. While in testing dataset, I just keep the data between 2018-5-1 and 2018-5-2.

Above all, we get 62 features at all, and delete the extra, such as ‘grid_station’ and ‘station_id’.

5. Model Construction and Optimization

5.1 Model Construction

Split data set: We split its training data into training set and validation set by 0.2 ratio in order to evaluate the performance of models.

Model choose: At first, I used the xgboost model before extract 48 hours features, but after add more features, this model need a long time to build and predict. So I change to lightGBM model to do the prediction task.

5.2 Model optimization

Model assessment: I use the smape function which was given with the data to assess the prediction accuracy, I use this to compare the prediction of validation set and the real value of validation set.

Model parameters adjustment: According to the return value of the smape function in validation set, I tried different parameters in model, such as different n_estimators and learning rate, here I choose learning rate:[0.1, 0.01, 0.001] and n_estimators: [200,400,600,800] to train the lightgbm model, and get the follow result:

Table 5.1: the accuracy of PM10 model

n_estimators\learning_rate	0.1	0.01	0.001
200	0.4697	0.5315	0.5962
400	0.4523	0.5013	0.5843
600	0.4457	0.4905	0.5745
800	0.4487	0.4839	0.5660

Table 5.2: the accuracy of PM2.5 model

n_estimators\learning_rate	0.1	0.01	0.001
200	0.5822	0.6594	0.7609
400	0.4810	0.6040	0.7446
600	0.6763	0.5874	0.7307
800	0.5530	0.5875	0.7179

Table 5.3: the accuracy of O3 model

n_estimators\learning_rate	0.1	0.01	0.001
200	0.3237	0.6032	0.7361
400	0.5152	0.5673	0.7049
600	0.5248	0.5524	0.6814
800	0.4765	0.5452	0.6629

Consider some of the result seems over fitting, I try to choose the parameters: PM2.5[600,0.01], PM10[400,0.1], O3[600, 0.01], and use this model to predict the pollution. And get the positive value of them.

6. Reference

[1] https://www.imooc.com/article/43784
[2] https://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000/001431937554888869fb52b812243dda6103214cd61d0c2000/
[3] https://blog.csdn.net/u013982164/article/details/80364500

7. Gain and Prospect

During this group project, I spent a long time to deal with it, because I am not good at coding, and the data of project is so complicated, I need time to understand it well. But in this process, I really learn a lot, the most important is the understanding of data mining and the python grammar. I also got some coding experience from this project.Although there are still many problem in my project, I think I can deal with similar problem better at next time! And thanks for this class , the professor and assistants!