PM 2.5 in Beijing Project

Report prepared for Feifang Hu

by Group5: Ziyu Huang, Yu Cao, Ying Cui, Renping Ge, Chaohui Li

Abstract

OBJECTIVE — To explore the relationship between air pollutants factors and weather factors that may affect the PM 2.5 concentration. To explore that if PM 2.5 concentration of Beijing has decreased in recent years. To fit the model so that it can predict the performance of PM 2.5 in the future.

METHODS — We used a large dataset, provided by U.S. Department of State Air Quality Monitoring Program. We used k-nearest neighbors method and medians to fill the missing values. After running data cleaning process and exploratory data analysis, we use the multiple linear regression model to fit the data. To solve the multicollinearity problem, we used the lasso regression and random forest to fit the data to get the importance of each feature and the relationship between these features and PM 2.5. Then we used the traditional ARIMA model and LSTM model to predict the future PM 2.5 values for daily data and monthly data. Finally, we used Mann-Kendall test and simple linear regression to test whether there existed a significant decreased trend in recent years.

RESULTS — For both daily data and monthly data, according to the full linear regression, we calculated the condition number and variance inflation factors, which indicated the multicollinearity. Then we fitted the lasso regression and rebuilt the linear regression model to know that p values for NO2, DEWP are all smaller than 0.05. From the random forest model, we got the importance of features. Values for CO and NO2 are larger than those for SO2 and O3. Values for DEWP and PRES are greater than TEMP and WSPM.

CONCLUSION — After investigation, we can conclude that air pollutants largely affect PM 2.5. Among them, NO2 is significant for both daily data and monthly data. Also, weather features affect the PM 2.5 and dew point temperature is significant for both daily and monthly data. For daily data, CO and NO2 are more important than SO2 and O3. Dew point temperature is also more important than other three weather features. For monthly data, NO2 and CO are also more important than SO2 and O3. Pressure and dew point temperature are more important than other two weather features. The southeast wind has a significant effect on PM 2.5. In the past years, PM 2.5 had an obvious downward trend, which meant that the air quality in Beijing became better. Besides, we can use long short term memory model to predict the future PM 2.5 values for both daily frequency and monthly frequency.

1. Introduction

1.1 Background

PM2.5 refers to atmospheric particulate matter that have a diameter of less than 2.5 micrometers, which means, this kind of matter is more than one hundred times thinner than a human hair. As a result, particles in this category cay be only seen with a microscope, and the effects of PM2.5 can be discussed from two aspects. On the one hand, PM2.5 is harmful for human health. Exposure to high concentration of PM2.5 can cause premature health, nonfatal heart attacks, irregular heartbreak, and aggravated asthma. Also, it would cause lung function decreasing, and increasing respiratory symptoms, such as coughing and difficulty breathing. On the other hand, PM2.5 also causes environmental damage. For example, making lakes and streams acidic, and changing the nutrients balance, which pit a threat an aquatic creature. In addition, this particle depletes the nutrients in soil and damages sensitive forests and farm crops. Furthermore, it contributes to acid rain effects. All in all, it will affect the diversity of ecosystems.

1.2 Project Summary

As to the PM2.5 in Beijing, our goal can be summarized as following. The first part is about to have a basic knowledge about the situation of PM2.5 in Beijing. It shows different values of PM2.5 in different seasons, different times of the day, different wind directions, peak periods, etc. The second part is to study the factors that affect PM2.5 values in Beijing. There are many factors that may affect the values of PM2.5. We will focus on the air pollutants factors and weather factors. The third part is to find suitable models and methods to predict the future PM2.5 in Beijing. Generally, we are interested in the following days or following months PM2.5. The last part is to test whether there has been a significant downward trend of PM2.5 in Beijing in recent years. In fact, both the country and the city has adopted many measures and implemented many policies to reduce PM2.5.

Based on the goal, we designed the following analysis plan.

1.3 Study Design

Firstly, we want to do the exploratory data analysis including the distribution of each variable, different PM2.5 values in different categories, for a basic overview of the dataset. Secondly, linear regression models and random forest model can be used to explore the relationship between various factors and PM2.5 and the importance of each factor. Thirdly, ARIMA model and LSTM model can be used to predict the future PM2.5 values. Finally, we can use regression model to test the significance of parameters so that we can know whether there exists a downward trend.

2. Data Structure

We collected two datasets. One included date, hour, month, day, TEMP, DEWP, WSPM, PRES, SO2, NO2, O3, CO, wind, and PM 2.5 from 2014 to 2020. The specific meaning of the variables is shown in Table1. The other one included PM 2.5 from 2010 to 2020.

Table1 Variables Summary

Variable	Meaning	Variable	Meaning
date	The date of collecting data	PRES	Pressure (hPa)
hour	The hour of collecting data	SO2	Sulfur dioxide concentration (ug/m^3)
month	The month of collecting data	NO2	Nitrogen dioxide concentration (ug/m^3)
day	The day of collecting data	O3	Ozone concentration (ug/m^3)
TEMP	Temperature (degree Celsius)	CO	Carbon monoxide (ug/m^3)
DEWP	Dew point temperature (degree Celsius)	wind	Wind direction
WSPM	Wind speed (m/s)	PM 2.5	PM 2.5 concentration (ug/m^3)

3. Data Preprocessing

In 533844 rows of records, 2390 SO2 records, 2394 NO2 records, 2394 O3 records, 3345 CO records, 4 Temp records, 22 DEWP records, 3917 PRES records, 8454 wind records and 2 WSPM records have missing values. This data set contains two type of missing value: continuous and categorical variable. Because data set contains hourly, daily, monthly, and yearly information, simply applying mean or mode of the entire data set will be inappropriate and biased. Every missing value must contain information in one short period of time. Therefore, introducing KNN imputation is a good option. Because it performs imputation of missing data in a data frame using the k-Nearest Neighbor algorithm. For categorical variables, mode will be applied. For continuous variables, the median value is instead taken.

After filling the missing data of independent variables, it is also crucial to transform the response variable. The original pm2.5 distribution is heavily skewed and contains plenty of zeros. To make sure it will fit the following models better, transformation is needed. Therefore, use transformation of log(1+Pm2.5) will solve heavy skewed problem and leave every number positive.

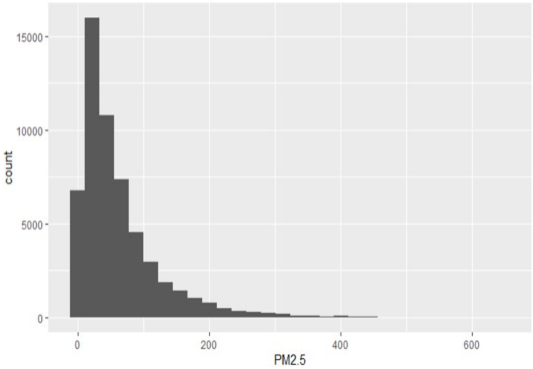

Figure3-1 Distribution of PM 2.5

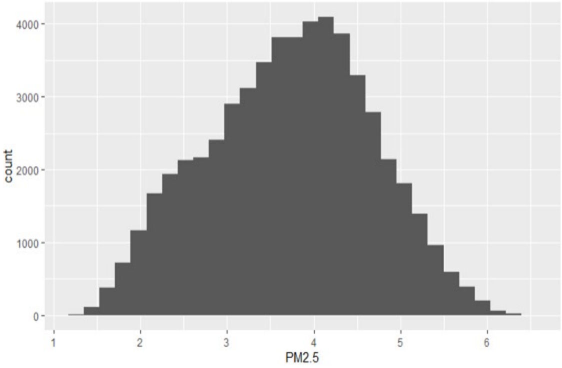

Figure3-2 Distribution of log(1+PM2.5)

In the data set contains 9 numerical independent variables. Variable Temp ranges from -14.5 to 32.579, variable DEWP ranges from -33.85 to 26.23, variable WSPM ranges from 0.9 to 9.2, variable PRES ranges from 1001 to 1041, variable SO2 ranges from 2.024 to 82.1, variable NO2 ranges from 4.75 to 147.45, variable O3 ranges from 3.29 to 175 and variable CO ranges from

0.19 to 7.72. Numerical attributes are not consistent with each other. Therefore, introducing scaling is necessary. Scaling process makes the features more consistent with each other, which allows the model to predict outputs more accurately.

