Title of report

BDAS (Steps 1-8)
Investigation of Beijing air pollution (PM25) using Machine Learning methodology

Commissioned by and prepared for:

Data Mining and Big Data course

Professor:

David Sundaram

Student name:

Mehdi Tadayoni

Submission Date:
October 2020

Contents

Figures

Tables

Summary

The main target of this study is Data mining and using different kind of machine learning (supervised method) to predict the most important air pollution matter which is called PM25 (particulate matter with diameter less than 2.5 _mμ). In this study, Beijing City is divided into 6 12 stations with same data. Each air quality monitor stations location, as well as the date information (month, day, hour) are included in the feature spaces. Major air pollutant (PM25, PM10, CO, NO2, O3) measurements and climate information (Temperature, Pressure, Wind speed, DEWP) are also recorded they were used as an input. In this project after data cleaning and choosing the best data mining methodology some Regression algorithms such as Multi Linear and Generalized Linear Regression approaches by using the different PySpark Packages were used. In this study the target was using simple method to get high correlation between inputs and target. But the important aspect of this study is searching to find relationship between only one parameter (PM10) as an input and air pollutants such as PM25. However, climate data such as Temperature, pressure, wind speed were used as an independent data in pattern 2 and 4 but this study shows the regression models are not effective to predict pollutant material by only climate data. The first pattern includes combination of climate data and air pollutants worked very well and the correlation was about 0.84. However, the amazing achievement was in pattern-3, using only one input (PM10) was only imported to Simple Linear model and correlation was 0.82 approximately and based on meeting the criteria (r2>=0.7), we can generalize this approach to many station to other cities in China.

1 BUSINESS AND/OR SITUATION UNDERSTANDING

1.1 IDENTIFY THE OBJECTIVES OF THE BUSINESS/SITUATION

These days most of the developing countries in different kinds of regions are suffering from air pollution. Various elements like fossil fuels consumption and industrial plant emissions play significant roles in air pollution in different countries. Based on WHO statistics, Air pollution kills about seven million people worldwide each year. WHO information indicates 9 out of 10 people breathe air containing high levels of pollutants. Meanwhile, air pollution leads to a major threat to people's health and weather quality. Based on WHO data, air pollution causes about 4.2 million deaths per year due to stroke, heart disease, lung cancer, and chronic respiratory diseases (World Health Organization, 2020).

Also, WHO shows air pollution influence developed and developing countries, low- and middle-income countries experience the highest-burden, with the greatest toll in the South-East Asia regions. **Fig. 1** shows the summation of mortality per 100,000 populations due to air pollution from 1990 to 2017 in different countries. Also, the distribution of mortality in some countries is clear in Fig. 2. Based on Fig. 3, the polluted countries are divided into 3 sections. The blue part reflects the most polluted countries and the red one indicates the developing countries with a high ranking in air pollution. In this category, China as a country with the most population in the world, high GDP, and high in air pollutants will be discussed in this research (Samet, J.M,2000). Below shows the PySpark code to generate the total death due to pollution number vs countries:

```
dfpp = dfp.toPandas()


fig = plt.figure(figsize = (5, 15))
plt.barh(dfpp['Country'], dfpp['sum(DeathPollution)'])

plt.xlabel("DeathPollution")
plt.ylabel("Country")
plt.title("country between 3000 to 5000 Death vs 100000 population")
```

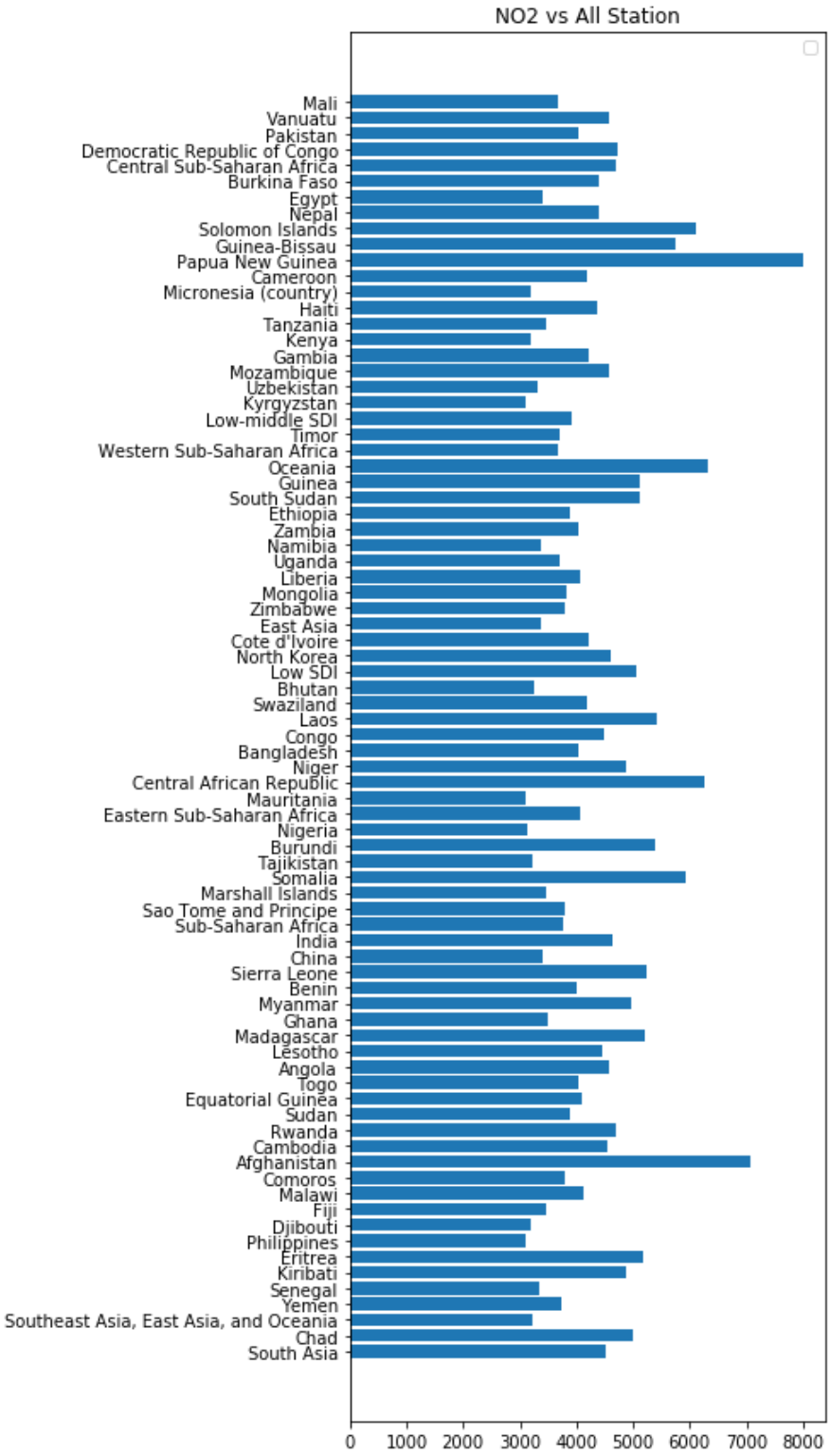


Fig. 1: total death per 100000 due to air pollution among countries from 1990 to 2017 in 50 countries that mortality [https://www.kaggle.com/akshat0giri/death-due-to-air-pollution-19902017].

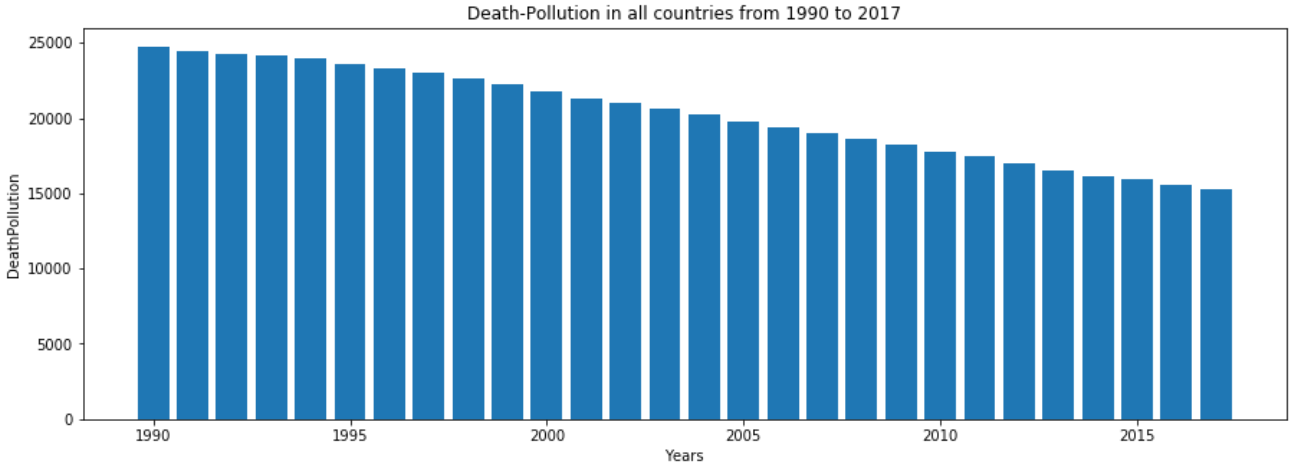


Fig. 2: death per 100000 population due to air pollution among some countries from 1990 to 2017 in three portions

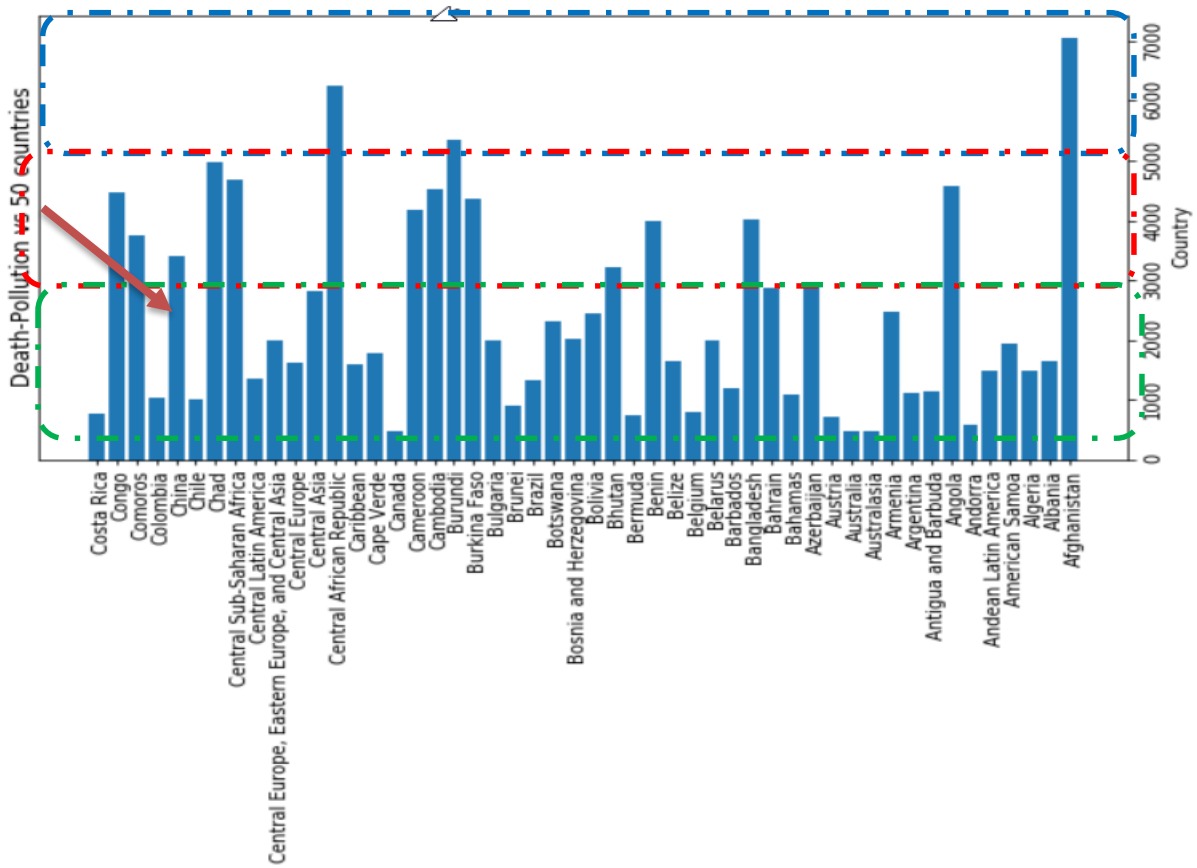


Fig. 3: summation of death per 100000 due to air pollution among countries from 2000 to 2017 in three portions

It is estimated that about 90% of the world's population lives in countries where the air pollution problem. The most important pollution sources are cars, power generation, the heating systems in houses, and industrial activities and also, there is a close relationship between air pollution and weather quality globally. In mentioned countries, the government or Environmental organizations usually consider the regulation of pollutants in the air as a vital issue. Air Pollution topic has been one of the most important environmental consideration in China. In 2017, 75 per 100000 people have been died due to a lack of the national standard for air quality (Fig. 4). Among the cities, Beijing with about 20 million people encounters a rising population on one hand and economic development on the other hand. In all industry, coal and fossil fuel are the majority type of energy which are used in this city. Air pollution in Beijing depends on two major elements; weather conditions and human activity. Local weather parameters such as Wind speed, Humidity, Temperature, Pressure in Beijing have a great impact on air pollution.

Outdoor air pollutants in Beijing mostly include ozone (O3), particle matter (PM25 and PM10), sulfur dioxide (SO2), carbon monoxide (CO), nitrogen oxides (ozone). Increased mortality rates have been determined in association with rising air pollutants including sulfur dioxide, PM25 , and PM10 concentrations (Du, X., Kong, 2010).

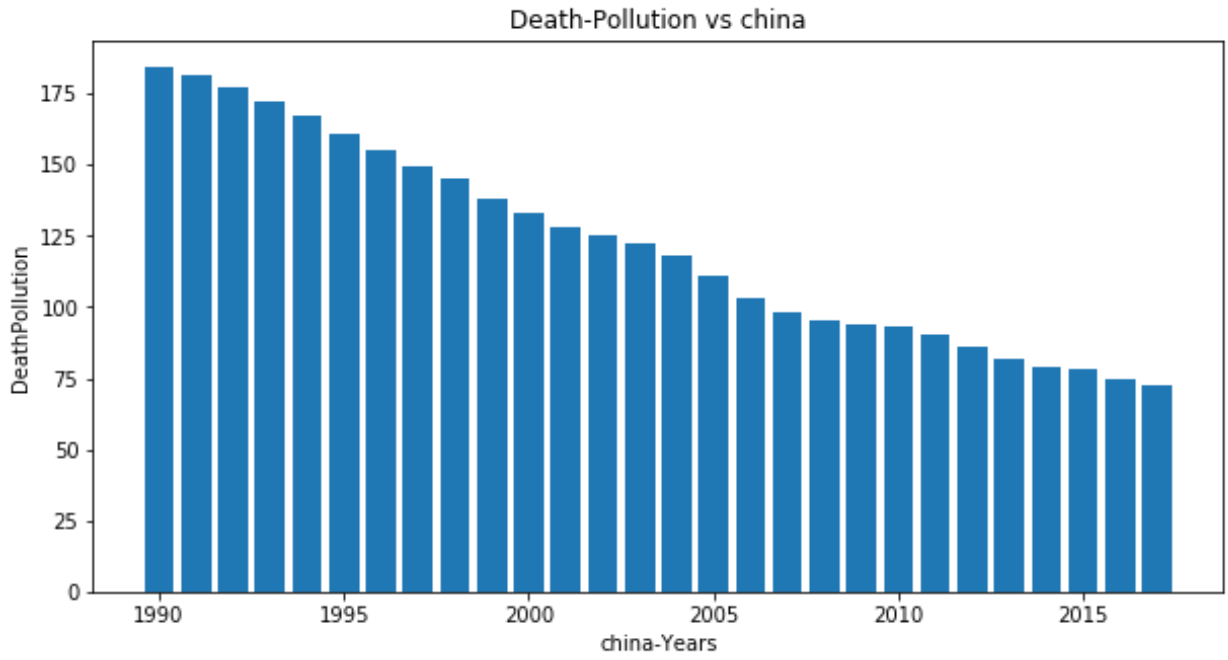


Fig. 4: distribution of air pollution total death from 1990 to 2017 in china

Year	Country	avg(DeathPollution)
1992	China	177.2834672
2005	China	111.3419029
1997	China	149.6834218
2012	China	85.77144184
1999	China	138.2017334
1990	China	184.4132044
1998	China	144.8416199
1996	China	155.3857949
2002	China	124.9978735
2007	China	97.83587858
2000	China	133.1366852
2001	China	128.0940057
2008	China	95.4391289
2003	China	122.3120249
1994	China	167.0829393
2013	China	81.73468623
2010	China	92.97014294
2011	China	90.6548837
2016	China	74.70570935
1991	China	181.8007043

Hence, this research will be conducted with the following objectives as below:

- Based on above fining most of countries are involved yearly death because of air pollution. This issue is very important for china due to high population who are concentrated in Beijing. Above table and figure 4 shows the average of death in china because of air pollution. However, any investigation to get information about air pollutant material and research about their relationship and factors which have impact on them are very important.
- The fine particles (PM25 to 10) are associated with negative impacts on the respiratory morbidity of people and these elements are able to deposit into the lung gas-exchange region, so any investigation or prediction of this element in the air can help to prevent more death among people in Beijing (Du, X., Kong, 2010). Also, SO2, Co, O3, and No2 are important gases in the air which are regarded with respiratory problems.
- Local weather parameters such as Wind speed, Humidity, Temperature, Pressure, are very vital in calculating the air pollutant concentrations (Chan, C.K, 2008 and Chen, W.Y. & Xu, R.N,2010). Based on the different research, there is a direct relationship between the weather elements such as temperature, wind speed, pressure, and humidity with the concentration of gases and particle matter (PM25) (Holloway, 2008). For example, High humidity usually happens with high concentrations of PM25, CO, and SO2 but with low concentrations of other air gases like NO2 (Elminir,2005).

- On the other hand, Increasing or decreasing wind speed has a different impact on air pollution measurement. For instance, when the wind speed was low, the pollutants (PM25) were found at the highest concentrations. However, by increasing wind speeds we have a high concentration of dust and other particles (Elminir,2005). Therefore, these important meteorological attributes have a great impact on air pollution and they can be considered for the prediction of air pollutant concentrations.
- In the previous study, statistical computation has been used frequently for air pollution estimation for a couple of days based on some limited weather elements (Zheng, Y, 2013). Currently, some studies on statistical modeling have been focused on some simple regression models to predict PM25 or very limited particles. PM25 is a very important particle that has not been investigated with comprehensive input data.

- Besides statistical models, machine learning methodology has been enhanced recently with a huge amount of application in different aspects of life and industries (Yuan, Z, 2017). Fortunately, different new methods in Machine learning and deep learning have been invented which allow us for data mining of air pollution and Meteorological parameters comprehensively
- In this research, the main objective is data mining of massive air pollution particles (PM25), other pollutant data and weather data (Temperature, Pressure, Humidity, Wind speed) as a first step and evaluating all Machine Learning methods to predict the pollutant effectively in 12 regions (or stations).
- A considerable difference between this study and the previous ones is that this research will try to focus on utilizing all ML models to enhance the generalization performance of a complex model from big historical data from 2013 to 2017 in 12 regions in Beijing (Kurt, A,2010).

- **Business Success Criteria**

In this research, all important variables such as meteorological parameters and air pollution data in 2 different and separate models will be used as inputs for the prediction of PM25. Evaluation methods such as SSE, SSR, or SST will be used for the accuracy of different models. However, in Machine Learning methodology the R2 about 0.70 and more could be reliable.

1.2 ASSESSING THE SITUATION

For the prediction of the pollutant material, it needs some Meteorological parameters such as Temperature, Pressure, Humidity, and wind speed. All data comes from different regions in Beijing in CSV files and any station includes 18*35000 rows of data. However, in this research, pollution factors (PM25) will be predicted and weather conditions and other pollutants in separate ML models are inputs in different kinds of machine learning approaches (regression, classification, clustering, deep learning…). In this research, different risks must be considered. Firstly, we face with big data as input. In this case, by the concept of data mining, we can refine the data and remove unrelated data (p-value can be a good method). Secondly, utilizing different ML approaches needs considerable time. This problem needs to make an effective plan. The last risk is using wind direction as an input to the model. By some research on different papers and also criteria such as R^2, we can check the relationship with other parameters.