This data set only have one categorical variable wind direction which contain 17 levels. They are no wind, E, W, N, S, NE, ENE, NNE, SE, ESE, SSE, NW, NNW, WNW, SW, SSW and WSW. It is clear to see that levels are too many. To avoid redundant levels in a categorical variable and to deal with rare levels, simply combining the different levels will do it. Level of NE, ENE and NNE wind direction fall in NE level, level of SE, ESE and SSE wind direction fall in SE level, level of NW,NNW and WNW wind direction fall in NW level and level of SW,SSW,WSW wind direction fall in SW level and leave other direction as it is. To better fit linear regression in the following model step, wind direction needs to transform to dummy variable.

4. Methodology

4.1 Linear Regression

Linear regression is a linear approach to fit a model that could conclude the relationship between response variable and independent variables. In this model, we assume that there exists a linear relationship between response variable and independent variables. The result of model could be used to predict numeric data and quantified the strength of the relationship between the prediction results and the predictors. The equation is like below:

$$\mathrm{y_i} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1} + \epsilon$$

4.2 Lasso Regression

Lasso regression is to add L1 regularization on the basis of standard linear regression, which could make the weights of linear regression sparser, that is, to make many coefficients in linear regression equals to 0, to help eliminate the effect of multicollinearity. Beyond that, it could do the feature selection at the same moment. The cost function is as below:

$$J = \frac{1}{n}\sum(f(x_i) - y_i)^2 + \lambda|\omega|_1$$

4.3 Random Forest

Random forest is an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classification regression of the individual trees. Forests are like the pulling together of decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree. Though not quite similar, forests give the effects of a K-fold cross validation.

4.4 ARIMA Model

ARIMA, short for ‘Auto Regressive Integrated Moving Average’, is a class of model that explains a given time series based on its own past values.

Any ‘non-seasonal’ time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models. It is characterized by three terms: p, d, q, where ‘p’ is the order of the AR term, ‘q’ is the order of the MA term and ‘d’ is the number of differencing required to make the time series stationary. ‘p’ refers to the number of lags of Y to be used as predictors. And ‘q’ refers to the number of lagged forecast errors that should go into the ARIMA Model. The equation of ARIMA model can be written as:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} +..+ \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} +..+ \phi_q \epsilon_{t-q}$$

If a time series has seasonal patterns, seasonal terms should be added.

4.5 LSTM Model

Long-short term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data. LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition, and anomaly detection in network traffic or intrusion detection systems. A RNN using LSTM units can be trained in a supervised fashion, on a set of training sequences, using an optimization algorithm, like gradient descent, combined with backpropagation through time to compute the gradients needed during the optimization process, in order to change each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) with respect to corresponding weight.

5. Data Analysis and Results

5.1 Exploratory Data Analysis

Firstly, analysis the relationship between PM2.5 and all the numerical attributes is needed.

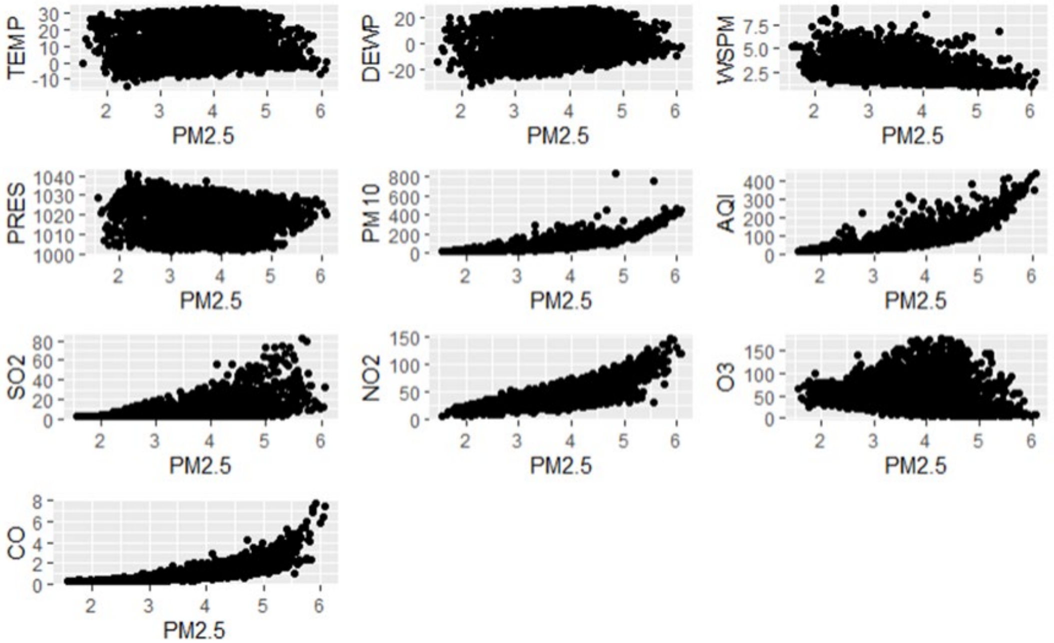

Figure5-1 Relationship between PM2.5 and numerical attributes

As we can tell from the Figure5-1, variable TEMP, DEWP, WSPM has no obvious relationship to PM2.5. Variable PRES has a negative trend as PM2.5 gets bigger. Variable PM10, AQI, SO2, NO2, CO has positive relationship to PM2.5. Variable O3 has special phenomena, PM2.5 before 4 has positive relationship between O3, and PM2.5 after 4 has negative relationship between O3.

Secondly, we find some interesting phenomena about different type of time periods.