- **Resource Inventory**

For running different ML approaches and software such as, Python, R, SQL, Weka, it needs a normal laptop with enough memory. There is no certain limitation for any hardware specification. The air pollutant and meteorological data came from the UCI repository[https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data]. They are in CSV format in 12 categories in different regions in the Beijing area. They are free and stored in 2019 in the UCI repository website and show air pollution data from 1990 to 2017.

- **Requirements, Assumptions, and Constraints**

There are no special requirements or any constrain to access the data. All of them are free and downloadable from the website. This data set contains air pollution data in 12 nationally-controlled air-quality monitoring sites in China-Beijing. The air-quality data such as temperature, pressure…are from the Beijing Municipal Environmental Monitoring Center legally. The meteorological data in each air-quality site are matched with the nearest weather station from the China Meteorological Administration [https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data].

- **Risks and Contingencies**

Data comes from a valid source and there is a guarantee for data quality. Also, the first screening of data shows relative validity. However, if there is any missing or Null data, we can monitor and estimate them in the data mining phase.

- **Terminology**

There is some expression of data as Tab. 1:

Tab. 1: a data description

Specification	Description
station code	**Twelve stations in Beijing**
SO2	**sulfur dioxide**
NO2	Nitrogen Dioxide
PM10	**particle matter**
PM25	**particle matter**
O3	**ozone**
Co	**carbon monoxide**

- **Cost/Benefit Analysis**

There is no cost in this project. Only it needs time which will be based on the project plan. One of the benefits of this project is the optimization of various Machine learning methods to predict air pollution elements by available and free meteorological data.

1.3 DETERMINE DATA-MINING OBJECTIVES

Up to now the business objective is obvious and it needed to turn it into a data mining concept. Relationship of rising the air pollutants to increasing of rate of mortality is the fundamental objective of the business but the estimation and measuring this kind of pollutants (in Beijing) are very critical information to help the business to protect the health of the people in society.
Based on understanding of air pollution problem in China (Beijing) we can summaries the data mining goals as below:

- The dataset which was collected includes 12 station in Beijing with two category of air pollution and climate data with suitable history from 2013 to 2017.

- It supposed there is close relationship between all air pollutants together. For example, it seems by increasing the NO2, probably the PM25 will increase. So one of the data mining goal in this study is fining the relationship between all materials; PM25, SO2, CO, O3, and NO2 together.
- Another important goal is finding relationship between air pollutant materials with climate data. It supposed to by decreasing the temperature, the PM25 will increase.
- By fining the correlation and relationship between all input data, using the historical data about previous information to generate a model by applying the machine learning is the final goal of data mining.
- The effective plan is dividing all data into 2 stages; training, testing which will be done by Python (PySpark). the air pollution matters such as PM25 will be the target with climate data with other gas data in *2 different machine learning models are the inputs*. In the first model, climate data such as temperature, pressure, wind speed are used as an input data and in the second model, more data such as climate and other air pollution matter such as Co2, No2, So2, will be used as a second input data. Finally, the best model will be introduced.
- In the future, by having the correct model of air pollution prediction, we are able to forecast the quantity of martials easily.

- **Data Mining Success Criteria**

Different evaluation methods will be used for Machine learning assessment. RMSE is the most popular evaluation metric used in the ML regression process. It says that errors are unbiased and follow a normal distribution and R-Squared or R^2 is another reliable method for prediction evaluation. Generally, in the training and testing sections, because of its importance in the ML workflow, the R^2 could be at least 0.70 to 0.80. However, the other models such as F-Measure or Confusion Matrix will be considered in ML (classification approach) workflow.

1.4 PROJECT PLAN

Tab. 2: Project plan for data mining project

Phase	Time	Description
Business Understanding	**1 week**	
-Objectives	2 days	Business objectives, success, and criteria
-Situation Assessment	2 days	Inventory of requirement, assumptions, constraints
-Data mining Goals	2 days	Determination of Data mining goals and success
-Project Plan	1 day	Making a project plan
Data Understanding	**1 week**	
-Initial Data Collection	2 days	Proving data collection report
-Data description	2 days	Proving data description report
-Data exploring	2 days	Proving data exploring report
-Data quality verification	1 day	Proving data verification report
Data Preparation	**3 weeks**	
-Data selection	5 days	Inclusion and Exclusion of data
-Data cleaning	4 days	Proving data cleaning report
-Data construction	4 days	Attributes derive
-Data integration	5 day	Data merging
-Data formatting	3 days	Data reformatting
Modeling	**2 weeks**	
-Model selection	4 days	Different ML models selection
-test designing	4 days	Test designing
-Model building	3 days	Parameters models setting
-Model assessment	3 day	Revised parameters settings
Evaluation	**1 week**	
-Result Evaluation	2 days	Assessment of data mining results
-Process Review	2 days	Review of process
-possible action decision	3 days	Providing the list of possible action
Deployment	**2 week**	
-Plan deployment	4 days	
-Plan monitoring	4 days	
-Produce Final Report	3 days	
-Project review	3 days	

2 DATA UNDERSTANDING

2.1 COLLECTING INITIAL DATA

Daily and hourly API from 2013 to 2017 at the data center of the Ministry of Environment Protection of China was gathered. Data from a total of 12 stations (include: Gucheng, Nongzhanguang, Guanyuan, Tiantan, Dongsi, Wanshouxigong, Aoti, Wanliu, Changping, Huairou, Shunyi, Dingling in Beijing area) was publicized on this website and can be downloaded freely. Each dataset was divided into three sections; the first section includes Year, Month, Day,

and Hour from 2013 to 2017. The second one contains air pollution elements; PM10, PM25 , SO2, NO2, CO, and O3. The last portion includes climate parameters such as Temperature, Pressure, Humidity, Wind speed measurement, and the name of the station. Among 12 distribution of the monitoring stations in Beijing, eight one is located in urban stations (red zone in Fig. 5), three of them are situated in the suburb of Beijing (purple one in Fig. 5) and one background station is located at Dingling, far away from human activity (green one in Fig. 5).

This information comes from air-quality monitoring sites in the Beijing Municipal Environmental Monitoring Center legally. Also, the meteorological data was taken from the nearest weather station from the China Meteorological Administration [https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data].

Tab. 3: data sample from Aotizhongxin station

year	month	day	hour	PM25	PM10	SO2	NO2	CO	O3	TEMP	PRES	DEWP	RAIN	wd	WSPM	station
2013	3	1	0	4.0	4.0	4.0	7.0	300	77.0	-0.7	1023.0	-18.8	0.0	NNW	4.4	Aotizhongxin
2013	3	1	1	8.0	8.0	4.0	7.0	300	77.0	-1.1	1023.2	-18.2	0.0	N	4.7	Aotizhongxin
2013	3	1	2	7.0	7.0	5.0	10.0	300	73.0	-1.1	1023.5	-18.2	0.0	NNW	5.6	Aotizhongxin
2013	3	1	3	6.0	6.0	11.0	11.0	300	72.0	-1.4	1024.5	-19.4	0.0	NW	3.1	Aotizhongxin
2013	3	1	4	3.0	3.0	12.0	12.0	300	72.0	-2.0	1025.2	-19.5	0.0	N	2.0	Aotizhongxin
2013	3	1	5	5.0	5.0	18.0	18.0	400	66.0	-2.2	1025.6	-19.6	0.0	N	3.7	Aotizhongxin
2013	3	1	6	3.0	3.0	18.0	32.0	500	50.0	-2.6	1026.5	-19.1	0.0	NNE	2.5	Aotizhongxin
2013	3	1	7	3.0	6.0	19.0	41.0	500	43.0	-1.6	1027.4	-19.1	0.0	NNW	3.8	Aotizhongxin
2013	3	1	8	3.0	6.0	16.0	43.0	500	45.0	0.1	1028.3	-19.2	0.0	NNW	4.1	Aotizhongxin
2013	3	1	9	3.0	8.0	12.0	28.0	400	59.0	1.2	1028.5	-19.3	0.0	N	2.6	Aotizhongxin
2013	3	1	10	3.0	6.0	9.0	12.0	400	72.0	1.9	1028.2	-19.4	0.0	NNW	3.6	Aotizhongxin
2013	3	1	11	3.0	6.0	9.0	14.0	400	71.0	2.9	1028.2	-20.5	0.0	N	3.7	Aotizhongxin
2013	3	1	12	3.0	6.0	7.0	13.0	300	74.0	3.9	1027.3	-19.7	0.0	NNW	5.1	Aotizhongxin
2013	3	1	13	3.0	6.0	7.0	12.0	400	76.0	5.3	1026.2	-19.3	0.0	NW	4.3	Aotizhongxin
2013	3	1	14	6.0	9.0	7.0	11.0	400	77.0	6.0	1025.9	-19.6	0.0	NW	4.4	Aotizhongxin
2013	3	1	15	8.0	15.0	7.0	14.0	400	76.0	6.2	1025.7	-18.6	0.0	NNE	2.8	Aotizhongxin
2013	3	1	16	9.0	19.0	9.0	13.0	400	76.0	5.9	1025.6	-18.1	0.0	NNW	3.9	Aotizhongxin
2013	3	1	17	10.0	23.0	11.0	15.0	400	74.0	4.3	1026.3	-18.7	0.0	NNE	2.8	Aotizhongxin
2013	3	1	18	11.0	20.0	8.0	20.0	500	70.0	3.1	1027.4	-18.4	0.0	NNE	2.1	Aotizhongxin
2013	3	1	19	8.0	14.0	12.0	30.0	500	60.0	2.3	1028.3	-18.4	0.0	N	2.8	Aotizhongxin

```
df_Aotizhongxin=df[df['station']=='Aotizhongxin']
df_Aotizhongxin.show()
```

```
df_Aotizhongxin.count()
```

```
35064
```

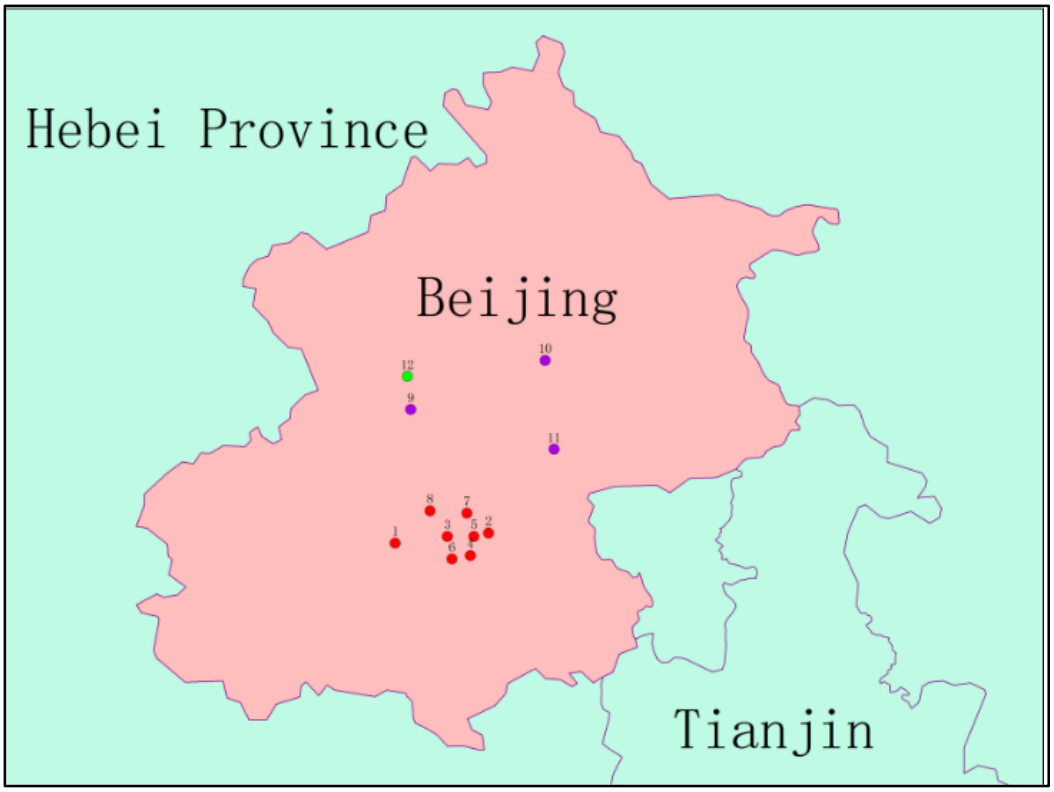


Fig. 5: Distribution of National Air Quality Monitoring Stations in Beijing(Wei Chen,2015)

- **Additional data**

Air Pollution is one of the important reasons for the killing of innocent people in different countries. So for in the previous chapter, this data was used to open the eyes and give you a great insight into the importance of air pollution's effect on lives. However, this kind of data will not be used in our Machine learning model. This data includes some column as below:

- Entity: It is the name of the country.
- Code: It is the Code of the country.
- Year: it shows the period 1990 to 2017
- Air pollution (total) (deaths per 100,000): total death
- Indoor air pollution (deaths per 100,000): death due to indoor air pollution.
- Outdoor particulate matter (deaths per 100,000): death due to outdoor pollution.
- Outdoor ozone pollution (deaths per 100,000) : Death due to ozone pollution [https://www.kaggle.com/akshat0giri/death-due-to-air-pollution-19902017].

2.2 DESCRIBING DATA

Tab. 4 explains the data type and size in all stations. It shows specification, data type, data unit, and description. All 12 stations data were loaded in Python software and they were appended together as a unique dataset. Based on that the number of row data about 420000 (

Parameters	format
year	int
month	int
day	int
hour	int
PM25	double
PM10	double
SO2	double
NO2	double
CO	int
O3	double
TEMP	double
PRES	double
DEWP	double
RAIN	double
wd	string
WSPM	double
station	string

Tab. 5). All air pollution data (So2, No2, PM25 , Co, O3) have a negative impact on lives and they could be as a target in the model. They are in the same format in unit and type. Other climate data such as Temperature, Pressure, Wind Speed are the input to the model. The time is divided into different categories from year to an hour in 12 situations in Beijing (Tab. 4). Below shows the Python code to show the data format including string, integer, and double. However, Table-5 indicates average of air pollution pollutants in all stations together.

```
from pyspark.sql.types import (StructField,StringType,IntegerType,StructType)
dataType=dfn.dtypes
Schema_dataType=StructType([StructField('Parameters',StringType())\
                           ,StructField('format',StringType())])
dataType=spark.createDataFrame(dataType,schema=Schema_dataType)
dataType.show()
```

Tab. 4: Brief description of data properties

Parameters	format
year	int
month	int
day	int
hour	int
PM25	double
PM10	double
SO2	double
NO2	double
CO	int
O3	double
TEMP	double
PRES	double
DEWP	double
RAIN	double
wd	string
WSPM	double
station	string

Tab. 5: Appended 12 file in a unique dataset

```
df.groupBy('station').mean('PM25','PM10','SO2','NO2','CO').show()
```

```
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|      station|        avg(PM25)|         avg(PM10)|          avg(SO2)|          avg(NO2)|           avg(CO)|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|    Changping|71.09974336541265| 94.65787077315701|14.958905587176204| 44.18208550745705|1152.3013445428255|
| Aotizhongxin|82.77361082632768|110.06039131194318| 17.375901409358608| 59.30583318645163|1262.9451453977408|
|Wanshouxigong|85.02413582402235|112.22345864661655|17.148603110917286|55.529560136986305|1370.3950306512274|
|       Wanliu|83.37471599100398|110.46461759631974| 18.376480570616696| 65.25878926575277|1319.3535125706724|
|       Dongsi|86.19429678848283|110.33674190837705| 18.53110660736606|53.699442802498275|1330.0691310760349|
|       Shunyi|79.49160200286961|  98.7370263066404|13.572038687514805| 43.90886473782605|1187.0639785927142|
| Nongzhanguan|84.83848298292484|108.99109577171902|18.689242012825694| 58.09717238740834|1324.3501978852855|
|      Tiantan|82.16491115828659|106.36367249833174| 14.367615106345372| 53.16264557400931| 1298.303317814839|
|     Dingling|65.98949686451802| 83.73972332015809|11.749649653404786|27.585466754360024| 904.8960728548954|
|      Huairou|69.62636686112984| 91.48269023244961|12.121553010210071| 32.49725021391175|1022.5545449140955|
|      Gucheng|83.85208902318554|118.86197849090333|15.366161622826057| 55.87107495348295|1323.9744229569558|
|     Guanyuan|82.93337203901532|109.02330301717915| 17.59094149754264| 57.90164251707599| 1271.294377232746|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
```

```
df.count()
```

```
420768
```

2.3 EXPLORING DATA

Based on business and datamining goals, exploration the data with charts, graphs or tables can assist us to address the goals which were mentioned in business understanding stage. The most important issue in this stage is understanding those air pollution data and their relationship to each other beneficially. Sometimes by declining of some parameters such as temperature, the air pollutants such as PM25 will be increased, Fig. 6 indicates that the PM25 is maximum in temperature -10 to 0. Another important exploration was the relationship between increasing of PM25 and time. Fig. 7 shows that PM25 usually rise in end and beginning of the year. It should say that increasing of air pollution in this time could be due to increasing of travelling of people in Beijing. Meanwhile by increasing the other air pollutant such as NO2, the PM25 will increased intensively. So the exploration of data pattern shows different relationship between PM25 and the other data.Another important of data exploration is investigating on air pollutants pattern from 2013 to 2017.

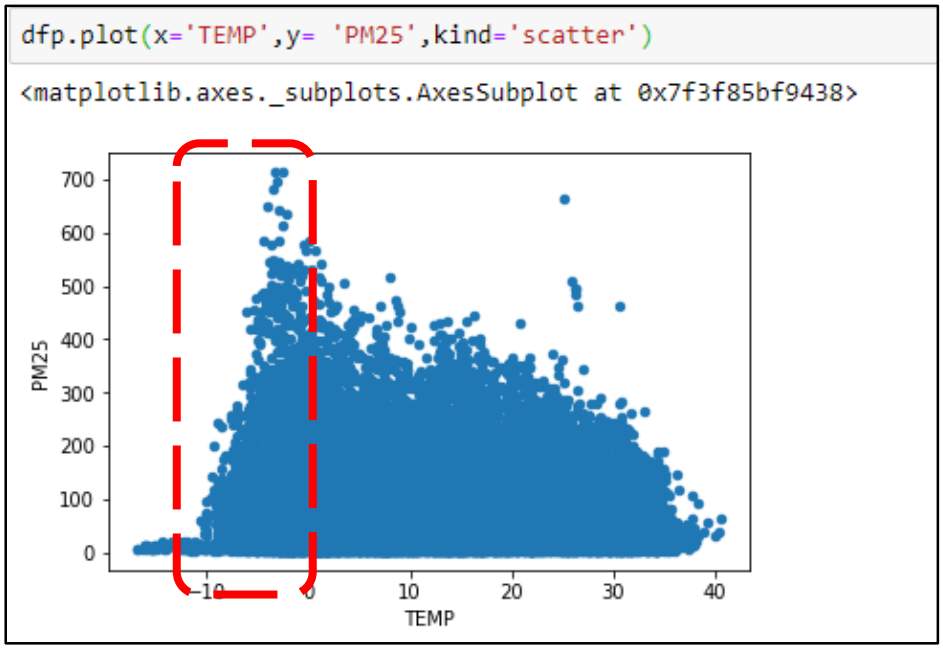


Fig. 6 Relationship between PM25 and temperature data from 2013 to 2017 in all station

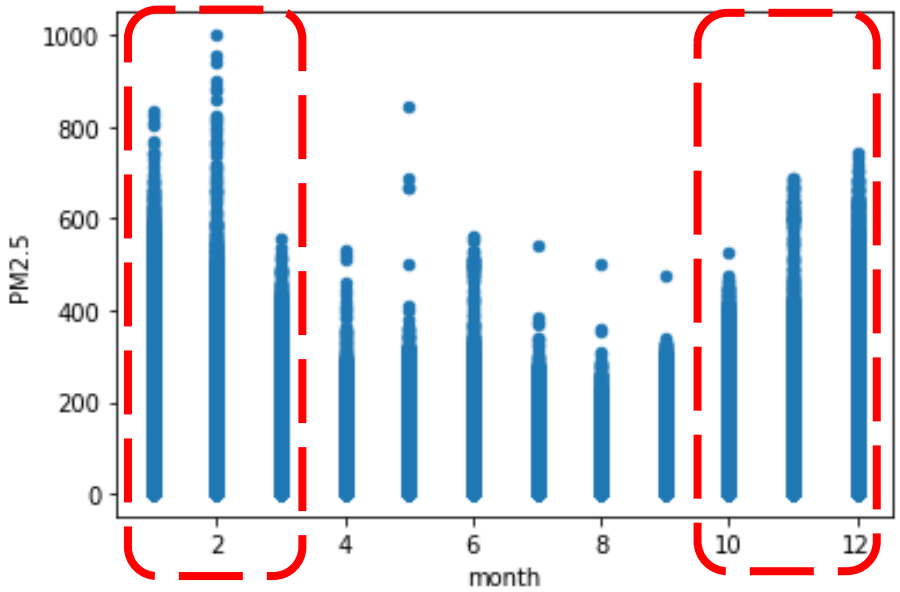


Fig. 7 Relationship between PM25 and month data from 2013 to 2017

Investigation to find the best correlation between parameters are very important in this section. Fig. 10 to Fig. 14 indicate the relationship between air pollution materials by PM25 from 2013 to 2017. Based on them, PM25 have a strong correlation with PM10, NO2 and CO. but the impact of

SO2 and O3 is not the same as others and there is no clear and obvious pattern with PM25. In most of them, the amount of pollutants is increasing gradually from 2013 to 2016.