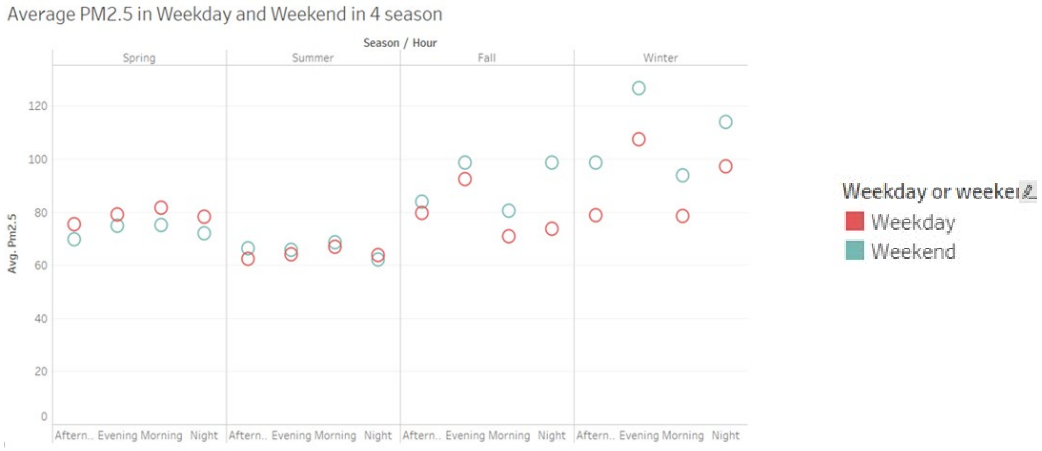

Figure5-2 Average PM2.5 in weekday and weekend in four seasons

As we can see from Figure5-2, average PM2.5 perform differently in different season. PM2.5 in spring and summer is relatively lower than PM2.5 in fall and winter. PM2.5 in evening and morning is relatively higher than PM2.5 in afternoon and night. In the graph, the red plot represents PM2.5 in weekday, the green plot represents PM2.5 in weekend. We intuitively think that Weekday's PM2.5 will be higher than that of the weekend because factories are open, everybody need transportation to go to work. However, in fall and winter, this is not the case. This phenomenon is call "weekend effect". People tend to go out and gather. They tend to have more activities in weekend than weekdays. Moreover, people tend to drive more to rural sightseeing location which increase the emission of PM2.5.

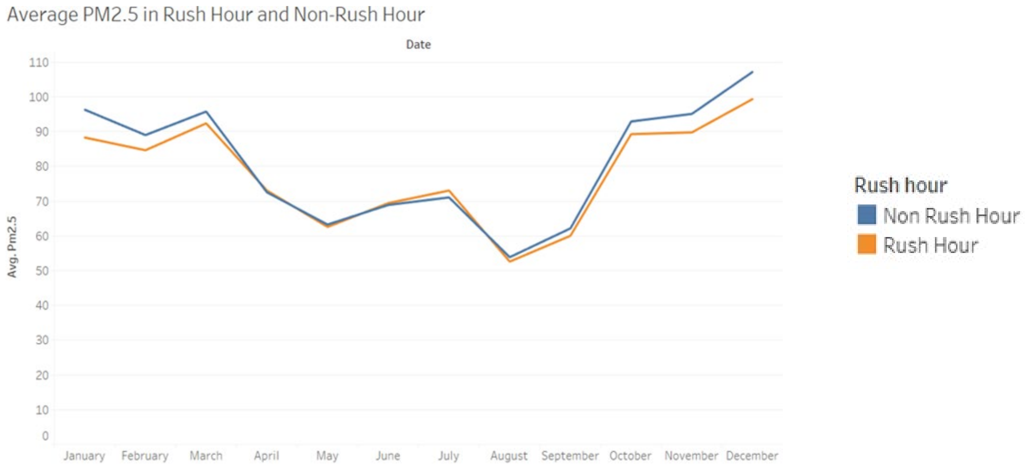

Figure5-3 Average PM2.5 in Rush hour and Non-Rush hour in a year

The trend graph from Figure5-3 shows the comparison of rush hour and non-rush hour in a year. The blue line represents the non-rush hour, red line represents the rush hour. From October to March, we can see that the average pm2.5 in non-rush hour is higher in rush hour. Temperature inversion can explain the problem. Normally, the air temperature decreases as the altitude increases. The lower air is hotter, and the upper air is colder. The cooler air sinks and the warmer air rises.

But at certain times, like autumn and winter nights, the lower atmosphere near the ground cools down because of a sharp drop in ground temperature, while the upper air cools down less quickly and gets hotter than the lower atmosphere.

Once this inversion layer is formed, the air cannot convection up and down, and it is difficult for pollutants to diffuse.

Temperature inversion generally does not occur in summer, and the concentration of PM2.5 in a day does not fluctuate as much as that in winter, and the overall concentration is lower.

5.2 Factors that affect the PM2.5

First, we considered all variables to fit the full linear regression model for daily data. The condition number was 4480, which indicated that there may be strong multicollinearity. To further detect this problem, we calculated variance inflation factor (Appendix1 Table3) and concluded that values for temperature, dew points, wind speed, pressure, NO2, O3 and CO are larger than 10, which meant the existence of multicollinearity. To solve the problem, we chose the lasso regression and random forest to analyze the factors. From lasso regression, we got the coefficients of variables (Appendix1 Table4) and found that coefficients of temperature, pressure, SO2 dropped to 0. After removing the variables with coefficients of zero, we retested the variation inflation factors of the remaining variables and found that all their values reduced to around 10 and below. Then for the new reduced model, we found that all continue variables, northeast wind and southeast wind were significant from Table5-1. To simultaneously study on the significance of each variable and solve the multicollinearity, we use the random forest model to fit the data, which can select variables and provide the importance of variables. The importance of features was shown in Figure5-4. From the importance of features (Appendix1 Table5), we concluded that among air pollutants factors, NO2 and CO were more important than SO2 and O3. Among weather factors, dew points temperature is more important. Air pollutants factors occupied the main impact.

Table5-1 The coefficients of variables in reduced model for daily data

variable	coefficients	t	variable	coefficients	t
const	2.031	0.000	wind_1.0	0.049	0.282
DEWP	0.007	0.000	wind_2.0	-0.044	0.465
WSPM	-0.040	0.000	wind_4.0	0.039	0.258
NO2	0.024	0.000	wind_5.0	0.071	0.013
O3	0.006	0.000	wind_6.0	0.081	0.009
CO	0.388	0.000	wind_7.0	0.060	0.094

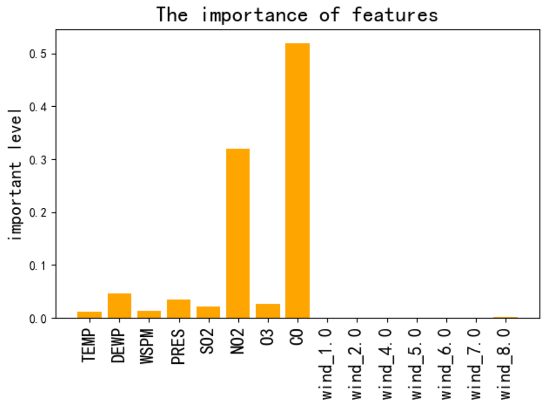

Figure5-4 The Importance of Features (Daily Data)

For monthly data, the modeling and analysis process was similar to the above-mentioned daily data. We first fitted the full linear regression model (Appendix1 Table2). According to the condition number and the significance of variables, we detected the multicollinearity by calculating the variance inflation factors (Appendix1 Table3). Then we fitted the lasso regression model and random forest model. For lasso regression model (Appendix1 Table4), we removed temperature, wind speed, pressure, and CO. To fit the new reduced model, we found that SO2, NO2, O3 and southeast wind are significant. After that, the importance of features from the random forest (Appendix1 Table5) indicated that air pollutants NO2 played the most important role in PM2.5, as shown in Figure5-5.