Also, the distribution plot between PM25 and all climate parameters (Fig. 13 and Fig. 14) will be seen. Also there is no strong correlation between PM25 and other pressure and temperature data. It shows the correlation between air pollutant material with climate data is weak and probably for modelling it needs to use air pollution data as an input. These kinds of graphs help us to find relevant or irrelevant climate parameters effectively.

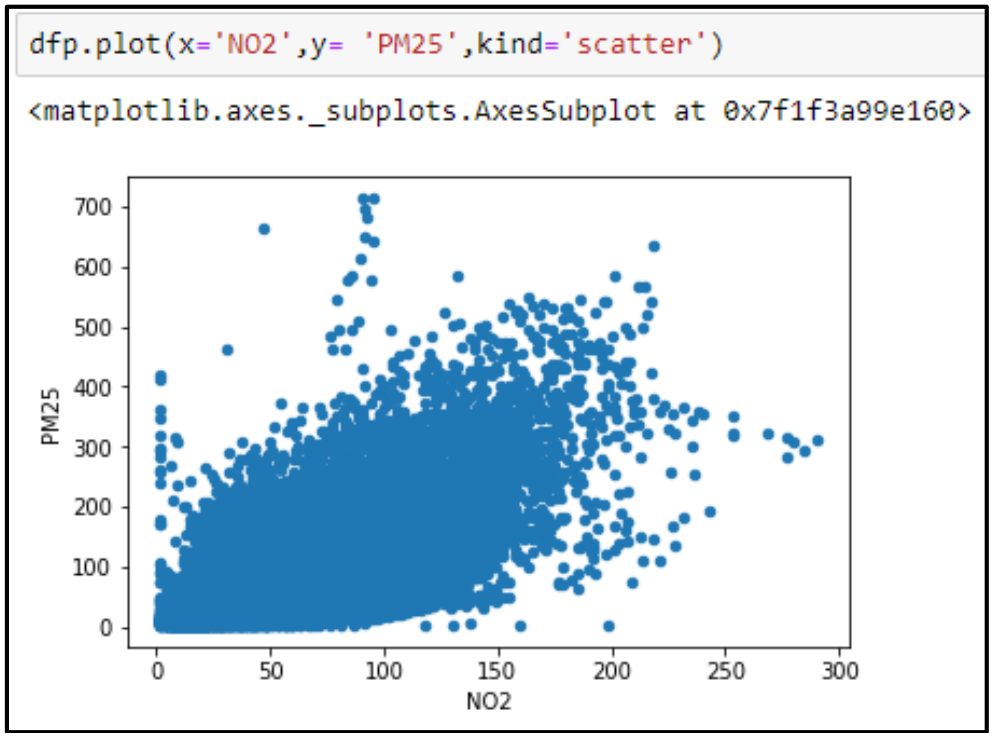


Fig. 8 Relationship between PM25 and NO2 data from 2013 to 2017 in all station

```
dfp.plot(x='SO2',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f3a4335f8>
```

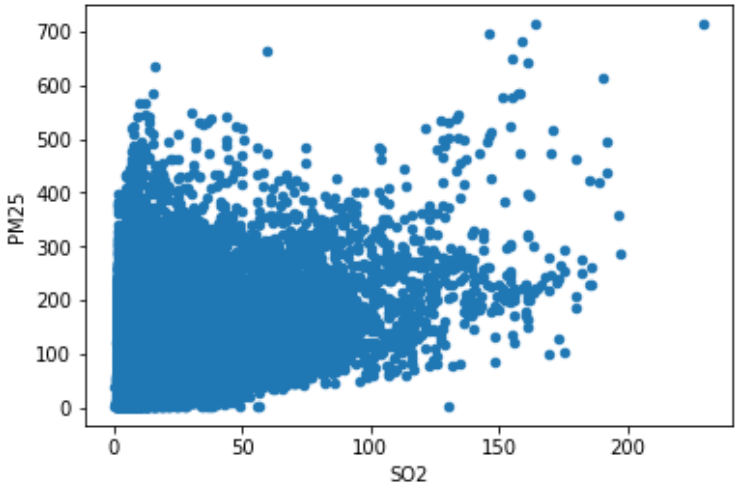


Fig. 9 Relationship between PM25 and wind direction data from 2013 to 2017 in all station

```
dfp.plot(x='CO',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f3a33ccf8>
```

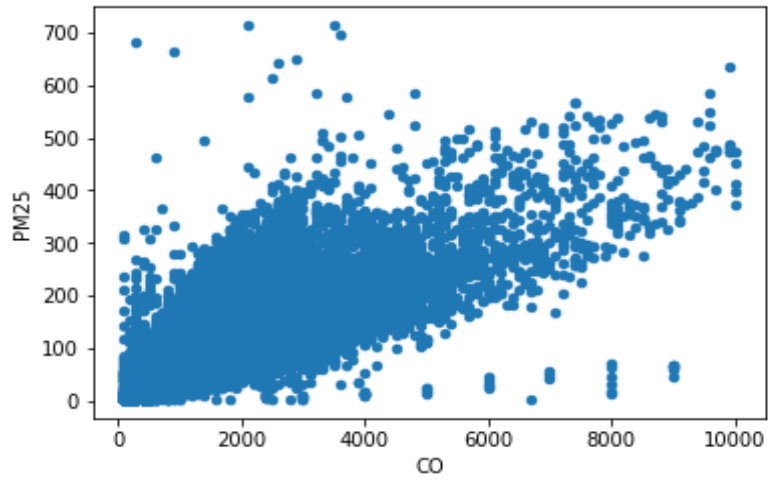


Fig. 10: Relationship between PM25 and CO data from 2013 to 2017 in all station

```
dfp.plot(x='O3',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f3a60a470>
```

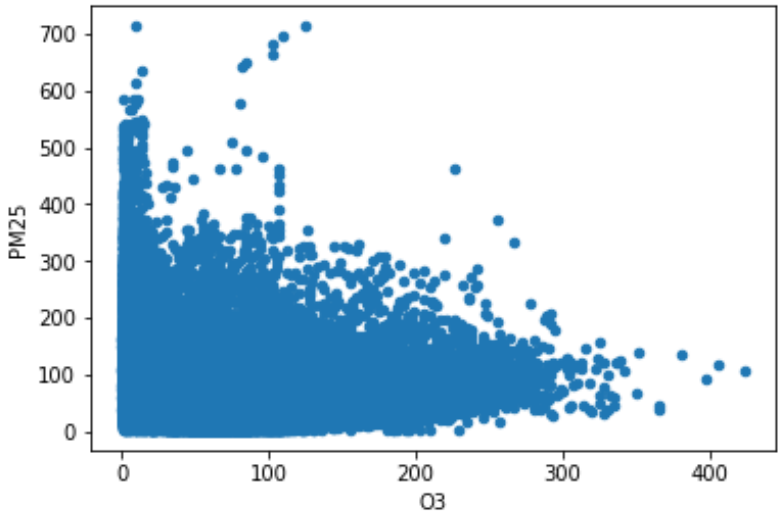


Fig. 11: Relationship between PM25 and O3 data from 2013 to 2017 in all station

```
dfp.plot(x='PM10',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f3a1a5240>
```

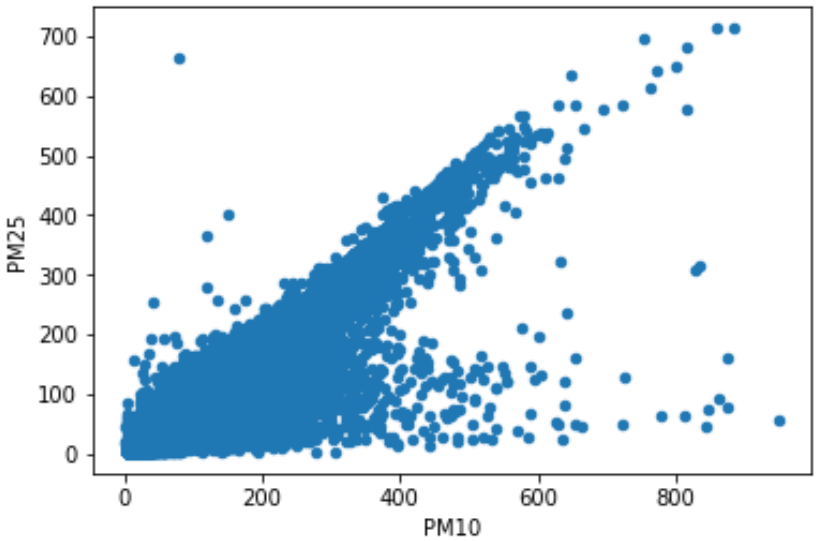


Fig. 12: Relationship between PM25 and PM10 data from 2013 to 2017 in all station

```
dfp.plot(x='PRES',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f3a13e588>
```

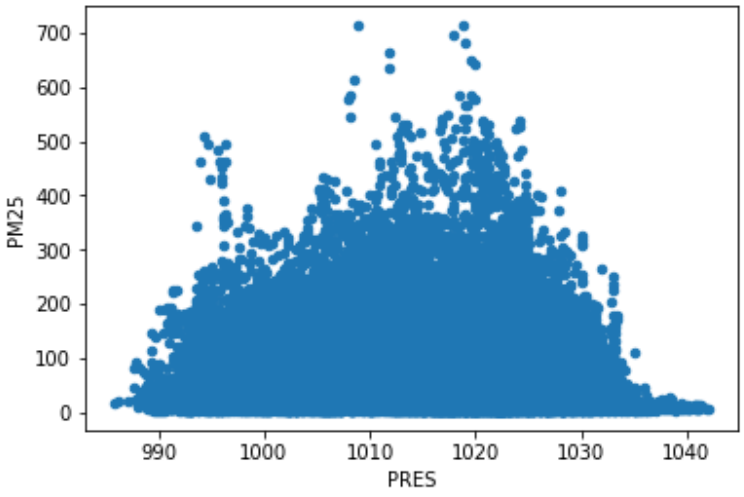


Fig. 13: Relationship between PM25 and PRES data from 2013 to 2017 in all station

```
dfp.plot(x='WSPM',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f3a0b2588>
```

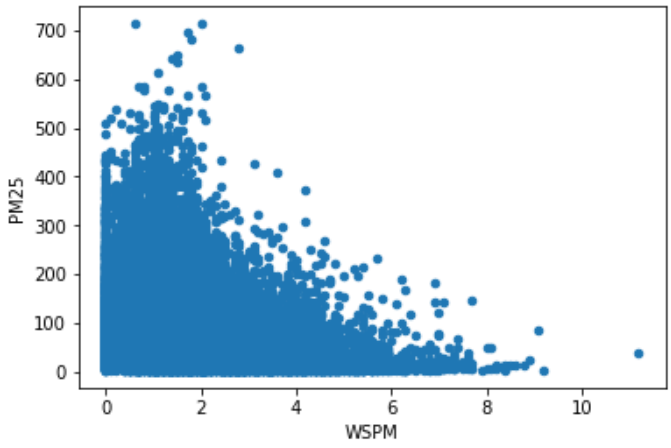


Fig. 14: Relationship between PM25 and WSPM data from 2013 to 2017 in all station

Investigation on data distribution is another important issue that we have to care about that. Actually based on above fining there is close relationship between two parameters PM10 and an input and PM25 as an output or Target. Fig. 15 to

Fig. 17 indicate on distribution of some critical parameters as three histograms. Distribution of PM25 and PM10 (Fig. 15) is very similar in sharing and frequency between 1 to 100. But the other ones have a little fluctuation to each other. Based on the

Fig. 17 most of the SO2 data are concentrated at the beginning of graph between 1 to 50 but NO2 distributed from 1 to 250 and it shows their relationship is not very clear and need more investigation.

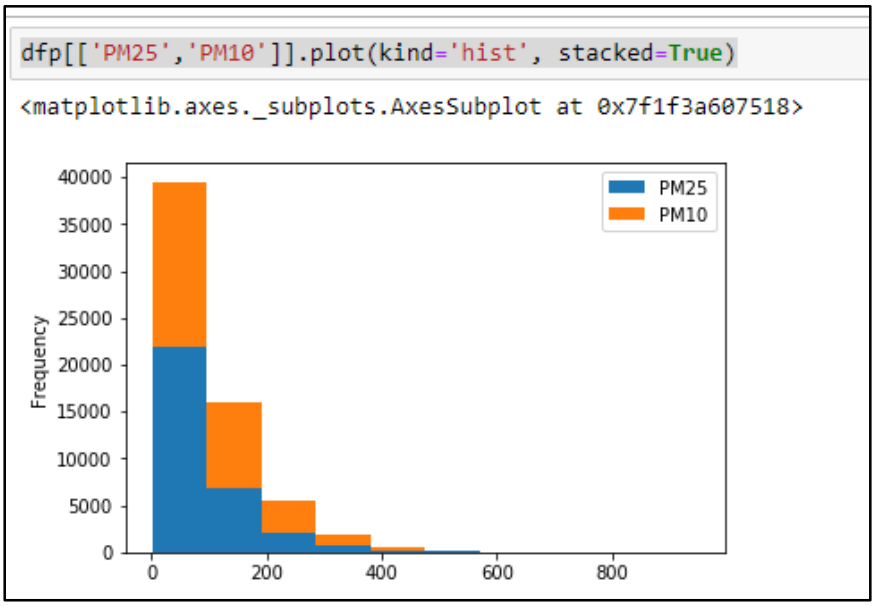


Fig. 15: distribution of PM25 vs PM10

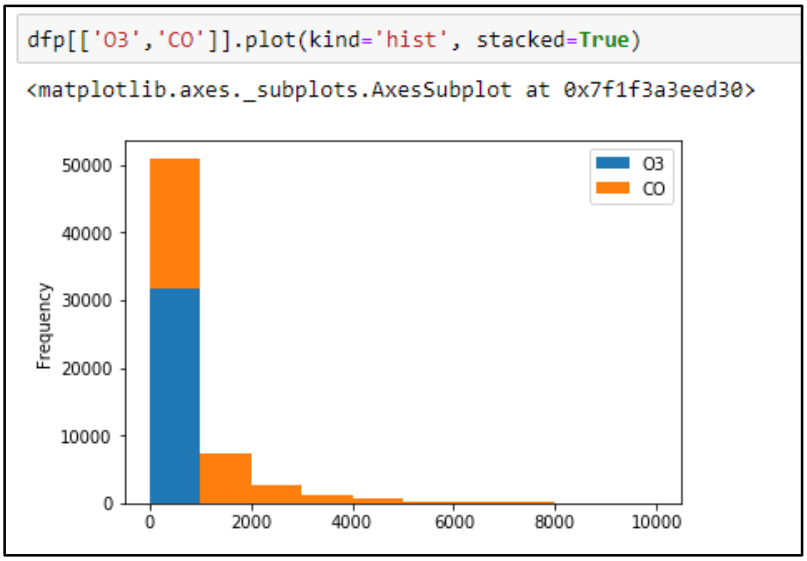


Fig. 16: distribution of O3 vs CO

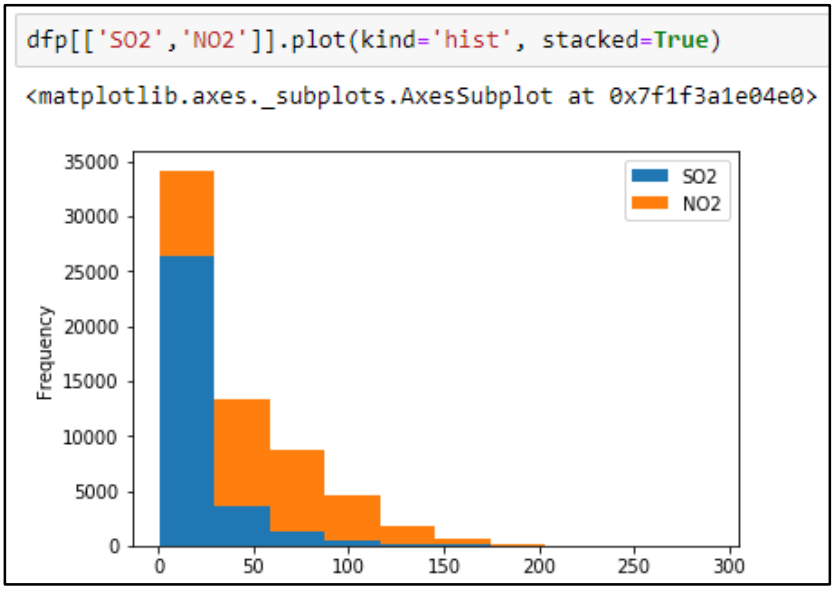


Fig. 17: distribution of SO2 vs NO2

2.4 VERIFYING DATA QUALITY

All 12 station data was loaded to Python (Fig. 19) shows about 420000 rows. Also Fig. 19 indicates the data audit in all 12 station from 2013 to 2017, more than 95 percent of data are valid and they can be used for modeling steps. In this phase, there are a few null values among air pollution data (Tab. 6).

The appended file includes more than 415000 valid row data for all 12 stations in Beijing.

Missing or null data are very vital and before any simulation, they must be investigated precisely. Among all data about 2 to 4 percent of data are missing. Tab. 6 shows the missing and complete data in all stations for all input and target data. Co and O3 are data that have the most null data (about 4 percent of all data).