Table5-2 The coefficients of variables in reduced model for monthly data

variable	coefficients	t
DEWP	-0.004	0.420
SO2	0.020	0.032
NO2	0.065	0.000
O3	0.016	0.000
wind_5	0.035	0.889
wind_6	0.474	0.009
wind_7	0.196	0.246

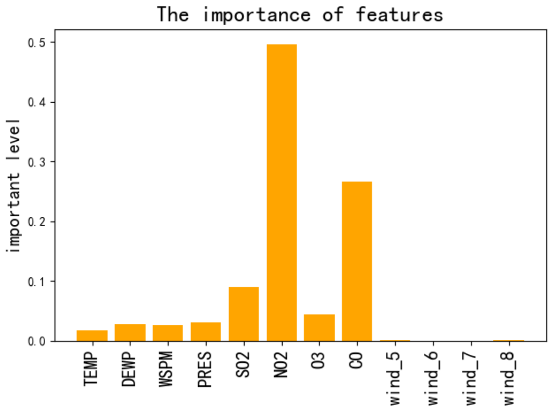

Figure5-5 The Importance of Features (Monthly Data)

5.3 Prediction on PM2.5

5.3.1 ARIMA model

ARIMA model requires time series is stationary. For daily data, to check stationarity of data, ADF (Augmented Dickey Fuller) test is used. The result is shown in Appendix1 Table6. P-value is smaller than 0.01. Null hypothesis is rejected indicating the series is stationary. To find p and q, the Partial Autocorrelation (PACF) plot and Autocorrelation (ACF) plots are shown in Appendix2 Figure3. By analyzing them, we got model ARMA(2, 0).

If the autocorrelations are positive for many numbers of lags, then the series needs further differencing. Figure4 in Appendix2 lists ACF and PACF plots after first-order difference. So, we got model ARIMA(0,1,1), ARIMA(0,1,2), ARIMA(0,1,3), ARIMA(0,1,4) and ARIMA(0,1,5).

All the models and their AIC value are listed in Table7 in Appendix1. From the result in Table7, we can see that model ARIMA(0,1,4) has the smallest AIC suggesting this model is the best one. So, we use it to do prediction. The result is shown in Figure5-6.

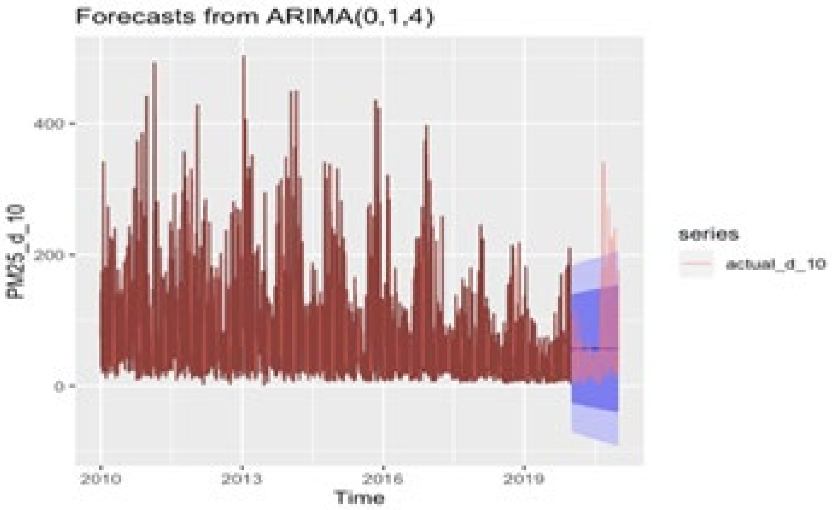

Figure5-6 ARIMA(0,1,4) Predicted and True Values

The red lines are the real values, the blue line is the forecast value, and the blue area represents confidence interval. And the blue area below 0 should be ignored. Some real value exceeded the confidence interval area. Thus, this prediction result is not very well.

For monthly data, we used the decompose() function in R to decompose the monthly series into seasonal, trend and remainder parts. From the Figure5 in Appendix2, we observe obvious seasonal changes. Therefore, we choose seasonal ARIMA model. Since seasonal factors change every twelve months, the period in seasonal ARIMA is 12.

ADF test is used to test the stationarity of the series. P-value in Appendix1 Table8 is smaller than 0.01. We also did first-order difference. By analyzing ACF and PACF images in Figure6 and Figure7 in Appendix2, we get five models and all the models, and their AIC values are shown in Table9 in Appendix1. We can see that ARIMA(0, 1, 1)(0,1,1)[12] has the smallest AIC showing that it is the best model. So, we use it to do prediction. The result is shown in Figure5-7.

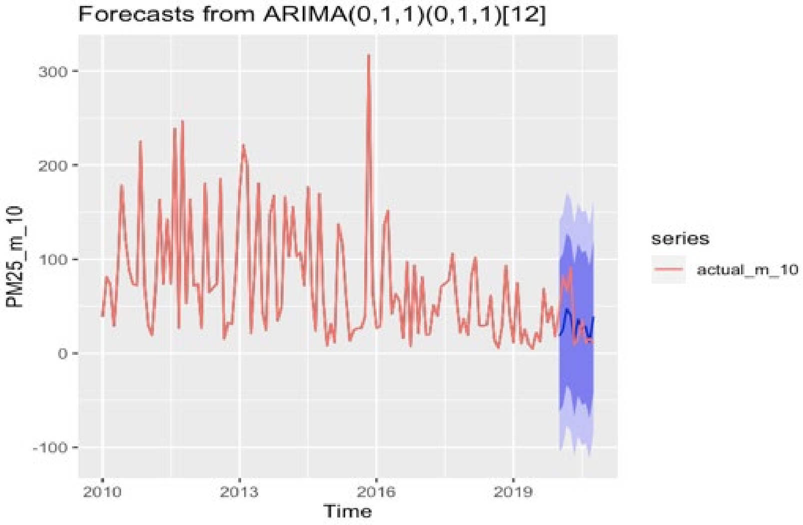

Figure5-7 ARIMA(0, 1, 1)(0,1,1)[12] Predicted and True Value

5.3.2 LSTM model

Although the prediction accuracy of ARIMA model was within an acceptable range, there still exist some errors. To get more accurate prediction results, wo adopted deep learning model – LSTM model. After scaling the daily data and monthly data to the range 0 to 1, we separate the data to the train set and test set. To fit the LSTM model, we set the hidden layers to be 4 and the loss function to be mean squared error. For optimizer, we used the Adam. After training the model, we got the predicted values of PM2.5 for both daily data and monthly data in 2020. The root of mean squared error is 30.5546 for daily data and the root of mean squared error is 31.4546. The results are shown in Figure5-8 and Figure5-9.