By more investigation on null value, PM25 as an Target must not include the Null or NaN values. By dropping Null value in PM25 and other parameters, the size of data is about 412029 (df25). Also, by dropping all null in all input data the size is about 382168 (dfn), so it seems to better the dfn model because we lost only few data by dropping all Null values.

```
df.count()
```

```
420768
```

```
# Drops a row if a value from a particular row is missing. Two rows are dropped.
df25=df.na.drop(subset="PM25")
```

```
df25.count()
```

```
412029
```

```
dfn=df.na.drop(subset=["PM25",'PM10','O3','NO2','CO','SO2'])
```

```
dfn.count()
```

```
382168
```

```
df=df.na.drop()
```

```
df.count()
```

```
382168
```

Tab. 6 Number of Missing data in all station

On the other hand Fig. 18, and Fig. 19 show the all station mean values for all parameters after removing the Null values. Fortunately, the size of all data are the same about 382168 number.

```
df.groupBy('station').mean('PM25','PM10','SO2','NO2','CO').show()
```

```
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|      station|        avg(PM25)|         avg(PM10)|          avg(SO2)|          avg(NO2)|           avg(CO)|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|    Changping|  70.312328264129| 94.08640188488725|15.061356751629388| 44.31903533857593|1151.7164407453872|
| Aotizhongxin|81.86363036303632|109.12068521137829| 17.16720528052805| 58.97633191890618|1256.5747289014616|
|Wanshouxigong|84.23851013183594|111.91869201660157|17.152568701171877| 55.30718125915527|1362.5316467285156|
|       Wanliu|81.98145851015212|108.76586146112164|17.700088215708004| 64.42928958020501|1293.4501534242997|
|       Dongsi|84.93315643747118|110.99191443074693|16.994465686597668|53.717313985760434| 1316.524391851803|
|       Shunyi|79.40072530966417|  99.3805458038021| 14.14410478903093| 44.83848316221766| 1201.778267205405|
| Nongzhanguan|84.72078275049827|109.17592861025548|18.560198103521167|  58.2134567826297|1327.7843812284834|
|      Tiantan|81.74984014858569|105.62786590749931|14.140742319520141| 53.21308315318332|1297.4341868891393|
|     Dingling|66.51251836708619| 84.52780297706508|11.944167252283906|27.267578100044716| 916.5472113971763|
|      Huairou|70.28566923173962|  92.5257915983348| 12.27582313611707|32.633827425255454|1032.3982906522012|
|      Gucheng|83.86565345803594|119.08500184592667|15.095190007383708|55.583878547255715|1325.5096911149396|
|     Guanyuan| 83.1010507392369|108.97455909245883|17.263085069584356| 57.69859070762173|1265.7097604066578|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
```

Fig. 18: Data import in all stations

```
df.describe('PM25','PM10','NO2','SO2','O3','CO').show()
```

```
+-------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|summary|             PM25|             PM10|              NO2|              SO2|               O3|               CO|
+-------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|  count|           382168|           382168|           382168|           382168|           382168|           382168|
|   mean|79.43238314039898|104.57383742228552| 50.57006844345943|15.634813954072515|57.37667572690541|  1229.94056278914|
| stddev|80.15490109533691| 91.37944579945759|35.062085641153246|21.306102815726863| 56.7090126243123|1157.1514763728323|
|    min|              2.0|              2.0|              2.0|            0.2856|           0.2142|              100|
|    max|            844.0|            999.0|            290.0|            500.0|           1071.0|            10000|
+-------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
```

Fig. 19: Data description in all station , includes: Min, Max, Mean, std,….

Fig. 20: Data quality in stations after removing the missing data in all rows

3 DATA PREPARATION

Data preparation is one of the most important and time-consuming aspects of data mining. In fact, it is estimated that data preparation usually takes most of a project's time and effort. However, spending suitable attempt on earlier business understanding and data understanding phases can lessen this overhead, but it still needs to expend a good amount of effort preparing and packaging the data.

3.1 SELECT THE DATA

In this section the relevant data can be selected correctly and based on that there are two different approaches to choose the best data as below:

- **Selecting items (rows)**
  As the data properties were discussed in previous chapters, this dataset includes 12 stations in Beijing area and this stations measure and collect the air pollutants concentration and climate parameters from 2013 to 2014. Each station includes more than 35000 rows data they were imported to Python and appended as a unique dataset. About 415000 rows after dropping the missing data were selected to make a model (Fig. 21 to Fig. 23).

- **Selecting attributes or characteristics (columns)**
  Two categories of data were selected for modelling which include air pollution data; PM2, PM10, CO2, NO2, CO, C and climate data; Temperature, Pressure, Wind direction and speed, DEWP.

  For more investigation on the efficiency of input data, two kinds of models with different input data will be applied in the prediction phase. In the first model, only climate data such as temperature, pressure, humidity, and wind speed (or wind direction) are inputs (Fig. 24). Meanwhile, in the second model all data will be applied as an input (Fig. 25). Day, year and Rain data were removed from selected data.

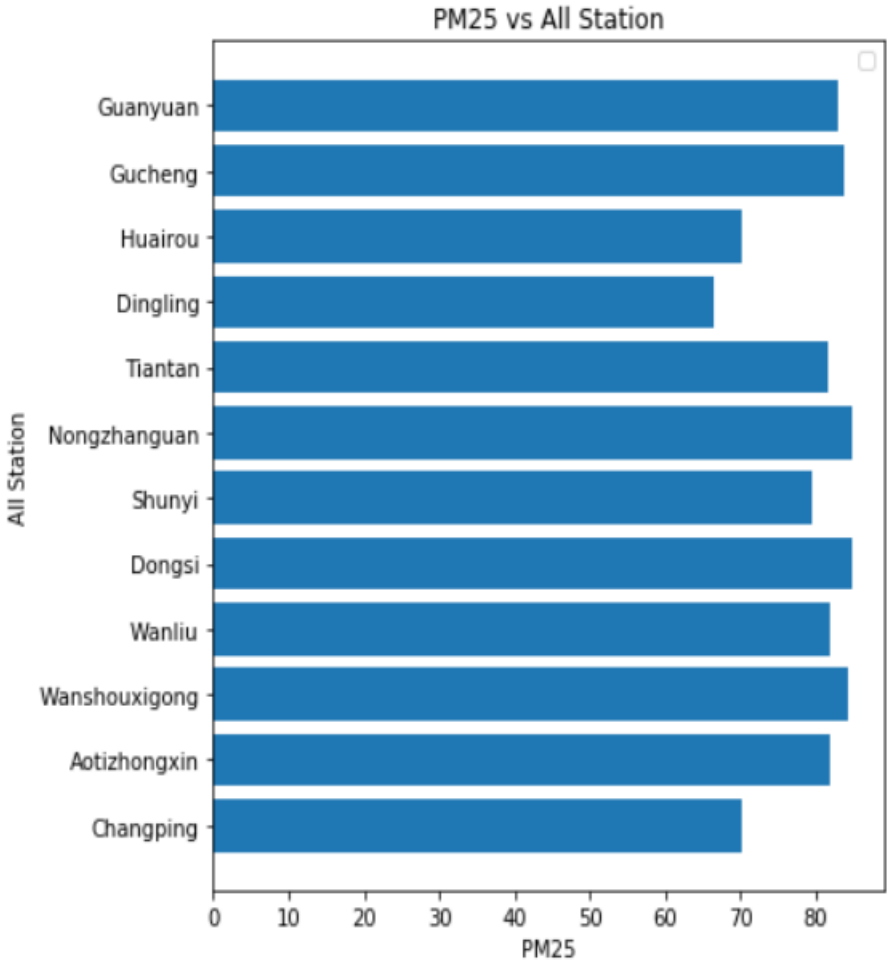


Fig. 21 appended data vs PM25 in all station

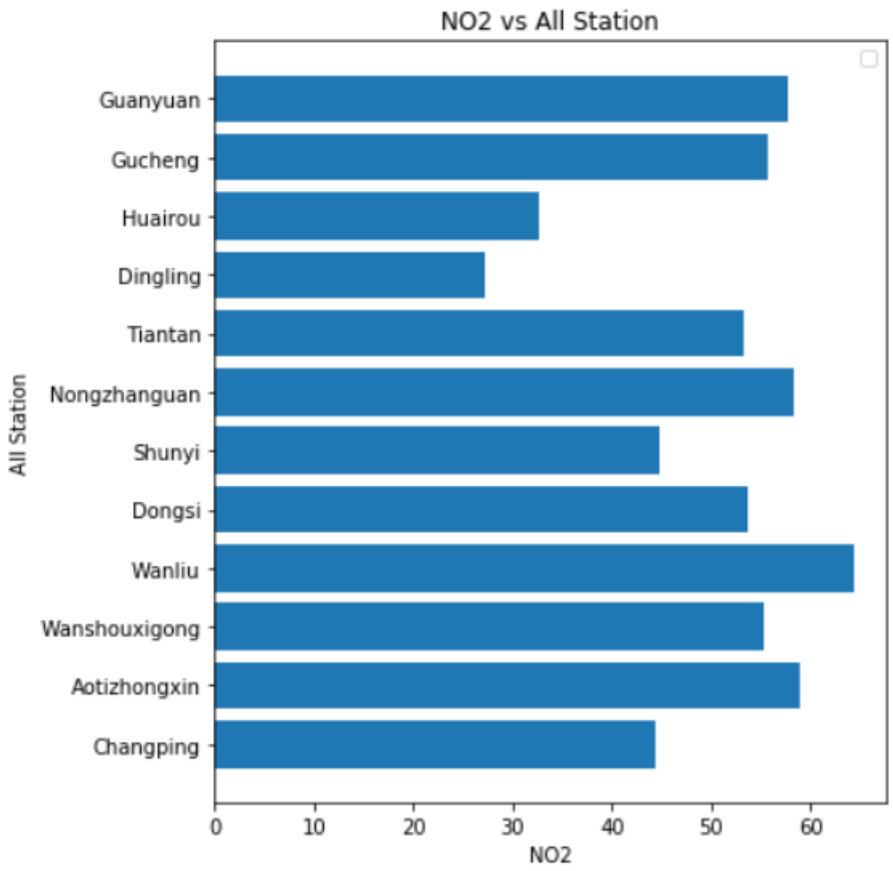


Fig. 22: appended data vs NO2 in all station

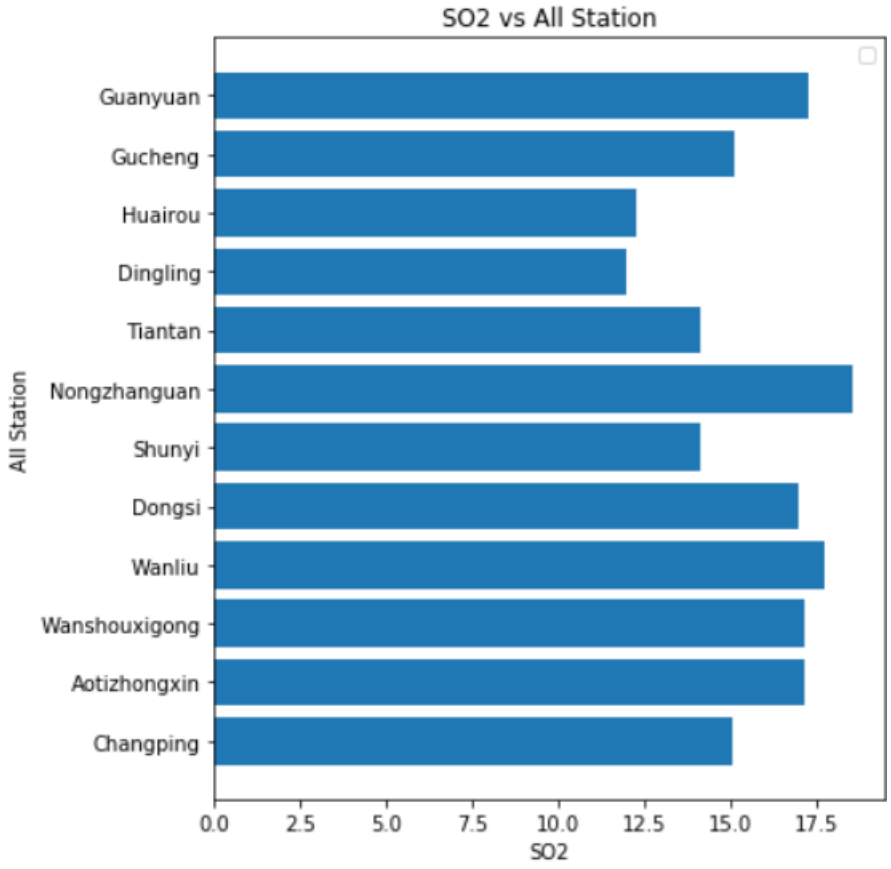


Fig. 23 appended data vs SO2 in all station

```
#average of properties in each station
dfg=df.groupBy('station').mean('PM25','PM10','SO2','NO2','CO')
dfg.show()
```

```
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|      station|        avg(PM25)|        avg(PM10)|         avg(SO2)|         avg(NO2)|          avg(CO)|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|    Changping|  70.312328264129| 94.08640188488725|15.061356751629388| 44.31903533857593|1151.7164407453872|
| Aotizhongxin|81.86363036303632|109.12068521137829| 17.16720528052805| 58.97633191890618|1256.5747289014616|
|Wanshouxigong|84.23851013183594|111.91869201660157|17.152568701171877| 55.30718125915527|1362.5316467285156|
|       Wanliu|81.98145851015212|108.76586146112164|17.700088215708004| 64.42928958020501|1293.4501534242997|
|       Dongsi|84.93315643747118|110.99191443074693|16.994465686597668|53.717313985760434| 1316.524391851803|
|       Shunyi|79.40072530966417| 99.3805458038021| 14.14410478903093| 44.83848316221766| 1201.778267205405|
| Nongzhanguan|84.72078275049827|109.17592861025548|18.560198103521167| 58.2134567826297|1327.7843812284834|
|      Tiantan|81.74984014858569|105.62786590749931|14.140742319520141| 53.21308315318332|1297.4341868891393|
|     Dingling|66.51251836708619| 84.52780297706508|11.944167252283906|27.267578100044716| 916.5472113971763|
|      Huairou|70.28566923173962| 92.5257915983348| 12.27582313611707|32.633827425255454|1032.3982906522012|
|      Gucheng|83.86565345803594|119.08500184592667|15.095190007383708|55.583878547255715|1325.5096911149396|
|     Guanyuan| 83.1010507392369|108.97455909245883|17.263085069584356| 57.69859070762173|1265.7097604066578|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
```

Fig. 24: the first model for PM25 prediction and relevant inputs (air pollutant data). Target data is in red region

```
#average of CLIMATE properties in each station
dfg2=df.groupBy('station').mean('TEMP','PRES','DEWP','WSPM','pm25')
dfg2.show()
```

```
+-------------+------------------+------------------+------------------+------------------+------------------+
|      station|         avg(TEMP)|         avg(PRES)|         avg(DEWP)|         avg(WSPM)|         avg(pm25)|
+-------------+------------------+------------------+------------------+------------------+------------------+
|    Changping|13.401676579712658|1007.9940087512003| 1.135298797466421|1.8657568617851341|  70.312328264129|
| Aotizhongxin|13.775610750644972|1011.8003847766261|3.2411063963539273|1.7204714757189838|81.86363036303632|
|Wanshouxigong|13.773187219529774|1011.5600415548229| 2.607440185546893|1.7559844970702887|84.23851013183594|
|       Wanliu|14.020991264246053|1010.7487723336236| 3.831086374616461|1.5029705555918151|81.98145851015212|
|       Dongsi|13.623337899612281|1012.8790166898078|2.2777935262706968|1.8725097237787321|84.93315643747118|
|       Shunyi|12.787636529806317|1013.6480448152555|1.5428727561767166|1.8423064184937326|79.40072530966417|
| Nongzhanguan|13.698473410997504| 1012.495573121202|2.5219876789273363| 1.858044935676724|84.72078275049827|
|      Tiantan| 13.63201256072133|1012.6077624354809|2.4954815333556724|1.8534147306883991|81.74984014858569|
|     Dingling|13.556273950663535|1007.7080069847617|1.4034114866159761| 1.850939117102149|66.51251836708619|
|      Huairou|12.276487150356086| 1007.805222919518| 2.064693452756403|1.6541062192506786|70.28566923173962|
|      Gucheng|13.926933343051434|  1008.79284781784| 2.632562146197413| 1.360549470834361|83.86565345803594|
|     Guanyuan|13.707450852837765| 1011.856881205573| 3.242757958032418| 1.719461922325861| 83.1010507392369|
+-------------+------------------+------------------+------------------+------------------+------------------+
```

Fig. 25: the second model for PM25 prediction and relevant inputs (weather data). Target data is in red region

3.2 CLEAN THE DATA

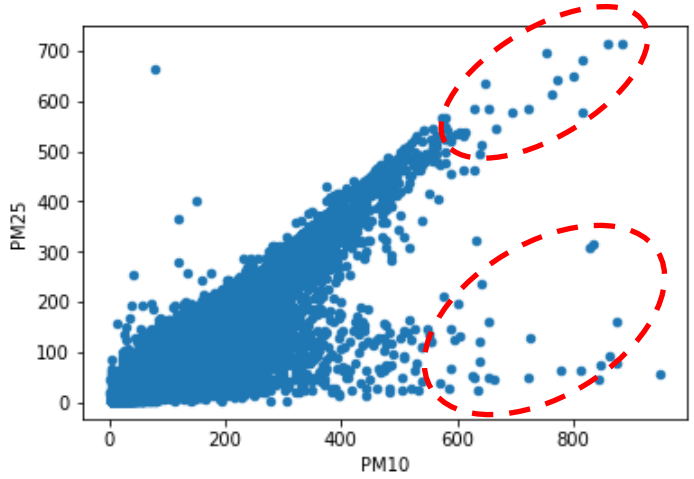
Cleaning the data involves taking a precise and serious investigation on the missing or null data in the above dataset. As *Fig. 26*, *Fig. 28*, and *Fig. 31* show outliers or extreme values that they are clear among all data in PM25, PM10 , CO and NO2. However, they could be less than 5 percent among all data. Most Outliers, Extreme and Null values belong to PM10, PM25, and CO and NO2. In Action column, the Outliers and Extreme values can be optimized. Coerce strategy was chosen for this issue, because of huge among of correct data we can change the outliers to precise data. On the other hand, two input data; TEMP and DEWP exclude any outliers' data.

Fig. 27 show the effect of removing the outliers from PM10 and PM25 as a using filter function and all data more than 600 and 400 for PM25 and PM10 respectively was removed. Also by using right threshold for NO2 and CO all outliers which were determined by red line, were omitted in *Fig. 30*.

However, based on programming in Python max and min quantile were chosen for mentioned parameters. **Error! Reference source not found.** to **Error! Reference source not found.** reflect the impact of correct data on the cross plots. However, the other data remains as before and all of them will be used for prediction models.

```
#outliers
dfp.plot(x='PM10',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5d8dbe0>
```



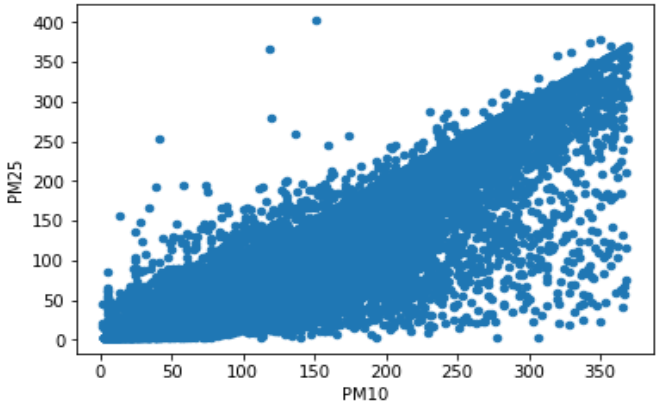

```
dfA.count()
```

```
31815
```

Fig. 26: Outliers in the scatter plot ;PM10 and PM25

```
dfpo1=dfA.filter(((dfA['PM25'] < 600) & (dfA['PM10'] < 370)))
dfpo=dfpo1.toPandas()
dfpo.plot(x='PM10',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5c08908>
```




```
dfpo1.count()
```

```
31224
```