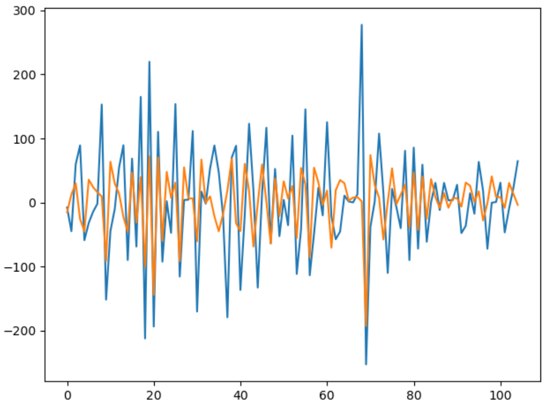

Figure5-8 LSTM predicted values for daily data

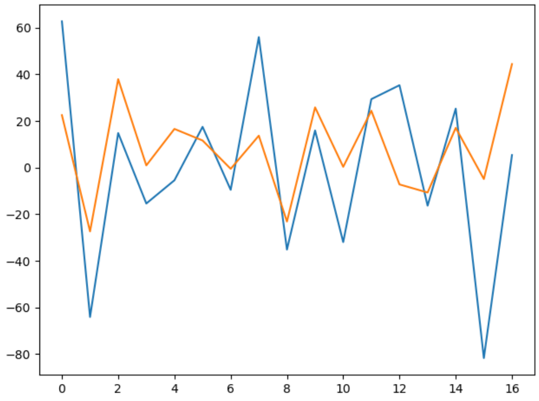

Figure5-9 LSTM Predicted for monthly data

5.3.3 Comparation between two prediction model

Overall, the prediction accuracy of the LSTM model was better than the traditional ARIMA model. For daily data, ARIMA model began to lose its accuracy on forecasts after a few days, and basically maintained the same value. The main reason was that the lag order of the model is small. When predicting later data, the model was not good. For monthly data, the phenomenon has improved significantly, and we have obtained predicted values consistent with the true values trend. However, there were still large errors. For the LSTM model, it captured the information of both long-term data and short-term data to make predictions through training process. Therefore, whether it is daily data or monthly data, the performance of LSTM model was better than ARIMA model. Also, the root of mean squared error for LSTM model is much smaller than that for ARIMA model.

5.4 Trend of PM2.5

In order to check whether PM2.5 decreases over time in statistic, we use Mann-Kendall test and general linear regression.

Mann-Kendall test (MK test) is a climate diagnosis and prediction technology. It can be used to determine whether there are climate mutations in the climate sequence. Simply put, the MK test is a testing method for judging the trend (rising or falling) of a sequence. We did MK test for daily data set and monthly data set, and the results are shown in Table10 and Table11 in Appendix1 separately. We focus on z-value and p-value. That z-value is smaller than 0 indicates a decreasing trend while that z-value is larger than 0 suggests an increasing trend. Here, we can see that both z-value are negative, and p-value are smaller than 0.01. Thus, in statistical, the trend of daily PM2.5 and monthly PM2.5 are decreasing over time, meaning PM2.5 in Beijing is getting better from 2010 to 2020.

For general linear regression model, we added one index column to be the independent variable, which ranged from 1 to the length of data. And we set the PM2.5 values to be the dependent variable. Then we fitted the linear regression model and found that the coefficient of the index was negative, and the corresponding p value was small than 0.05, which meant that PM2.5 had a decreased trend. The results were shown in Table12 and Table13 in Appendix1.

6. Conclusions and Suggestions

From the linear regression model, lasso regression model and random forest, we found that all air pollutants had positive effect on PM2.5, which meant that the higher concentration of these air pollutants, the higher the PM2.5 values. Among them, NO2 and CO played important roles, which indicated that we should control the emission of NO2 and CO. Also, we found that wind speed had the negative relationship with PM2.5 and dew points had the positive relationship with PM2.5. It meant that if the wind speed became higher, PM2.5 became smaller. For some specific wind direction like southeast, it also has significant effect on PM2.5.

From the ARIMA model, we used ARIMA(0,1,4) model for daily data and ARIMA(0, 1, 1)(0,1,1)[12] for monthly data. As the prediction results showed that prediction of monthly data was close to true value and it distributed within confidence interval, which was better than that of daily data. From the LSTM model, its performance was better than ARIMA model. The root of mean squared error was also smaller.

And we used MK test and general linear regression to check whether trend of these two data sets was falling. As the MK test results showed, the z value is negative, and the corresponding p value is smaller than 0.01. As the general linear regression showed, the coefficient of index was negative and the corresponding p value was smaller than 0.01, the trend was significantly decreased. It indicated that PM2.5 in Beijing was getting lower in the past several years.

PM2.5 in Beijing became lower in the past several years, which indicated that the air quality became better. To further improve the air quality, the government should take more measures. Firstly, we should control the source, not to build enterprises with heavy air pollution in the upper direction of the city, and to eliminate outdated technology and equipment. Secondly, we should promote the use of clean energy such as natural gas, hydropower, wind, nuclear and solar energy and minimize the use of polluting fuels such as coal, heavy oil, and waste. Thirdly, we should develop public transportation, reduce the traffic volume, and raise the emission standards of motor vehicles. Fourthly, it is necessary to control domestic pollution. The cooking range hoods should be cleaned regularly, and the dry-cleaning machines should be closed to operate. Sprinkler operations should be used to prevent dust from buildings and roads. Lastly, we must pay attention to agricultural and rural pollution, not burn straw and garbage in the open air, use less pesticides and fertilizers.

Reference

- Brownlee, J. (2020, August 14). How to Check if Time Series Data is Stationary with Python. Retrieved from https://machinelearningmastery.com/time-series-data-stationary-python/
- Srivastava, T. S. (2020, June 26). Time Series Analysis: Time Series Modeling In R. Retrieved from https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/
- Shumway, R. H., & Stoffer, D. S. (2017). Time series analysis and its applications: With R examples. Cham, Switzerland: Springer.
- Huang, X.; Yun, H.; Guan, Z.; Li, X.; He, L.; Zhang, Y.; Hu, M. Source apportionment and secondary organic aerosol estimation of PM2.5 in an urban atmosphere in China. Sci. Sin. Terrae 2014, 44, 723–734.
- Liu Z, Hu B, Wang L, Wu F, Gao W, Wang Y. Seasonal and diurnal variation in particulate matter (PM10 and PM2.5) at an urban site of Beijing: analyses from a 9-year study. Environmental Science & Pollution Research. 2015;22(1):627–642.
- Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., . . . Chen, S. X. (2015). Assessing Beijing’s PM 2.5 pollution: Severity, weather impact, APEC and winter heating. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 471*(2182), 20150257. doi:10.1098/rspa.2015.0257
- Kai Guo. (n.d.). Beijing releases list of top air polluting sources. Retrieved from http://www.chinadaily.com.cn/a/201805/15/WS5afa22eca3103f6866ee8561.html
- Yuan, G., & Yang, W. (2019). Evaluating China’s air pollution control policy with extended AQI indicator system: Example of the Beijing-Tianjin-Hebei region. Sustainability, 11(3), 939.

Appendix1:

List of Tables

Table1: Full linear regression model results for daily data

variable	coef	std err	t	P>|t|	variable	coef	std err	t	P>|t|
const	-189.09	12.17	-15.54	0.00	CO	364.02	23.61	15.42	0.00
TEMP	-52.02	2.71	-19.19	0.00	wind_1.0	-0.07	0.14	-0.53	0.60
DEWP	41.65	2.10	19.85	0.00	wind_2.0	-0.08	0.14	-0.59	0.56
WSPM	27.54	10.98	2.51	0.01	wind_4.0	-0.04	0.14	-0.32	0.75
PRES	190.68	12.12	15.73	0.00	wind_5.0	-0.07	0.14	-0.53	0.60
SO2	4.45	1.36	3.27	0.00	wind_6.0	-0.06	0.14	-0.41	0.68
NO2	38.07	1.04	36.51	0.00	wind_7.0	-0.05	0.14	-0.38	0.71
O3	24.67	1.07	23.15	0.00	wind_8.0	-0.09	0.13	-0.69	0.49

Table2: Full linear regression model results for monthly data

variable	coef	std err	t	P>|t|	variables	coef	std err	t	P>|t|
TEMP	-33.30	17.71	-1.88	0.07	O3	22.92	5.87	3.91	0.00
DEWP	22.71	13.60	1.67	0.10	CO	209.35	130.84	1.60	0.12
WSPM	-60.25	102.09	-0.59	0.56	wind_5	-187.57	70.96	-2.64	0.01
PRES	189.44	70.86	2.67	0.01	wind_6	-187.57	71.01	-2.64	0.01
SO2	6.05	6.45	0.94	0.35	wind_7	-187.81	71.00	-2.65	0.01
NO2	37.67	4.84	7.79	0.00	wind_8	-187.74	71.00	-2.64	0.01

Table3: Variance inflation factors for daily data and monthly data

	VIF(daily)	VIF(monthly)	features
0	32.7	251.9	TEMP
1	11.8	78.2	DEWP
2	13.9	145.1	WSPM
3	281.8	216.9	PRES
4	4.3	12.1	SO2
5	21.1	71	NO2
6	10.5	47	O3
7	10.5	46.1	CO
8	13.2	-	wind_1.0
9	7.5	-	wind_2.0
10	24.1	-	wind_4.0
11	40.5	4.8	wind_5.0
12	31.5	6.2	wind_6.0
13	28.6	8.1	wind_7.0
14	120.5	70.3	wind_8.0

Table4: Lasso regression model results for daily data and monthly data

variables	coef (daily)	coef (monthly)	variables	coef (daily)	coef (monthly)
const	2.27	2.02	wind_1.0	0.00	-
TEMP	0.00	0.00	wind_2.0	-0.03	-
DEWP	-42.14	2.94	wind_3.0	-0.08	-
WSPM	34.90	0.00	wind_4.0	-0.03	-
PRES	0.00	0.00	wind_5.0	0.00	0.00
SO2	0.00	5.68	wind_6.0	0.00	0.28
NO2	5.55	31.60	wind_7.0	0.01	-0.07
O3	33.22	4.77	wind_8.0	-0.08	0.00
CO	8.78	0.00			

Table5: The importance of features for daily data and monthly data (Random Forest)

variables	importance(daily)	importance(monthly)
TEMP	0.013	0.015
DEWP	0.044	0.034
WSPM	0.013	0.029
PRES	0.036	0.035
SO2	0.021	0.074
NO2	0.275	0.51
O3	0.027	0.044
CO	0.564	0.256

wind_1.0	0.001	-
wind_2.0	0	-
wind_4.0	0.001	-
wind_5.0	0.001	0.001
wind_6.0	0.001	0
wind_7.0	0.001	0.001
wind_8.0	0.002	0.002

Table6: Daily PM2.5 ADF test

	Value
Test Statistic Value (ADF)	-11.799
P-value	0.01
Lags Used	15

Table7: AIC of daily PM2.5 models

Model	AIC Value
ARMA(2, 0)	39577.36
ARIMA(0, 1, 1)	40301.66
ARIMA(0, 1, 2)	39542.06
ARIMA(0, 1, 3)	39461.50
ARIMA(0, 1, 4)	39445.42
ARIMA(0, 1, 5)	39447.02

Table8: Monthly PM 2.5 ADF test

	Value
Test Statistic Value (ADF)	-5.6375
P-value	0.01
Lags Used	4

Table9: AIC of monthly PM2.5 models

Model	AIC Value
ARIMA(0, 0, 1)(0,0,1)[12]	1334.97
ARIMA(0, 0, 1)(0,0,2)[12]	1334.08
ARIMA(0, 1, 1)(0,1,1)[12]	1217.06
ARIMA(3, 1, 2)(1,1,1)[12]	1220.50
ARIMA(3, 1, 2)(2,1,1)[12]	1220.25

Table10: Mann-Kendall trend test (MK test) for daily data

	Value
Z Value (MK Test)	-24.558
P-value	< 2.2e-16

Table11: Mann-Kendall trend test (MK test) for monthly data

	Value
Z Value (MK Test)	-5.0723
P-value	3.93e-07

Table12: Linear regression model for trend testing (daily data)

	coef	std err	t	P>|t|
const	111.2203	2.03	54.70	0.00
time	-0.0193	0.00	-21.31	0.00

Table13: Linear regression model for trend testing (monthly data)

	coef	std err	t	P>|t|
const	115.3284	9.91	11.64	0.00
time	-0.6525	0.13	-4.97	0.00

Appendix2:

List of Figures

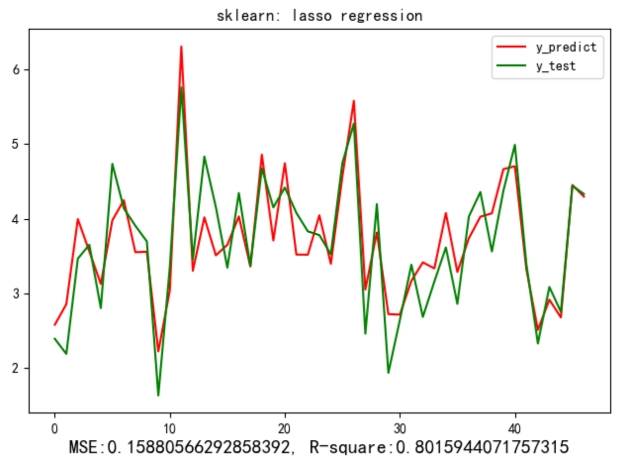

Figure1: Performance of lasso regression (daily data)

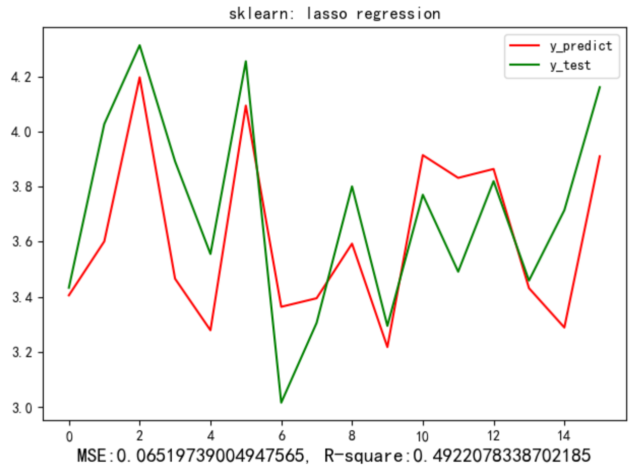

Figure2: Performance of lasso regression (monthly data)