Fig. 27:removing the Outliers in the scatter plot ;PM10 and PM25

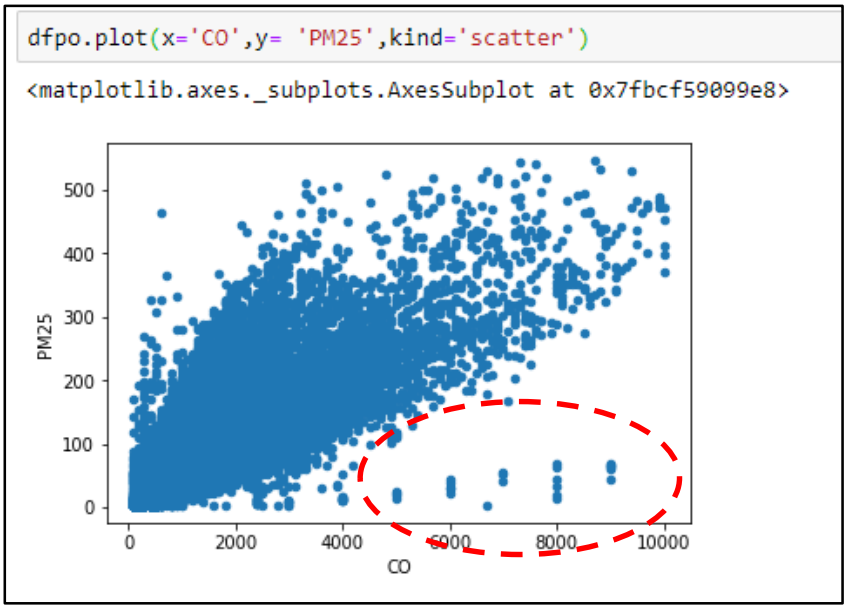


Fig. 28: Outliers in the scatter plot ;CO and PM25

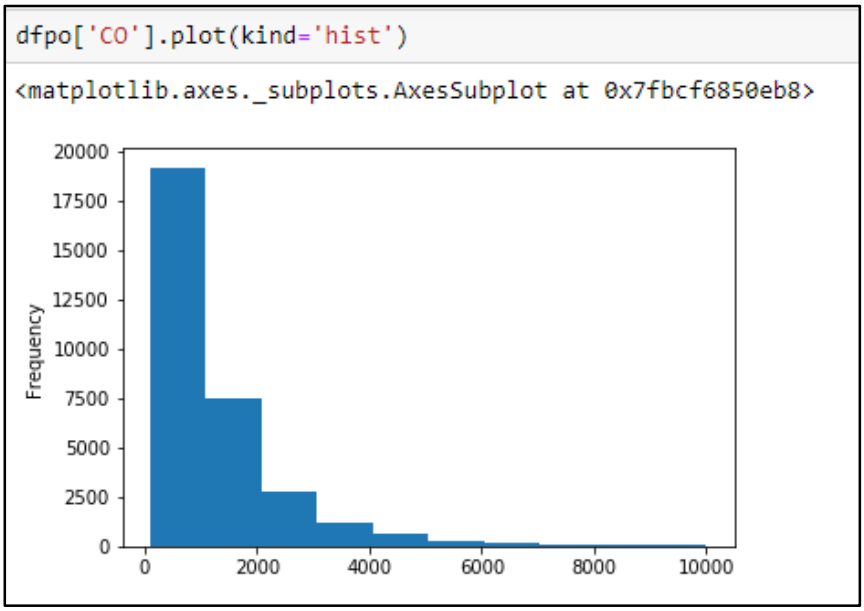


Fig. 29: distribution of CO

```
dfpo=dfA.filter(((dfA['PM25'] < 600) & (dfA['CO'] < 5000)))
dfpo=dfpo.toPandas()
dfpo.plot(x='CO',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5c098d0>
```

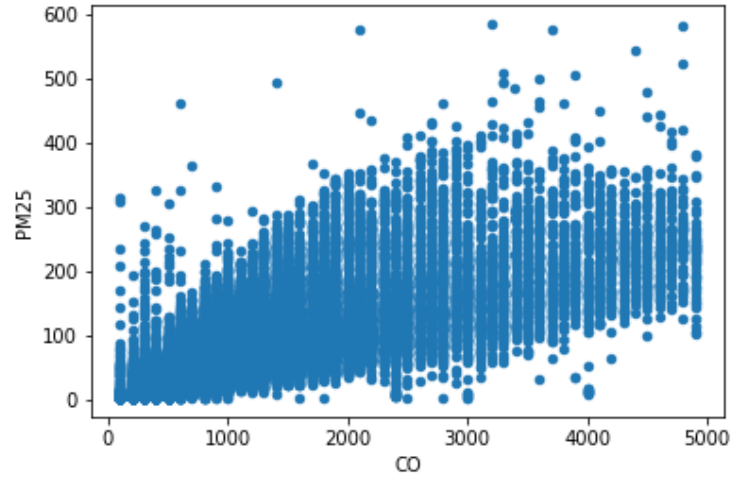


Fig. 30:removing the Outliers in the scatter plot ;CO and PM25

```
dfpo.plot(x='NO2',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5cad198>
```

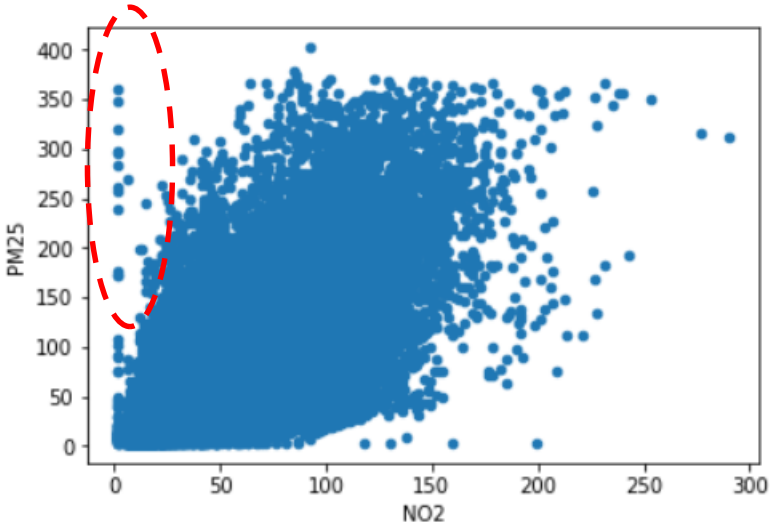


Fig. 31: Outliers in the scatter plot ;NO2 and PM25

By using the threshold for outliers in NO2, CO, PM25 and PM10, all data was filtered and based on below code programming in PySpark ‘dfn’ is the final dataset which is excluded from Null data and Outliers.

```
dfn=df.filter(((df['PM25'] < 600) & (df['CO'] < 5000) & (df['PM10'] < 370)))
dfn.count()
```

```
371515
```

3.3 CONSTRUCT THE DATA

In this stage, we mention the analysis results about the different PM25 concentration values for each station in Beijing. Fig. 49 shows PM25 distribution in all station and displays the duration of three PM25 concentration ranges PM25 <100, 100<PM25 <300, 600<PM25 µg/m3. However, based on the PM25 distributions in Fig. 32, 50 percent of data is less than 100 µg/m3 and this level could be good indicator of pollution boundary in the Beijing. By Drive application in PySpark, the new data as a flag was generated which is called pollution –index. However, this data will not be used as an input to model. Because it was calculated from target PM25.

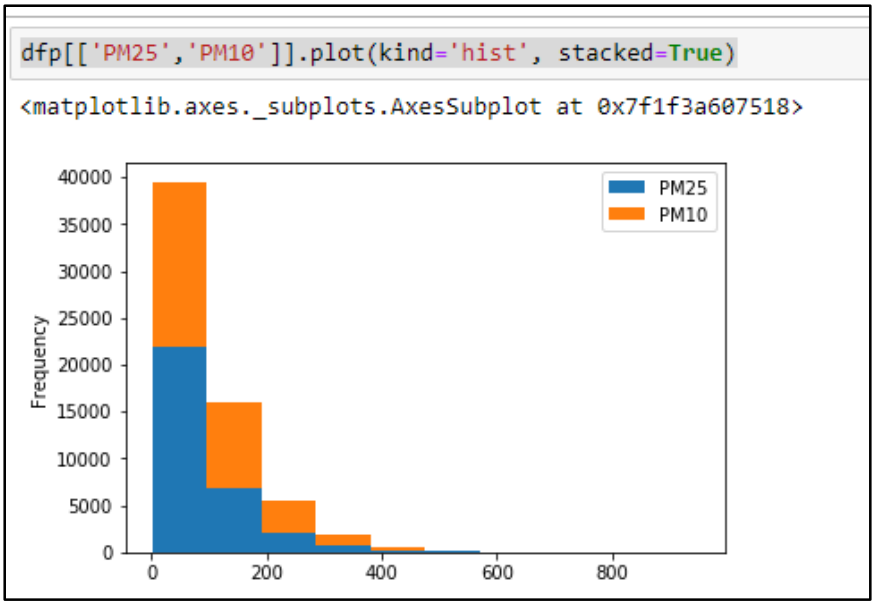


Fig. 32: distribution of PM25

Consideration outlier to all data

```
#after rmoving the null data
df.count()
```

```
382168
```

```
df1000=df.filter(df['PM25'] < 100)
df1000.count()
```

```
272554
```

```
df3000=df.filter((df['PM25'] > 100) & (df['PM25'] < 300))
df3000.count()
```

```
99167
```

```
df500=df.filter(df['PM25'] > 600)
df500.count()
```

```
188
```

3.4 INTEGRATE VARIOUS DATA SOURCES

As the data description was described in previous section, in this study, 12 station data were imported to model and because of high data size, all Null and outliers were deleted. Below code in PySpark explains the data of any station and import them as a unique file with applying all thresholds. However, working on unique dataset for missing value investigation is very effective. Based on aggregation function all 12 station in Beijing are integrated to each other. Average values for important factors such as NO2, PM25 and SO2 in all station in figure Fig. 35 to Fig. 36.

```
df.groupBy('station').mean('PM25','PM10','SO2','NO2','CO').show()
```

```
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|      station|        avg(PM25)|        avg(PM10)|         avg(SO2)|         avg(NO2)|          avg(CO)|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|    Changping|71.09974336541265| 94.65787077315701|14.958905587176204| 44.18208550745705|1152.3013445428255|
| Aotizhongxin|82.77361082632768|110.06039131194318| 17.375901409358608| 59.30583318645163|1262.9451453977408|
|Wanshouxigong|85.02413582402235|112.22345864661655|17.148603110917286|55.529560136986305|1370.3950306512274|
|       Wanliu|83.37471599100398|110.46461759631974| 18.376480570616696| 65.25878926575277|1319.3535125706724|
|       Dongsi|86.19429678848283|110.33674190837705| 18.53110660736606|53.699442802498275|1330.0691310760349|
|       Shunyi|79.49160200286961|  98.7370263066404|13.572038687514805| 43.90886473782605|1187.0639785927142|
| Nongzhanguan|84.83848298292484|108.99109577171902|18.689242012825694| 58.09717238740834|1324.3501978852855|
|      Tiantan|82.16491115828659|106.36367249833174| 14.367615106345372| 53.16264557400931| 1298.303317814839|
|     Dingling|65.98949686451802| 83.73972332015809|11.749649653404786|27.585466754360024| 904.8960728548954|
|      Huairou|69.62636686112984| 91.48269023244961|12.121553010210071| 32.49725021391175|1022.5545449140955|
|      Gucheng|83.85208902318554|118.86197849090333|15.366161622826057| 55.87107495348295|1323.9744229569558|
|     Guanyuan|82.93337203901532|109.02330301717915| 17.59094149754264| 57.90164251707599| 1271.294377232746|
+-------------+-----------------+-----------------+-----------------+-----------------+-----------------+
```

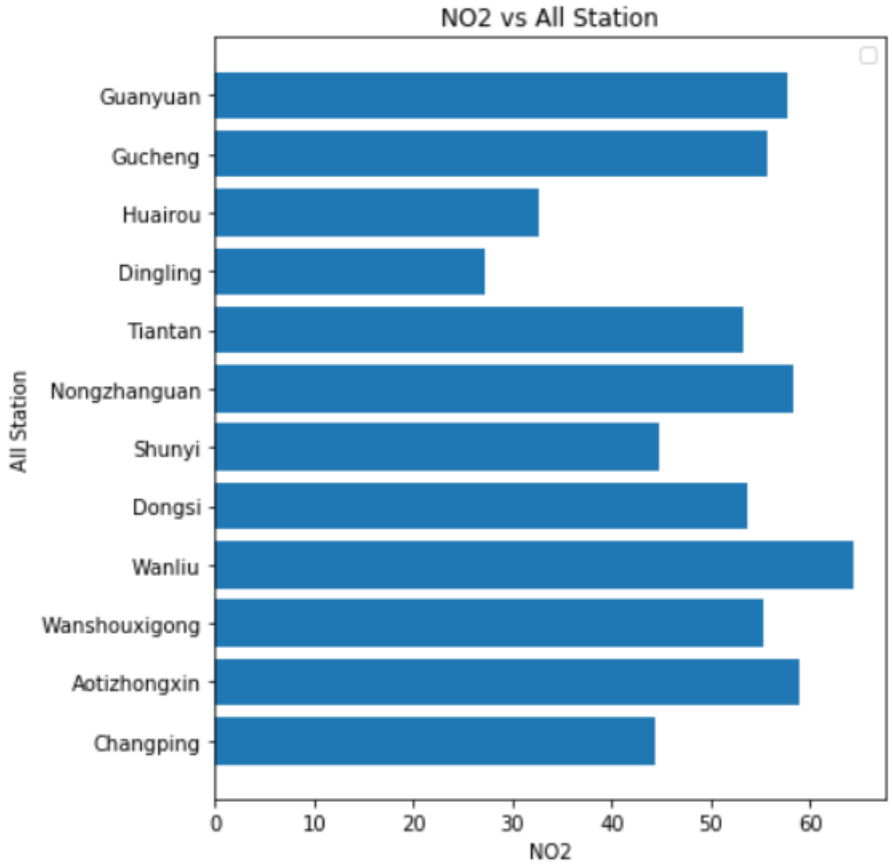


Fig. 33 *Bar chart of the average of NO2 in all station*

```
#'NO2 vs selected Station'
fig = plt.figure(figsize = (6, 7))
plt.barh(dfgp1['station'],dfgp1['avg(PM25)'])
plt.legend()
plt.title('PM25 vs All Station')
plt.xlabel('PM25')
plt.ylabel('All Station')
plt.show()
```

```
No handles with labels found to put in legend.
```

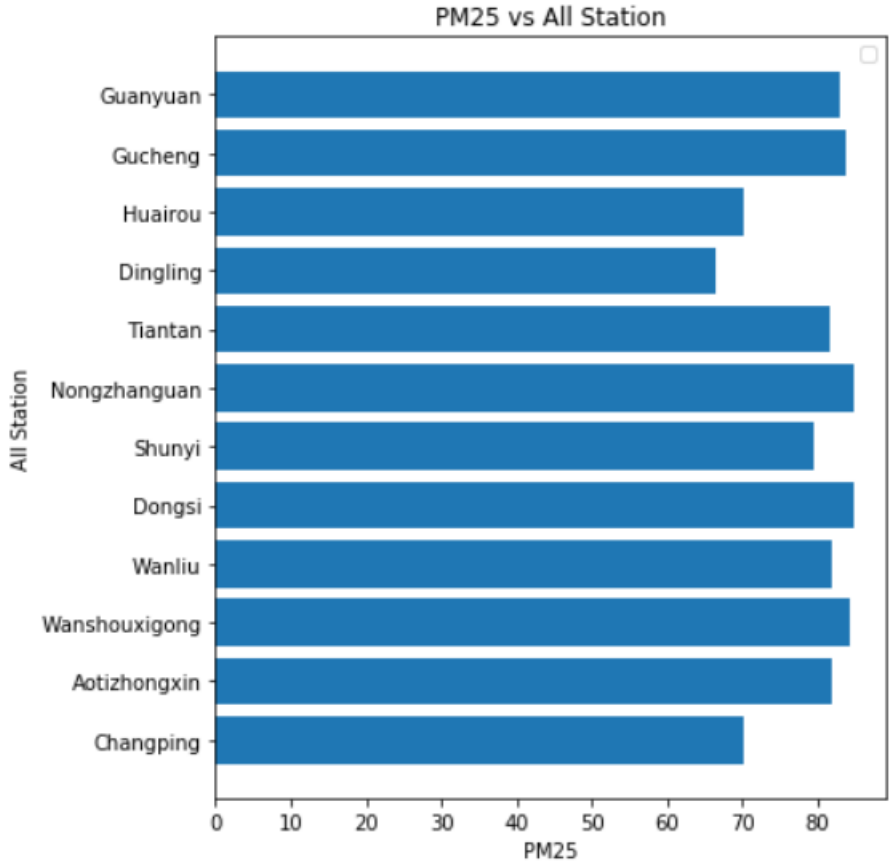


Fig. 34 *Bar chart of the average of PM25 in all station*

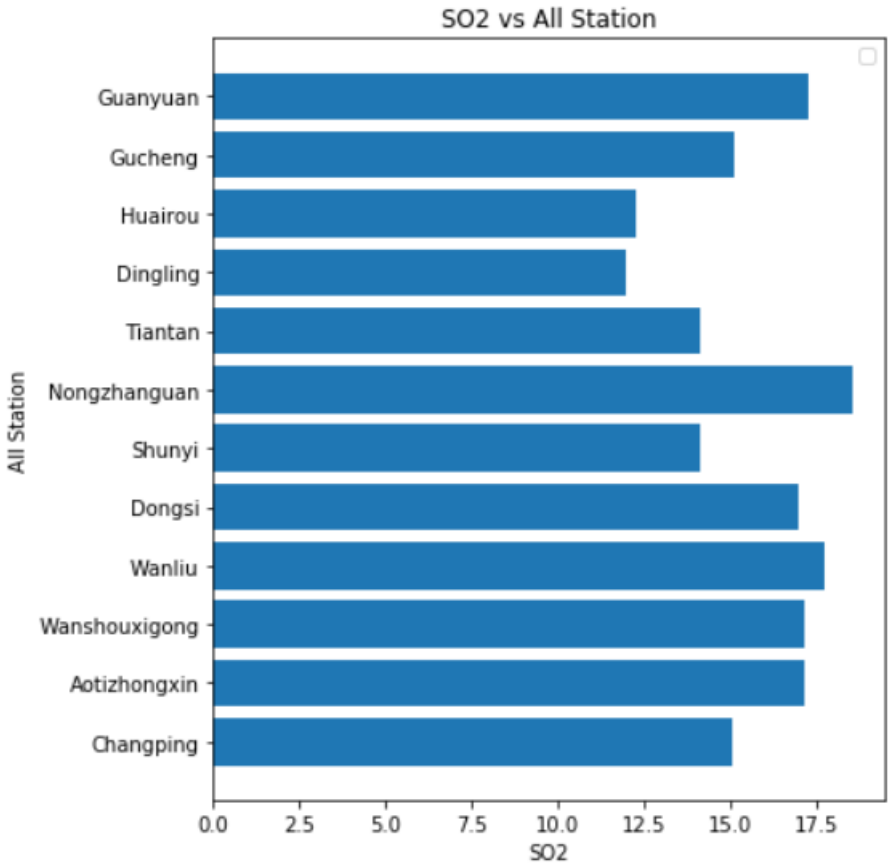


Fig. 35 Bar chart of the average of SO2 in all station

3.5 FORMAT THE DATA AS REQUIRED

The last part of data preparation is data formatting and in this stage the input and target must be checked whether certain techniques require a particular format or not. For example, all pollution data are numeric (integer and float data) and double but the pollution index is flag type data. However, some other data such as station are text or categorical data which will not to import to the model.

```
from pyspark.sql.types import (StructField,StringType,IntegerType,StructType)
dataType=dfn.dtypes
Schema_dataType=StructType([StructField('Parameters',StringType())\
                            ,StructField('format',StringType())])
dataType=spark.createDataFrame(dataType,schema=Schema_dataType)
dataType.show()
```

Tab. 7: Brief description of data properties

Parameters	format
year	int
month	int
day	int
hour	int
PM25	double
PM10	double
SO2	double
NO2	double
CO	int
O3	double
TEMP	double
PRES	double
DEWP	double
RAIN	double
wd	string
WSPM	double
station	string

4 DATA TRANSFORMATION

4.1 REDUCE THE DATA

In this section we must reduce some data which is not very important in modelling. in data mining workflow, we usually may involve huge amount of data that can potentially be used as and inputs. However, we have to spend plenty of time to find relationship between all variable step by step. For limitation of choices, the Feature Selection algorithm can be utilized to recognize the fields that are most important for a given analysis. Here for predicting the PM25 as a vital parameter in air pollution issue we must discover which inputs have a great impact on PM25. the below coefficient shows there is good correlation between PM25 and other data. Among them PM10 and CO have a proper correlation with PM25 rather than the parameters (Fig. 36 and Fig. 37, because of the memory the feature selection was run separately in two steps). Fig. 38 to Fig. 43 indicates the relationship between PM25 and other variables. However, the climate parameters such as Pressure (PRES) have a low correlation with PM25, but because of its important role to estimate the PM25, it will be used as an input to simulation.

```
  Specs          Score
3    CO   2.177728e+08
0  PM10   3.061565e+07
2   NO2   4.324122e+06
1   SO2   2.787069e+06
4    O3   4.883934e+05
5  TEMP   1.273665e+05
6  PRES   1.559138e+03
```