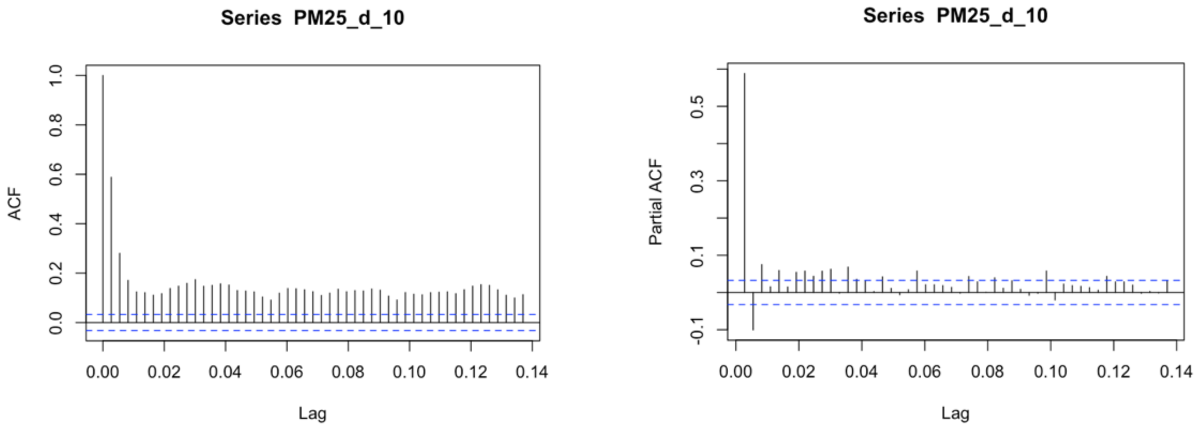

Figure3: Daily data ACF and PACF

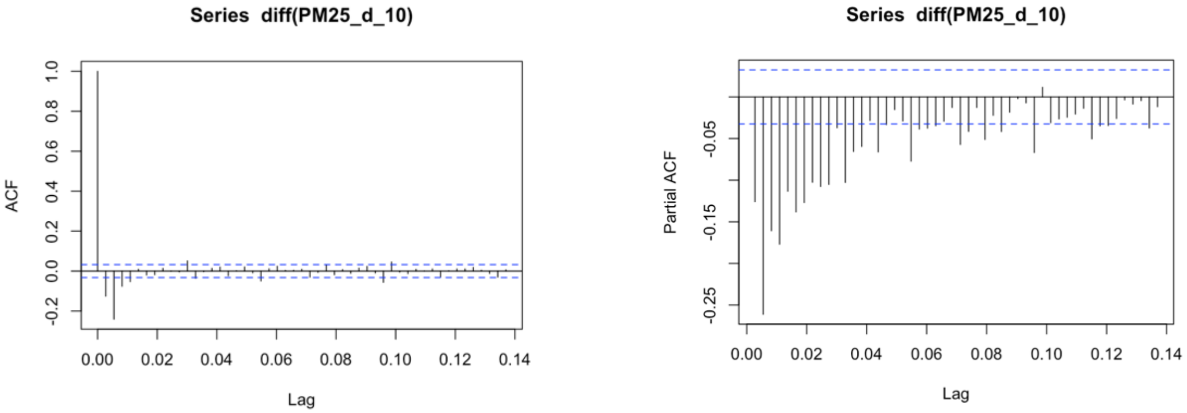

Figure4: Daily data ACF and PACF after differencing

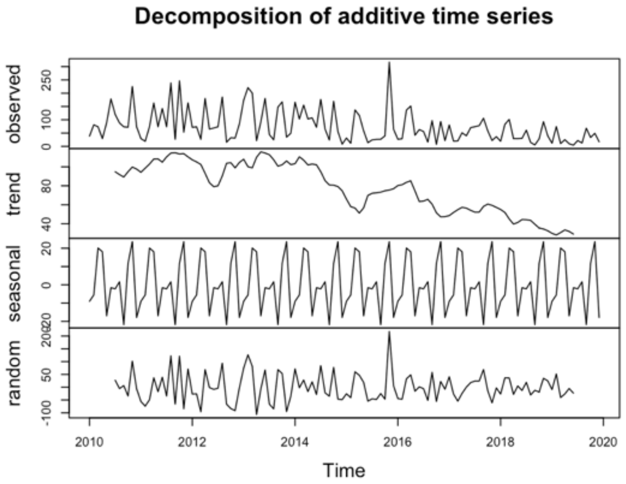

Figure5: Seasonal decomposition of monthly data

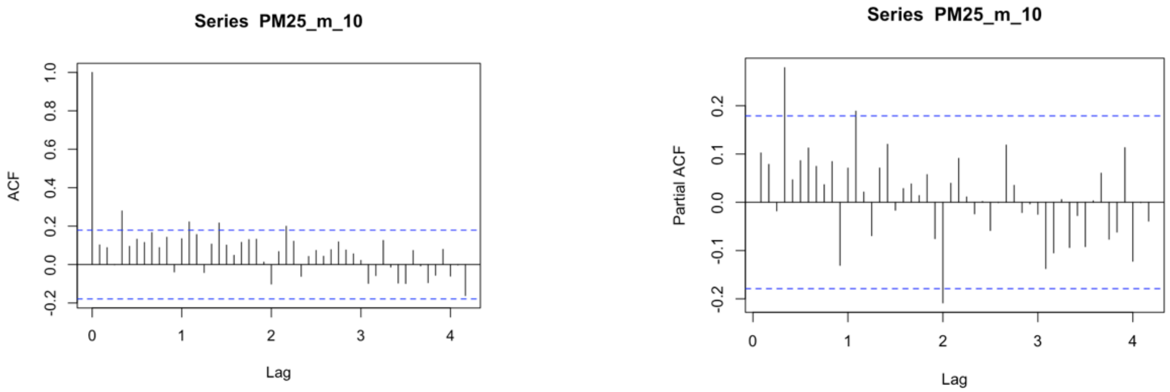

Figure6: Monthly data ACF and PACF

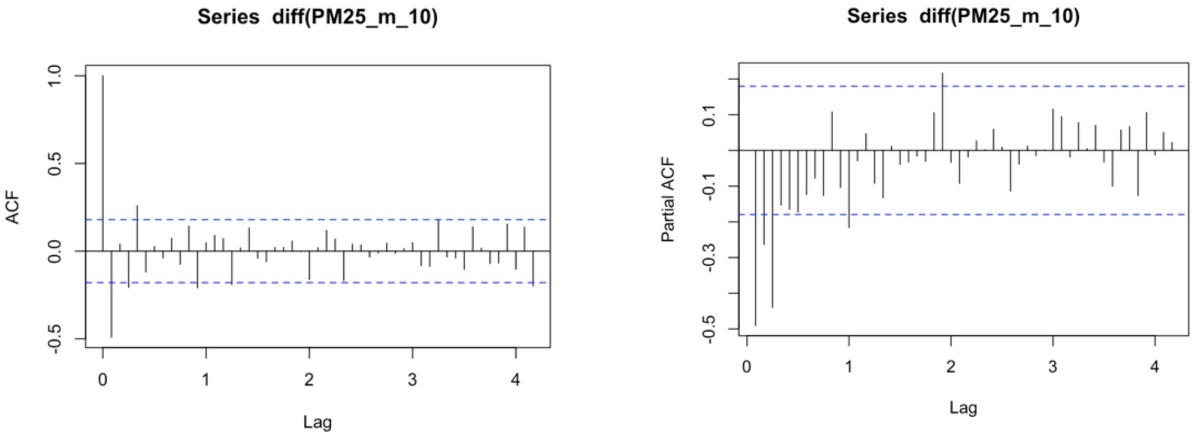

Figure7: Monthly data ACF and PACF after differencing

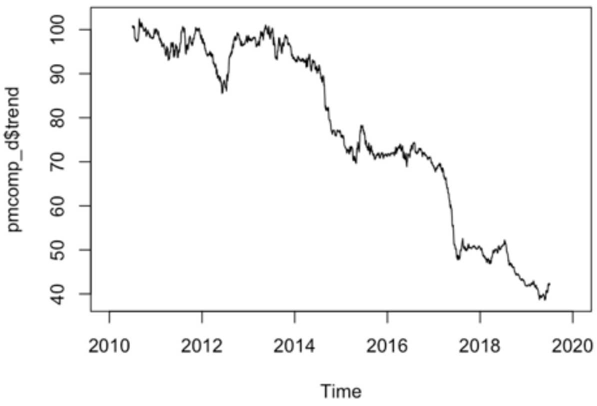

Figure8: Trend of daily data

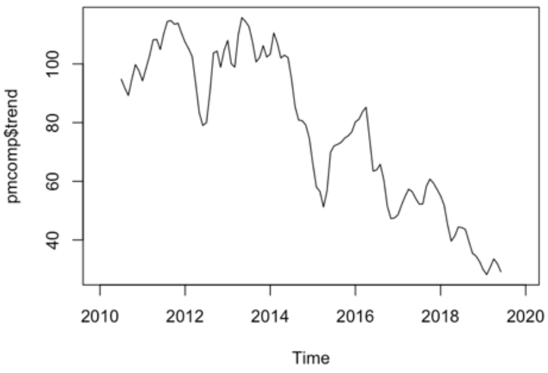

Figure9: Trend of monthly data

Appendix3:

Code Summary

We used R, Python, Excel, and Tableau software. Here, we will give key parts of our code (not the whole code).

Data Preprocessing in R:

```
library(openair)

library(data.table)

library(fastDummies)

library(VIM)

library(sqldf)

pm2.5<-read.csv("C:/Users/User/Desktop/PM2.5.csv",header = T)

names(pm2.5)[1] <- "date"

head(pm2.5)

pm2.5 <- data.table(pm2.5)

str(pm2.5)

pm2.5$date<- as.Date(pm2.5$date, "%m/%d/%Y")

pm2.5[,list(avg=mean(Value)),by=date]

daily <- timeAverage(pm2.5, avg.time = "day")

write.csv(daily,"C:/Users/User/Desktop/daily.csv" )

daily<-read.csv("C:/Users/User/Desktop/daily_data.csv",header = T)

monthly<-read.csv("C:/Users/User/Desktop/monthly_data.csv", header = T)

head(daily)

daily<- kNN(daily,k = 5)

daily<- daily[,c(1:15)]

monthly<- kNN(daily,k = 5)

monthly<-monthly[,c(1:15)]

write.csv(daily,"C:/Users/User/Desktop/daily_data.csv")

write.csv(monthly,"C:/Users/User/Desktop/monthly_data.csv")
```

Model fitting in Python:

```
# random forest

regr = RandomForestRegressor()

regr.fit(data1_1, pm25)

labels = data1_1.columns

importances = np.round(regr.feature_importances_, 3)

imp = []

for f in range(data1_1.shape[1]):

   print(f + 1, labels[f], importances[f])

plt.title("The importance of features", fontsize=18)

plt.ylabel("important level", fontsize=15, rotation=90)

plt.rcParams['font.sans-serif'] = ["SimHei"]

plt.rcParams['axes.unicode_minus'] = False

for i in range(data1_1.shape[1]):

   plt.bar(i+1, importances[i], color='orange', align='center')

   plt.xticks(np.arange(1, data1_1.shape[1]+1), labels, rotation=90, fontsize=15)

plt.show()

# full linear regression

data1_3 = sm.add_constant(data1_2)

x_train, x_test, y_train, y_test = train_test_split(data1_3, log_pm25, random_state=1, test_size=0.02)

lm0 = linear_model.LinearRegression()
```

```
lm0.fit(x_train, y_train)
score0 = lm0.score(x_train, y_train)
print(lm0.coef_)
print(lm0.intercept_)
print(score0)
y_pred1 = lm0.predict(x_test)
plt.plot(range(len(y_pred1)), y_pred1, 'r', label='y_predict')
plt.plot(range(len(y_test)), y_test, 'g', label='y_test')
plt.legend()
plt.xlabel("MSE:{}, R-square:{}".format(metrics.mean_squared_error(y_test, y_pred1), r2_score(y_test, y_pred1)),
           fontsize=14)
plt.title('sklearn: full linear regression')
plt.show()
result1 = sm.OLS(y_train, x_train).fit()
print(result1.summary())
# multicollinearity
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(data1_2.values, i)
                     for i in range(data1_2.shape[1])]
vif["features"] = data1_2.columns
print(vif.round(1))
plt.figure(figsize=(14, 12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(df_con.astype(float).corr(), linewidths=0.1, vmax=1.0,
            square=True, linecolor='white', annot=True)
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.show()
print(df_con.corr())
# lasso regression
clf1 = linear_model.Lasso(alpha=0.0001)
clf1.fit(x_train, y_train)
score1 = clf1.score(x_train, y_train)
print(clf1.coef_)
print(clf1.intercept_)
print(score1)
y_pred2 = clf1.predict(x_test)
plt.plot(range(len(y_pred2)), y_pred2, 'r', label='y_predict')
plt.plot(range(len(y_test)), y_test, 'g', label='y_test')
plt.legend()
plt.xlabel("MSE:{}, R-square:{}".format(metrics.mean_squared_error(y_test, y_pred2), r2_score(y_test, y_pred2)),
           fontsize=14)
plt.title('sklearn: lasso regression')
plt.show()
data2_2 = sm.add_constant(data2_1)
x_train, x_test, y_train, y_test = train_test_split(data2_2, log_pm25, random_state=1, test_size=0.02)
result1 = sm.OLS(y_train, x_train).fit()
print(result1.summary())
# trend
trend_data1 = pd.read_csv('E:\\GWU\\6245Statistical Consulting\\project1_group5\\daily_pm25.csv')
data_v1 = pd.DataFrame(trend_data1['time'])
data_i1 = trend_data1['PM25']
data_v11 = sm.add_constant(data_v1)
lm0 = linear_model.LinearRegression()
lm0.fit(data_v11, data_i1)
score0 = lm0.score(data_v11, data_i1)
```

```
print(lm0.coef_)
print(lm0.intercept_)
print(score0)
result1 = sm.OLS(data_i1, data_v11).fit()
print(result1.summary())
```

ARIMA model fitting in R:

```
# adf test
adf.test(PM25_d_10)
#acf, pacf test
acf(PM25_d_10, 50)
pacf(PM25_d_10, 50)
pmcomp_d <- decompose(PM25_d_10)
plot(pmcomp_d$trend)
#fit arima model
(fit2 <- arima(PM25_d_10, c(2, 0, 0)))
#with diff
acf(diff(PM25_d_10), 50)
pacf(diff(PM25_d_10), 50)
#models
(fit3 <- arima(PM25_d_10, c(0, 1, 4)))
(fit4 <- arima(PM25_d_10, c(0, 1, 3)))
(fit5 <- arima(PM25_d_10, c(0, 1, 2)))
(fit6 <- arima(PM25_d_10, c(0, 1, 1)))
(fit7 <- arima(PM25_d_10, c(0, 1, 5)))
#predict
autoplot(forecast(fit3, 365)) + autolayer(actual_d_10, alpha=0.5)
# daily data trend
mk.test(project10$PM25, continuity = TRUE)
plot(project10$PM25, type = 'l',xlab = 'day',ylab='PM2.5')
```

LSTM model fitting in Python:

```
def create_dataset(dataset, look_back):
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back - 1):
        a = dataset[i:(i + look_back)]
        dataX.append(a)
        dataY.append(dataset[i + look_back])
    return np.array(dataX), np.array(dataY)



look_back = 1
trainX, trainY = create_dataset(trainlist, look_back)
testX, testY = create_dataset(testlist, look_back)


trainX = np.reshape(trainX, (trainX.shape[0], trainX.shape[1], 1))
testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))


# create and fit the LSTM network
model = Sequential()
model.add(LSTM(4, input_shape=(None, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
joblib.dump(model, "lstm_daily.dat")
```