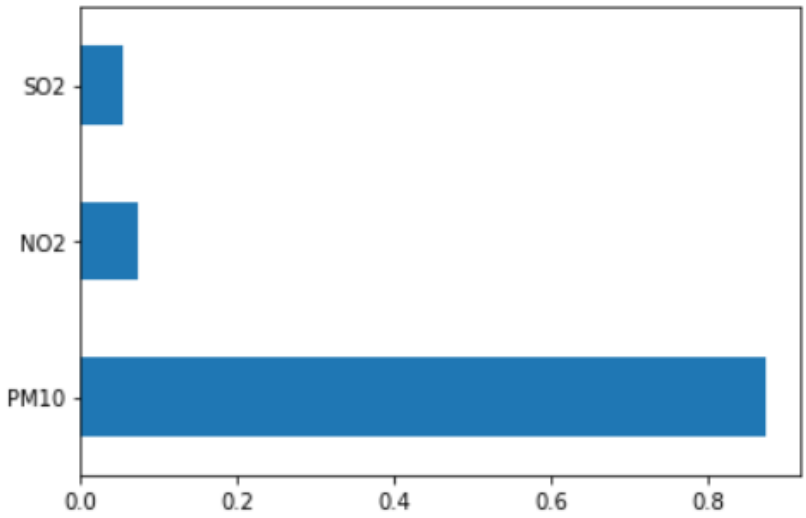


Fig. 36 Feature selection to choose the best input; PM25 , NO2 and SO2

```
[0.57666528 0.20496582 0.21836891]
```

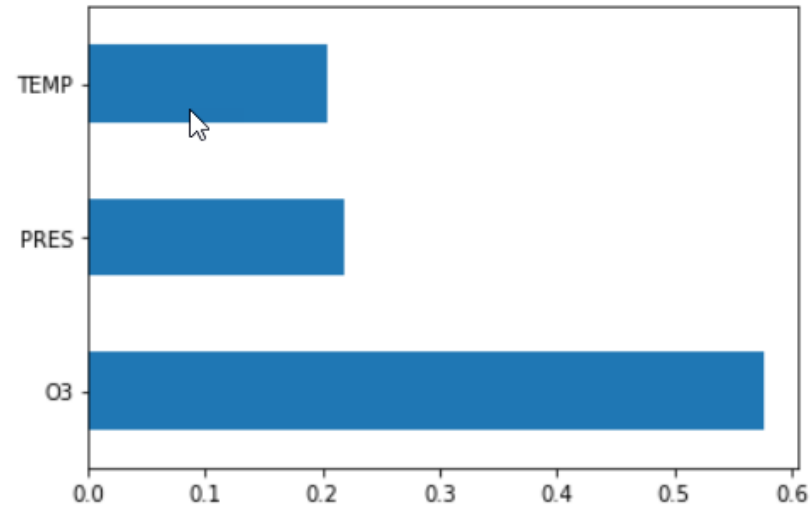
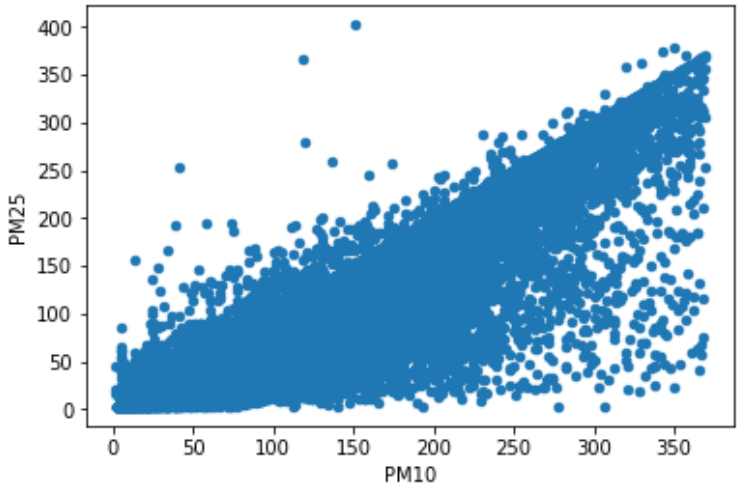


Fig. 37 Feature selection to choose the best input; O3, PRES and TEMP

```
dfpo1=dfA.filter(((dfA['PM25'] < 600) & (dfA['PM10'] < 370)))
dfpo=dfpo1.toPandas()
dfpo.plot(x='PM10',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5c08908>
```




```
dfpo1.count()
```

```
31224
```

Fig. 38Correlation between PM25 and PM10

```
dfpo=dfA.filter(((dfA['PM25'] < 600) & (dfA['CO'] < 5000)))
dfpo=dfpo.toPandas()
dfpo.plot(x='CO',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5c098d0>
```

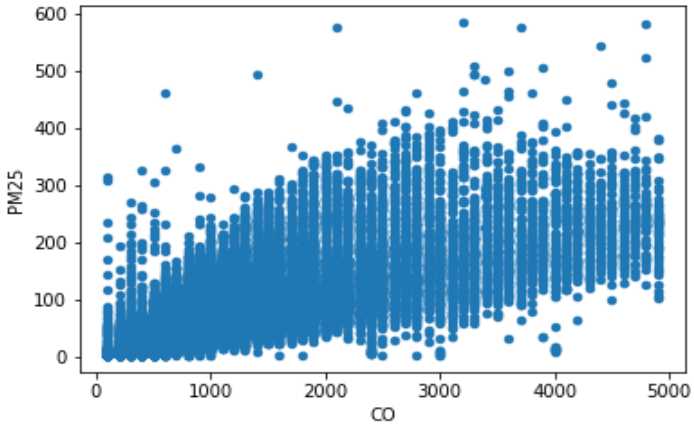


Fig. 39 Correlation between PM25 and CO

```
dfpo.plot(x='SO2',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5ec1400>
```

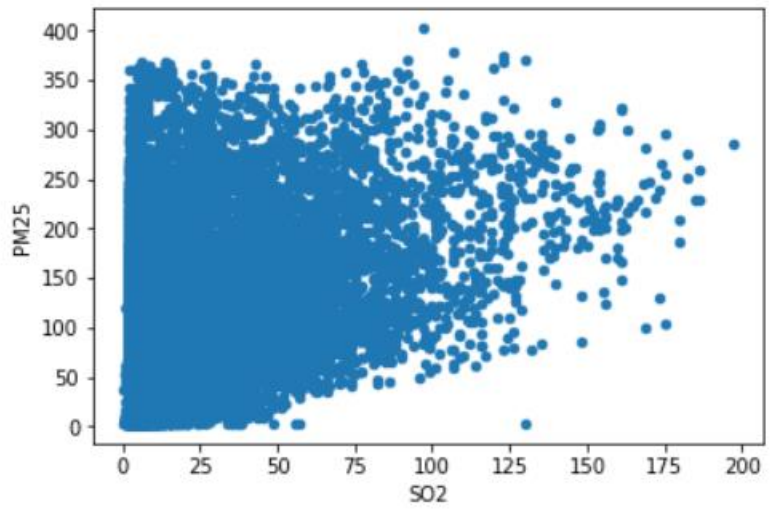


Fig. 40: Correlation between PM25 and SO2

```
dfpo.plot(x='NO2',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5cad198>
```

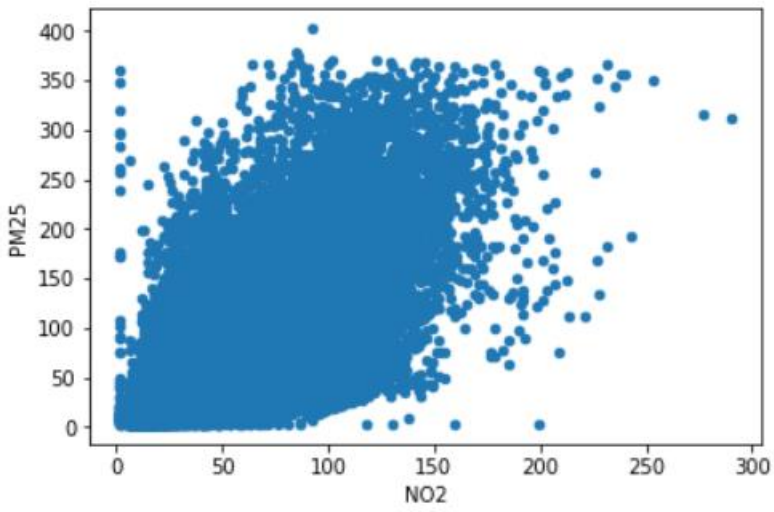


Fig. 41 Correlation between PM25 and NO2

```
dfpo.plot(x='O3',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf664b198>
```

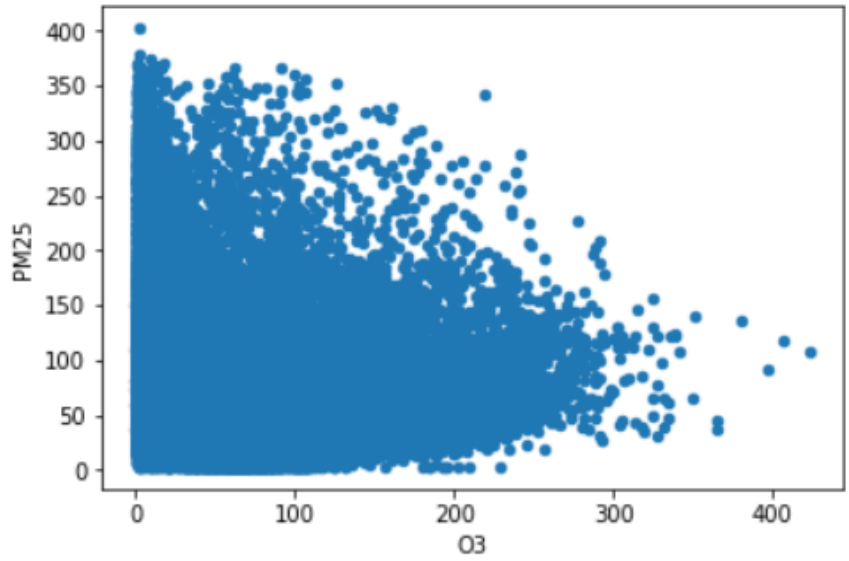


Fig. 42 Correlation between PM25 and O3

```
dfp.plot(x='PRES',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f3a13e588>
```

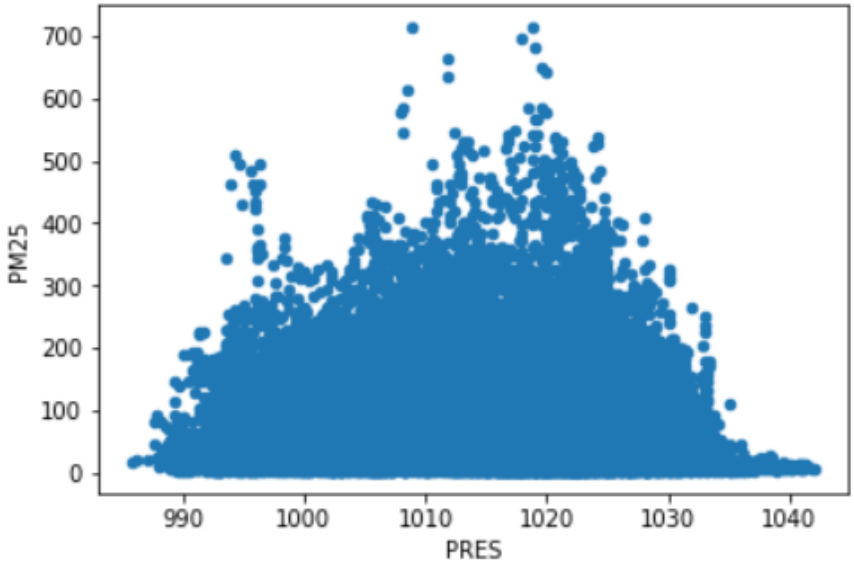


Fig. 43 Correlation between PM25 and pressure data

Based on station location and relationship of for all parameters in each stations, it needs to ignore some of the the station that their values are not inline to each other. Fig. 44 to Fig. 47 show distribution of average of PM10, PM25, SO2, CO, O3, NO2 in all station. Based on that only the values of 3 stations are not in line with others and finally the data in these station; Changding, Dinglig and Huairou were skipped from final data set. Fig. 48 indicates the aggregation of parameters in selected station and also Fig. 49 and Fig. 50 explain the NO2 and SO2 value in selected station. It indicates that now the data are consistent more.

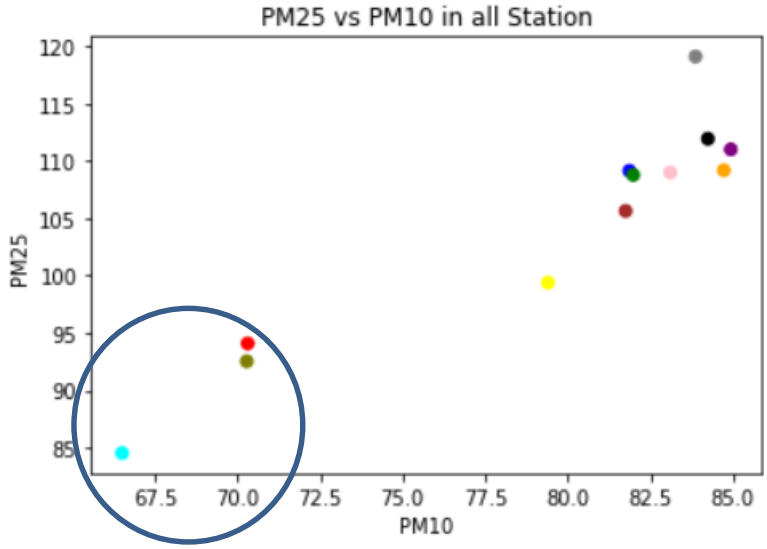


Fig. 44 PM10 and PM25 distribution in 12 station

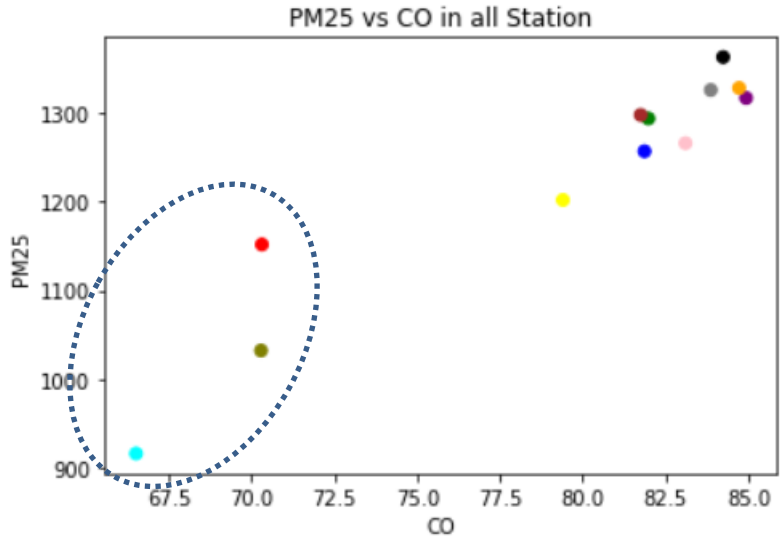


Fig. 45 CO and PM25 distribution in 12 station

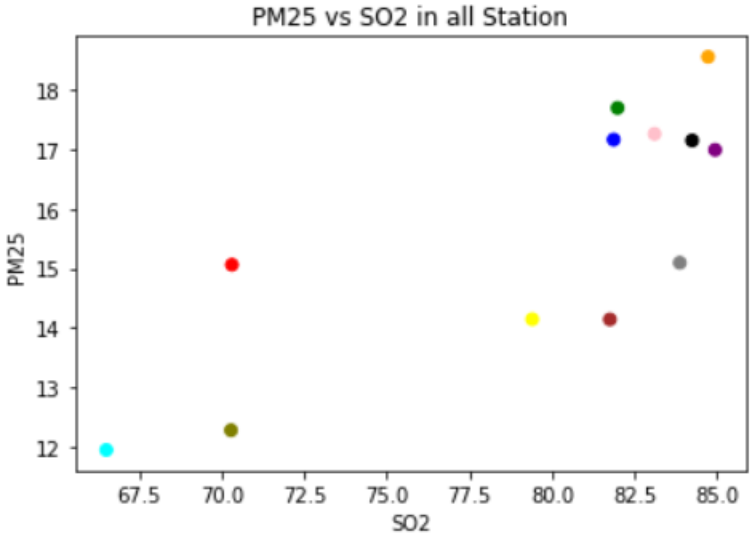


Fig. 46 SO2 and PM25 distribution in 12 station

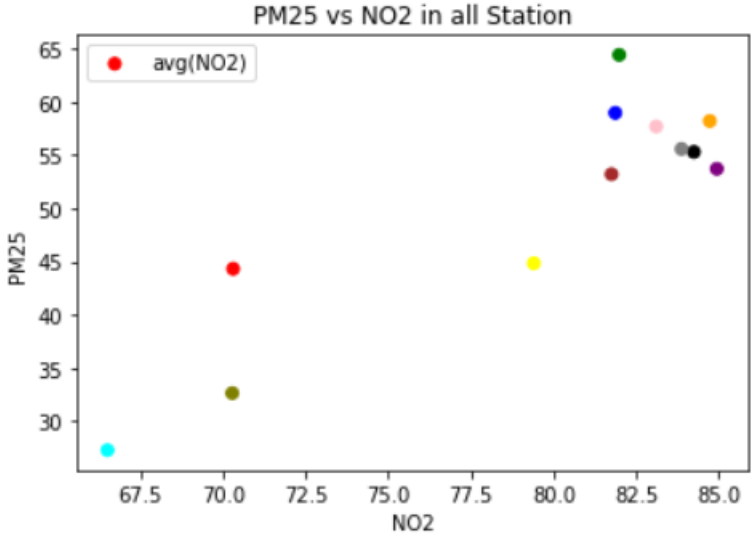


Fig. 47 SO2 and PM25 distribution in 12 station

```
dff=df.filter(((df['PM25'] < 600) & (df['CO'] < 5000) & (df['PM10'] < 370)))
dff.count()
```

```
385528
```

```
dfx=dff.filter((df['station']=='Aotizhongxin') |(df['station']=='Wanliu') |(df['station']=='Wanshouxigong')|
               (df['station']=='Dongsi')|(df['station']=='Shunyi')|(df['station']=='Nongzhanguan')|
               (df['station']=='Tiantan')|(df['station']=='Gucheng')
               |(df['station']=='Guanyuan')).select('PM25','PM10','CO','SO2','NO2','TEMP','PRES','DEWP','WSPM','station')
dfx.count()
```

```
287713
```

station	avg(PM25)	avg(PM10)	avg(SO2)	avg(NO2)	avg(CO)
Aotizhongxin	74.87202192638999	100.35584025058732	16.27717391714636	56.99160803861565	1133.2509631949883
Wanshouxigong	76.17865878200564	101.85097681042757	15.927248702991186	53.10110530001245	1232.9406483172854
Wanliu	75.03125176692318	100.56396104915973	16.925849483621583	62.57490663665595	1168.5365792366892
Dongsi	77.7518245061364	100.65843998162367	17.19145279484736	51.573507603866375	1204.1223009778828
Shunyi	73.47176492867062	92.47742709253882	13.202548257628846	43.25336102068608	1082.1802171443
Nongzhanguan	75.65853960013597	98.50510491023145	17.239130031247097	55.62347711203939	1176.15861685362
Tiantan	75.43294834679263	98.17333231753771	13.696896754432103	51.34997523238747	1185.18479353954
Gucheng	76.53120099178678	110.04636293196964	14.055162047968189	53.469156463884275	1201.7443049744304
Guanyuan	76.17102480112308	100.64751832787394	16.393985481601902	55.8848074350843	1158.0096084854156

Fig. 48 Aggregation of data average in selected station

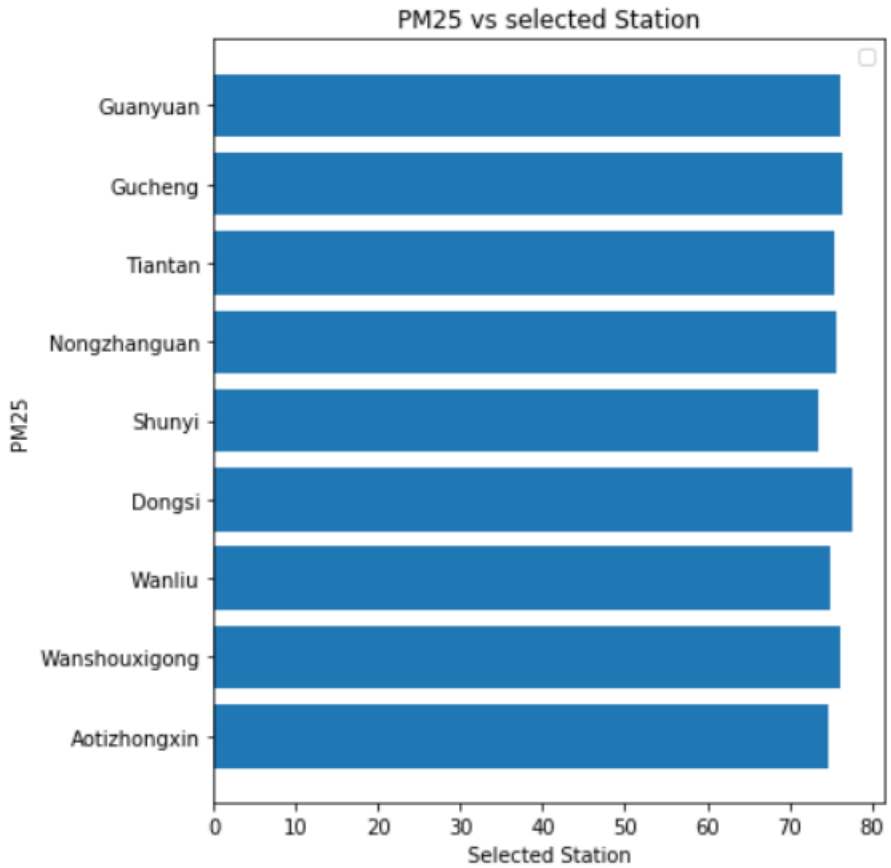


Fig. 49 Boxplot of PM25 distribution in 9 station

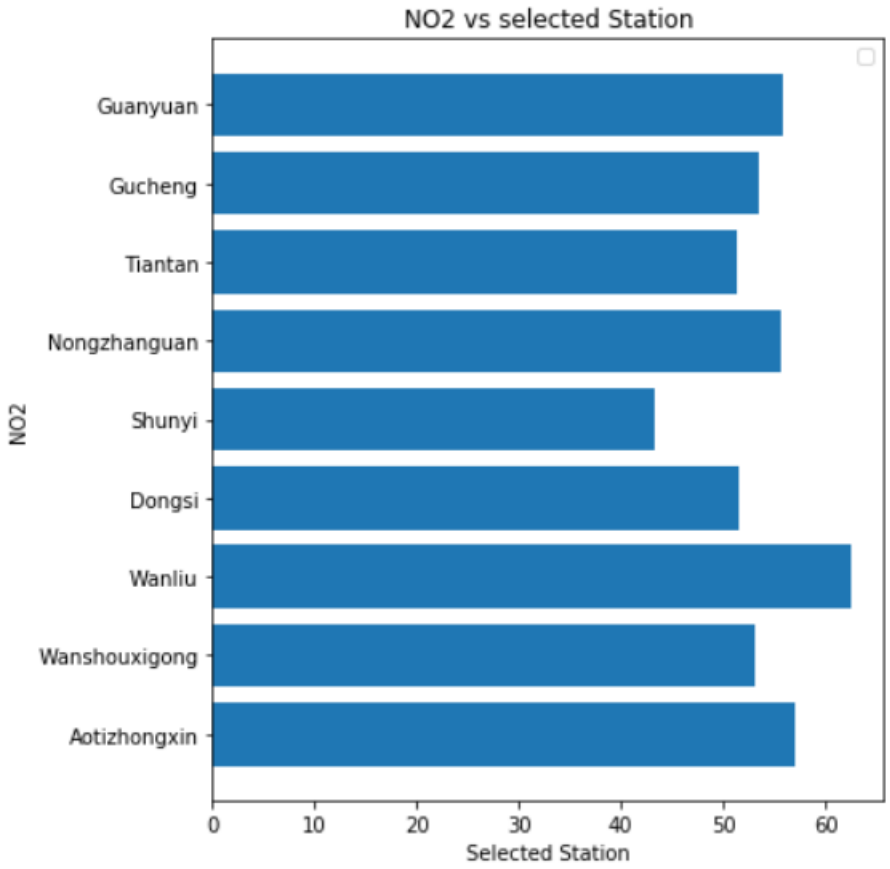


Fig. 50 Boxplot of NO2 distribution in 9 station

```
#'NO2 vs selected Station'
fig = plt.figure(figsize = (6, 7))
plt.barh(dfxp['station'],dfxp['avg(NO2)'])
plt.legend()
plt.title('NO2 vs selected Station')
plt.ylabel('NO2')
plt.xlabel('Selected Station')
plt.show()
```

4.2 PROJECT THE DATA

Pollution index is one of the new data which was generated as a new feature to classify the pollution as a different category. The objective is to have index with flag type from low, medium, high pollution. Based on the below category we are able to generate new flag data and consider it as a new value as a target except than PM25. most of pollution data distribution is low to medium.

```
df1000=df.filter(df['PM25'] < 100)
df1000.count()
```

```
272554
```

```
df3000=df.filter((df['PM25'] > 100) & (df['PM25'] < 300))
df3000.count()
```

```
99167
```

```
df500=df.filter(df['PM25'] > 600)
df500.count()
```

```
188
```

5 DATA-MINING METHOD(S) SELECTION

5.1 MATCH AND DISCUSS THE OBJECTIVES OF DATA MINING (1.1) TO DATA MINING METHODS

Air Pollution has been always considered as one of the most important problem about people heath in Beijing. Major air pollution matters, especially Fine particles (the matter with diameter less than 2.5 _µm) which is called PM25 are robustly associated with adverse health effects, including cardiac and respiratory morbidity and finally mortality. Research from an independent group even showed that air pollution causes 1.6 million deaths in 2016 (Rohde, 2015). So measuring, monitoring and prediction of this features like PM25 by using data mining approach can help to reduce the mortality rate.

Two different data will be used for model the target. The first group is combination of climate data with other related air pollutants and second group is only climate data such as Pressure, Humidity, Temperature….

ML teaches the computers to do what comes naturally to humans: learn from all experience. ML algorithms utilize computational approach to learn all knowledge from data without relying on a predetermined equation as a model. The algorithms adaptively enhance their performance as the number of samples available for learning rising up. Supervised and unsupervised are two kinds of Machine Learning methods who are applied a lot. Supervised learning entails learning a mapping between a different relevant input data and a target and applying this mapping to predict target (PM25) in future. Unsupervised ML focus on dataset that the target is not available and the goal is looking for the relationship between input data (Dayan, P. ,1999).

There are different types of the methodology to categorize the data mining technique but base of PYTHON system, Data mining modeling methods can be divided into these categories (Knowledge Center):

- **Supervised**
- **Association**
- **Segmentation**

Supervised approach:

In this method the objective is predicting the outputs by using one or more input data. This technique includes wide range of approaches such as *decision trees* as a one of the important classification methodology (C&R Tree, QUEST, CHAID and C5.0 algorithms), *regression* approach which includes variety of methods algorithm (linear, logistic, generalized linear, and Cox regression algorithms), *Artificial Neural Networks* (ANN), *support vector machines (SVM, LSVM).*

Supervised models is useful method to predict a target which is clear and the objective is prediction of that data. It this project, the main goal is predicting the air pollutants such as PM25 by using two different types of the data.

Among of the supervised method, the ***classification and Regression*** (C&R method in PYTHON) Tree node generates a decision tree that allows to predict or classify target. The method uses recursive partitioning to divide the training dataset into some similar segments.

A generalized linear mixed model (GLMM) is another supervised method which is used a lot for finding the logical relationship between input data and output. Generalized linear mixed models in PYTHON include different of regression models such as simple linear regression to complex multiple models for non-normal longitudinal data.

Neural networks work similar of the way of human`s nervous system works. The basic units are neurons, which are typically organized into input, middle and output layers. It operates by modelling of the huge amount of interconnected processing units that resemble abstract versions of neurons. The ANN workflow are arranged in various layers; **input layer** includes the input data, **hidden layers** which cover different number of neurons, and **output layer** contain the target or outcome neuron. ANN approach has ability to predict more than one targets., with a unit or units representing the target field(s). The units are connected with varying connection strengths (or weights). Input data are presented to the first layer, and values are propagated from each neuron to every neuron in the next layer. Eventually, a result is delivered from the output layer.

Association approach

Association models tries to explore patterns in input dataset which are associated with one or more other entities. This method builds rule sets that explain these relationships. Apriori and Carma can be the algorithm of this model. One other type of association model is a sequence detection model, which finds sequential patterns in time-structured data.

Association method are applied mostly for predicting more than one targets—this method is very useful to predict more than one air pollutants such as CO2, SO2, Co and other air pollution elements. Association models associate a particular conclusion (such as the decision to buy something) with a set of conditions.

Segmentation approach

Segmentation models classifies the data into different clusters, of records that have similar patterns of input fields. This method could be one of the unsupervised method which focus only on input data similarities. Kohonen networks, K-Means clustering, two-step clustering and anomaly detection are some well-known example of this methodology.

Segmentation or clustering approach are useful when the target or outcome is not clear. Clustering models concentrate on recognizing segments of similar input data and labeling them based on the segments to which they belong. Accuracy of the input data is very important because the result of this method is not right or wrong outcome.

5.2 SELECT THE APPROPRIATE DATA-MINING METHOD(S) BASED ON DISCUSSION

In this study, based on input nature different Data mining / Machine Learning Techniques such as Multiple Regression, Decision Tree, and Support Vector Machine can be used to find the best method to generalize the target. However, based on good relationship between PM25 and some input we can use some kind of approaches. At this stage supervised method can help to predict the PM25 by using input data. One set of input data include both weather data (Temperature, pressure, win speed…) and air pollutants elements and the other input set is only climate data. So it seems for prediction of PM25 by using second input data we need to test different algorithms in supervised method. But based on previous section description the Regression and Artificial Neural Network could be the one of the optimized approach.

6 DATA-MINING ALGORITHM(S) SELECTION

6.1 CONDUCT EXPLORATORY ANALYSIS AND DISCUSS

With a huge number of data in all stations (420,000 rows data), we are now interested in finding out the trends and predicting them. The data extraction techniques help in converting the raw data into useful knowledge. An algorithm in machine learning for massive dataset is a kind of calculations that makes a model from row data. For making a proper model, one algorithm first analyzes the data which was provided, looking for specific types of patterns or trends. One appropriate algorithm utilizes the consequences of this analysis over many iterations to find the best coefficient for creating the mining model. The coefficients are then applied to all data set to extract actionable model and detailed statistics.

However, selecting the optimized algorithm to utilize for some analytical task can be a critical. by using various algorithms to perform the same business task, each algorithm generates a diverse result, and some algorithms can generate more than one type of result. For example, Decision Trees algorithm is not only for prediction, but also as a way to reduce the number of columns in a dataset, because the decision tree can identify columns that do not influence the final model.

Linear Regression algorithm as a variation of Decision Trees algorithm that can estimate a linear relationship between an input and target variable, and then use that relationship for prediction. The relationship is a linear equation which represents a series of data. For example, the line in the following diagram is the best possible linear representation of the data.

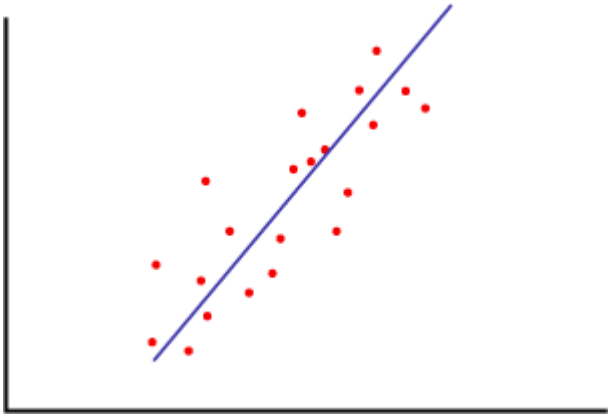


Fig. 51 linear representation of the data

Each data point in the diagram has an error associated with its distance from the regression line. The coefficients a and b in the regression equation adjust the angle and location of the regression line. the regression equation would be adjusting a and b until the sum of the errors that are associated with all the points reaches its minimum.

multiple and Polynomial regression are another regression method can be used for predicting the parameters. However, linear regression is a useful and famous method for modeling a response to a change in some underlying factor.

Decision tree as a one of the supervised method (classification technique) is one of the more famous method among other Regression methodology. The algorithm is a method that is called as recursive binary splitting. Beginning from the top, all observations are in the same region. Then a

set of questions are asked at each node for dividing of different observations into sub regions based on the answer of those questions. The algorithm stops when the threshold stopping criterion is met.

Support vector classifier divides a test observation depending on which of a hyperplane it lies. It let some observations to be on the wrong side of the margin or hyperplane. It is the solution to the following optimization problem. An example of Support Vector Classifier is as a below figure.

When **Support Vector Machine** emerged to ML methodology, SVM was known as support vector networks as a one of the supervised learning models that accompanies with associated learning algorithms which then investigates data that are utilized for the analysis of regression and classification. This algorithm is a representation of the examples as points in space, that are further mapped so that the examples of the separate categories are then classified by a obvious blank data that is should to be as wide range as possible.

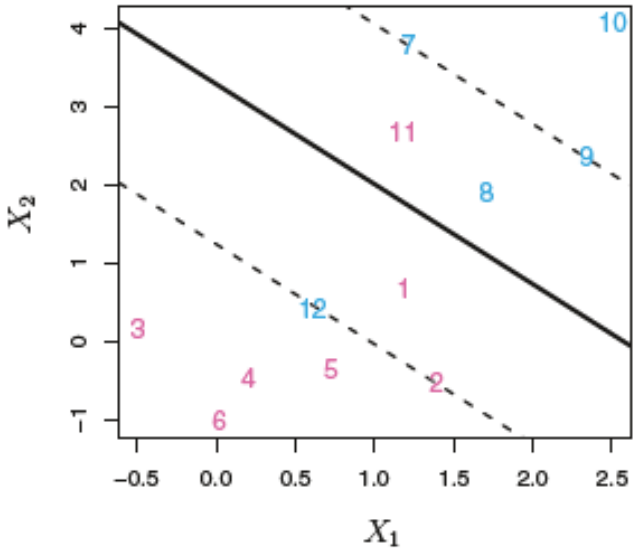


Fig. 52 SVM approaches (James, G. , 2015)

Neural Network algorithm is one of the well-known and flexible neural network architecture for ML. The algorithm operates by testing each input data against each possible state of the predictable attribute, and estimating probabilities for each combination based on the training data. All data after data cleaning and dealing with missing data must be divided to 2 or 3 categories such as Training set, Testing set and validation set data. One of the strangest point of this algorithm is

application in both classification or regression tasks and to predict a target based on some input. A neural network can also be used for association analysis. Even also there is possibility to predict more than one target by using ANN (Microsoft Neural Network Algorithm, 2018).

6.2 SELECT DATA-MINING ALGORITHMS BASED ON DISCUSSION

Based on the objective of this project PM25 as a dangerous air pollutant must be predicted by using combination of climate and pollutants data. Based on Fig. 53 and

Fig. 54, there is high and low correlation between PM25 and PM10 and O3. On the other side correlation between PM25 and climate parameters are complicated and it needs to use algorithm to find this relationship between target and input effectively (Fig. 55 and Fig. 56). Different regression algorithms are effective approach for reduce the uncertainty and predict the PM25 as a target. However, by using the Evaluation methodology, the best algorithm can be recognized well. Also, PM25 as continuous data which change from 0 to 1000 so using some algorithm such as logistic regression is not recommended for this kind of data.

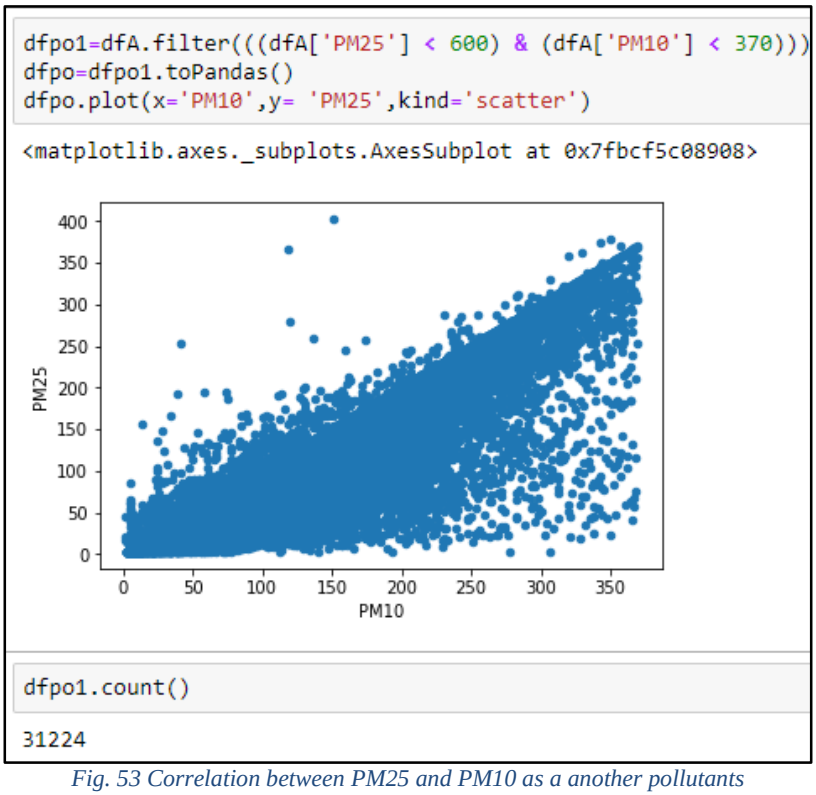


Fig. 53 Correlation between PM25 and PM10 as a another pollutants

```
dfpo.plot(x='NO2',y= 'PM25',kind='scatter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbcf5cad198>
```

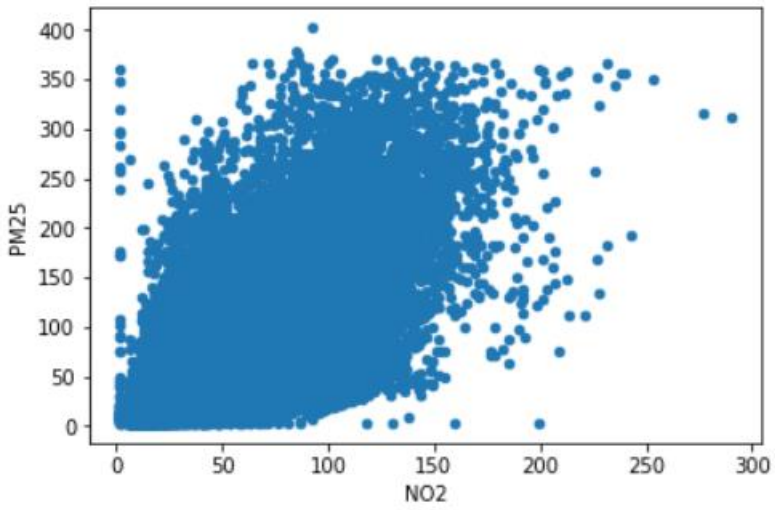


Fig. 54 Correlation between PM25 and NO2 as a another pollutants

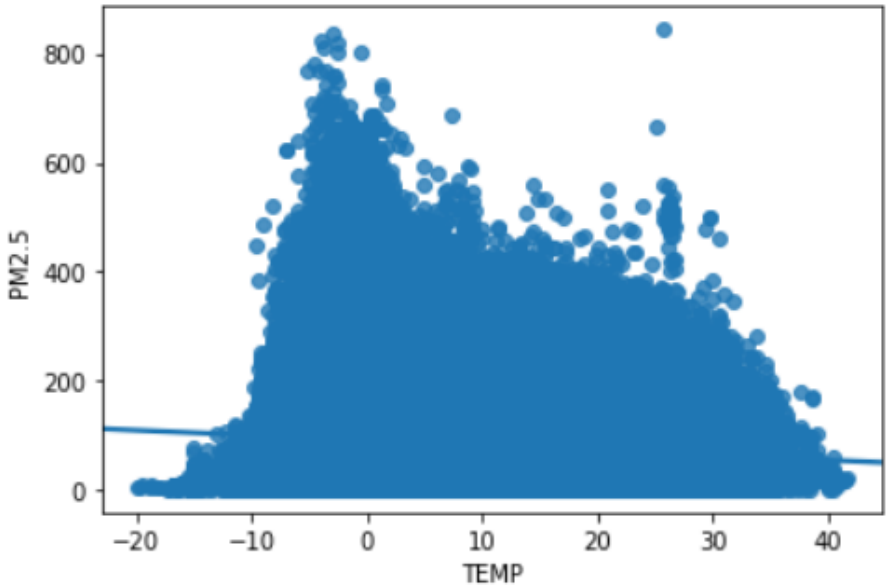


Fig. 55 Correlation between PM25 and TEMP as a climate data

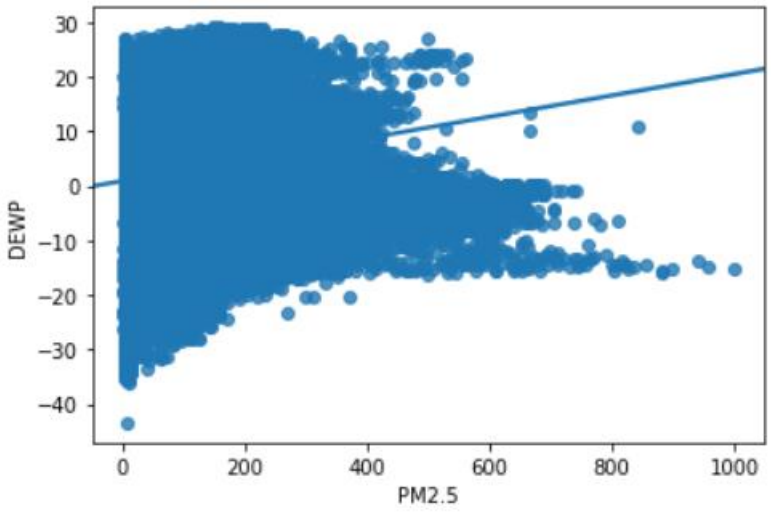


Fig. 56 Correlation between PM25 and DEWP as climate data

6.3 BUILD/SELECT APPROPRIATE MODEL(S) AND CHOOSE RELEVANT PARAMETER(S)

- **Linear Regression mechanism:**

This section, we use PySpark to implement simple linear regression. Then, we split our data into training and test sets, create a model using training set, evaluate your model using test set, and finally use model to predict unknown value. Simple Linear Regression is a method to help us understand the relationship between two variables:

- The independent variable (X)
- The dependent variable (that we want to predict) (Y)

The result of Linear Regression is a linear function that predicts the target variable as a function of the input variable. Based on very good correlation between PM25 and PM10 we are able to use this method to predict PM25 effectively. Fig. 57 shows good correlation between two variable and we can use simple linear model based of this data.

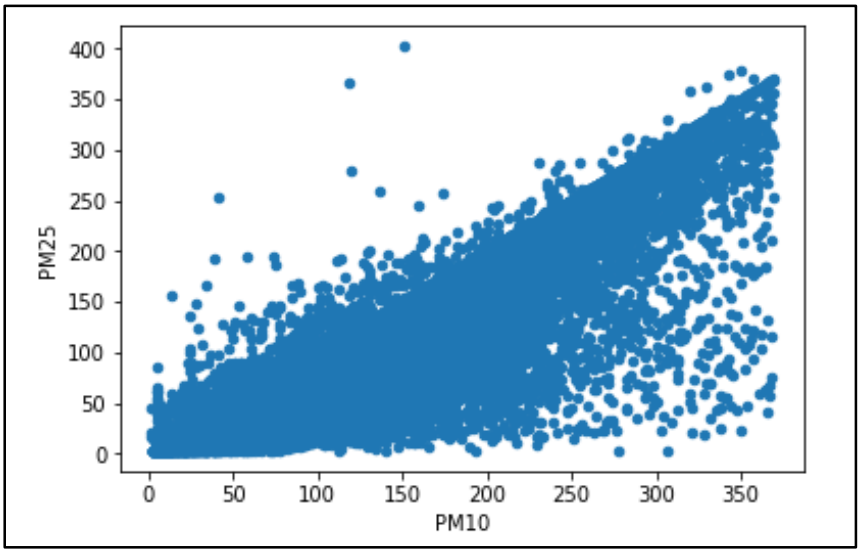


Fig. 57Residual plot between PM25 and PM10

- **Multiple Linear Regression**

If we want to use more variables in our model (such as NO2, CO, 03, TEMP, PRESS…) to predict PM25, we can use Multiple Linear Regression. Multiple Linear Regression is very similar to Simple Linear Regression, but this method is used to explain the relationship between one continuous response (dependent) variable and two or more predictor (independent) variables. We will explain the structure by using several predictor variables, but these results can generalize to any integer. Based on the Fig. 58 and Fig. 59 there is a suitable correlation between PM25, CO AND NO2. So we can run new model as a multi linear regression approach to obtain the result

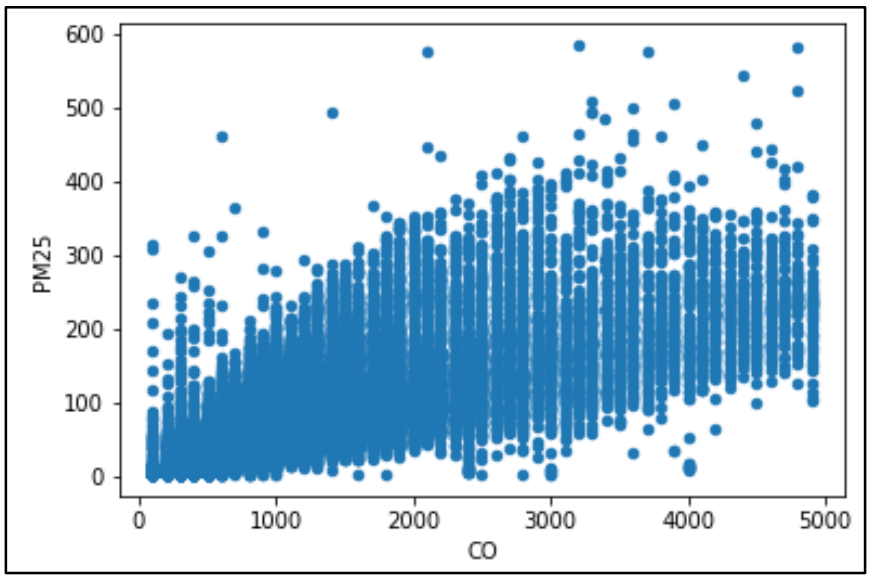


Fig. 58:correlation between CO and PM25

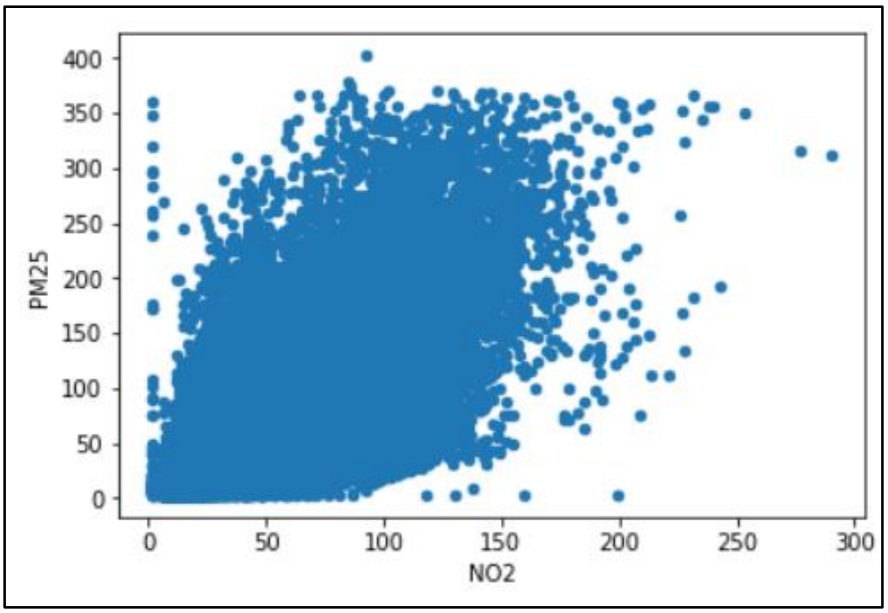


Fig. 59: correlation between NO2 and PM25

- **Generalized linear regression**

Spark's GeneralizedLinearRegression interface let for adjustable characterization of GLMs which able to apply for different prediction issues contain linear regression, Poisson regression, logistic regression, and others. Currently in spark.ml, only a subset of the exponential family distributions are supported and they are listed below.

GLMs require exponential family distributions that can be written in their "canonical" or "natural" form, aka natural exponential family distributions. Spark's generalized linear regression interface provides summary statistics for diagnosing the fit of GLM models, including residuals, p-values. However different parameters combination of PM10, NO2, SO2 and CO were used in different model of GLM in this study.

- **Multilayer perceptron classifier**

Multilayer perceptron classifier (MLPC) is a classifier based on the feedforward artificial neural network. MLPC include of multiple layers of nodes. Each layer is fully connected to the next layer

in the network. Nodes in the input layer represent the input data. All other nodes map inputs to outputs by a linear combination of the inputs with the node's weights and bias b and applying an activation function. Below code program was used in this study. All air pollutant material were used as an first layer or input layer.

```
from pyspark.ml.classification import MultilayerPerceptronClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

```
# specify layers for the neural network:
# input layer of size 4 (features), two intermediate of size 5 and 4
# and output of size 3 (classes)
layers = [3,2]

# create the trainer and set its parameters
trainer = MultilayerPerceptronClassifier(maxIter=10, layers=layers,featuresCol='WP_param',labelCol='PM25')
```

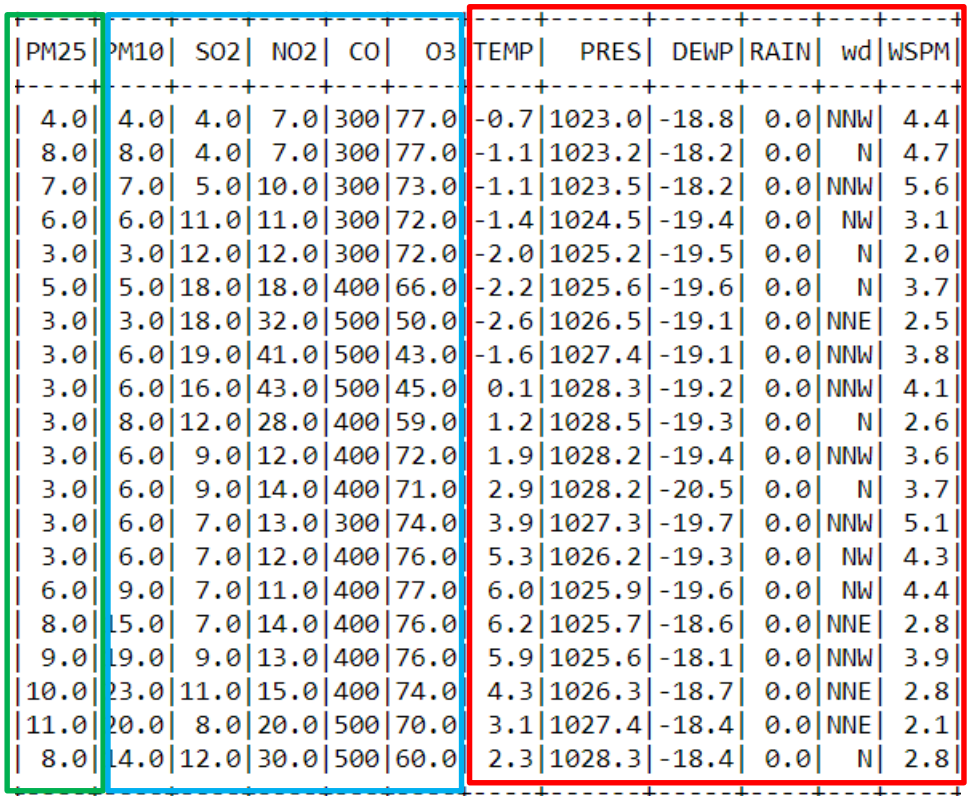
```
# train the model
model = trainer.fit(train_data1)
```

7 DATA MINING

7.1 CREATE AND JUSTIFY TEST DESIGNS

Dataset includes different row data which is classified into one target, the all climate and air pollution data was imported to the first model (**Error! Reference source not found.**) and also the climate data was imported to second model . After running the model probably some irrelevant input data will be skipped to final model. However, in both models, all data was divided to Test, Train data set. Based on them 383590 rows, 77000 rows and 306600 rows are dedicated to test, and train approaches respectively. There is no missing data in all of them and Extreme data mostly was dropped. Meanwhile most of outliers from air pollution data was removed. In addition, Fig. 61 and Fig. 62 indicate the train and test set spec in model 1 and 2 respectively. Model-1 includes all available data but model 2 includes only climate data. Also data sharing between test and train set data must be logical. Data share in training set must be more than test set data. Because in this step all relationship must be built. But based on more investigation, divided all data into 70, 30 or 80,20 could be recommended.

```
# Let's do a randomised 70/30 split.
# Remember, you can use other splits depending on how easy/difficult it is to train your model.
train_data1,test_data1 = final_data1.randomSplit([0.7,0.3])
```



PM25	PM10	SO2	NO2	CO	O3	TEMP	PRES	DEWP	RAIN	wd	WSPM
4.0	4.0	4.0	7.0	300	77.0	-0.7	1023.0	-18.8	0.0	NNW	4.4
8.0	8.0	4.0	7.0	300	77.0	-1.1	1023.2	-18.2	0.0	N	4.7
7.0	7.0	5.0	10.0	300	73.0	-1.1	1023.5	-18.2	0.0	NNW	5.6
6.0	6.0	11.0	11.0	300	72.0	-1.4	1024.5	-19.4	0.0	NW	3.1
3.0	3.0	12.0	12.0	300	72.0	-2.0	1025.2	-19.5	0.0	N	2.0
5.0	5.0	18.0	18.0	400	66.0	-2.2	1025.6	-19.6	0.0	N	3.7
3.0	3.0	18.0	32.0	500	50.0	-2.6	1026.5	-19.1	0.0	NNE	2.5
3.0	6.0	19.0	41.0	500	43.0	-1.6	1027.4	-19.1	0.0	NNW	3.8
3.0	6.0	16.0	43.0	500	45.0	0.1	1028.3	-19.2	0.0	NNW	4.1
3.0	8.0	12.0	28.0	400	59.0	1.2	1028.5	-19.3	0.0	N	2.6
3.0	6.0	9.0	12.0	400	72.0	1.9	1028.2	-19.4	0.0	NNW	3.6
3.0	6.0	9.0	14.0	400	71.0	2.9	1028.2	-20.5	0.0	N	3.7
3.0	6.0	7.0	13.0	300	74.0	3.9	1027.3	-19.7	0.0	NNW	5.1
3.0	6.0	7.0	12.0	400	76.0	5.3	1026.2	-19.3	0.0	NW	4.3
6.0	9.0	7.0	11.0	400	77.0	6.0	1025.9	-19.6	0.0	NW	4.4
8.0	15.0	7.0	14.0	400	76.0	6.2	1025.7	-18.6	0.0	NNE	2.8
9.0	19.0	9.0	13.0	400	76.0	5.9	1025.6	-18.1	0.0	NNW	3.9
10.0	23.0	11.0	15.0	400	74.0	4.3	1026.3	-18.7	0.0	NNE	2.8
11.0	20.0	8.0	20.0	500	70.0	3.1	1027.4	-18.4	0.0	NNE	2.1
8.0	14.0	12.0	30.0	500	60.0	2.3	1028.3	-18.4	0.0	N	2.8

Fig. 60: Train and test data; blue one is model-1 as input data, red one is model-2 and green one is target

```
train_data1.describe().show()
```

summary	PM25
count	192604
mean	75.82448235758345
stddev	68.02640183742136
min	2.0
max	533.0

```
test_data1.describe().show()
```

summary	PM25
count	82692
mean	75.633813428143
stddev	68.44003606995318
min	3.0
max	500.0

Fig. 61 : Description of train and test set in first model

```
train_data2.describe().show()
+-------+-----------------+
|summary|             PM25|
+-------+-----------------+
|  count|           193017|
|   mean|75.62382795297825|
| stddev|68.07789301756269|
|    min|              2.0|
|    max|            500.0|
+-------+-----------------+


test_data2.describe().show()
+-------+-----------------+
|summary|             PM25|
+-------+-----------------+
|  count|            82279|
|   mean|76.10356834672275|
| stddev|68.32090165256585|
|    min|              2.0|
|    max|            533.0|
+-------+-----------------+
```

Fig. 62 : Description of train and test set in first model

7.2 CONDUCT DATA MINING

- **Multi Linear Regression**

Based on data mining methods and algorithms which were described in previous chapter, Supervised method was selected. Multi-regression approach was chosen for PM25 predictions. In the first model by using combination of all input, regression algorithms have a very good correlation and it is about 0.82 (**Error! Reference source not found.**). In the second model that I used only climate data, the correlation shows 0.2. in PySpark the data must be assembled and transformed as a unique data we call it features. But here based on below coding 2 kind of data was used as an input the first category include PM10, CO and NO2 which is called WP_param and second model contain only weather parameters which is called W_Param. Based on the result only the first model had a suitable result. The transformed data in both models are shown in Fig. 63 and Fig. 64. Also the description of test and train set data in both models are available in Fig. 65 and Fig. 66.

First model

```
# The input columns are the combination of weather and airpollution
assembler_1 = VectorAssembler(
    inputCols=["PM10",'NO2','CO'],outputCol="WP_param")
# Now that we've created the assembler variable, let's actually tra
output_1 = assembler_1.transform(dfx)
# Let's select two columns (the feature and predictor).
# This is now in the appropriate format to be processed by Spark.
final_data1 = output_1.select("WP_param",'PM25')
final_data1.show()
# Let's do a randomised 70/30 split.
# Remember, you can use other splits depending on how easy/difficul
train_data1,test_data1 = final_data1.randomSplit([0.7,0.3])
```

SECOND MODEL

```
# The input columns are the combination of weather and airpollution parame
assembler_2 = VectorAssembler(
    inputCols=["TEMP",'DEWP','WSPM'],outputCol="WP_param")
# Now that we've created the assembler variable, let's actually transform
output_2 = assembler_2.transform(dfx)
# Let's select two columns (the feature and predictor).
# This is now in the appropriate format to be processed by Spark.
final_data2 = output_2.select("W_param",'PM25')
final_data2.show()
# Let's do a randomised 70/30 split.
# Remember, you can use other splits depending on how easy/difficult it is
train_data2,test_data2 = final_data1.randomSplit([0.7,0.3])
```

```
+-----------------+----+
|         WP_param|PM25|
+-----------------+----+
|  [4.0,7.0,300.0]| 4.0|
|  [8.0,7.0,300.0]| 8.0|
| [7.0,10.0,300.0]| 7.0|
| [6.0,11.0,300.0]| 6.0|
| [3.0,12.0,300.0]| 3.0|
| [5.0,18.0,400.0]| 5.0|
| [3.0,32.0,500.0]| 3.0|
| [6.0,41.0,500.0]| 3.0|
| [6.0,43.0,500.0]| 3.0|
| [8.0,28.0,400.0]| 3.0|
| [6.0,12.0,400.0]| 3.0|
| [6.0,14.0,400.0]| 3.0|
| [6.0,13.0,300.0]| 3.0|
| [6.0,12.0,400.0]| 3.0|
| [9.0,11.0,400.0]| 6.0|
|[15.0,14.0,400.0]| 8.0|
|[19.0,13.0,400.0]| 9.0|
|[23.0,15.0,400.0]|10.0|
|[20.0,20.0,500.0]|11.0|
|[14.0,30.0,500.0]| 8.0|
+-----------------+----+
only showing top 20 rows
```

Fig. 63 : Transformed data in first model

```
+----------------+----+
|         W_param|PM25|
+----------------+----+
|[-0.7,-18.8,4.4]| 4.0|
|[-1.1,-18.2,4.7]| 8.0|
|[-1.1,-18.2,5.6]| 7.0|
|[-1.4,-19.4,3.1]| 6.0|
|[-2.0,-19.5,2.0]| 3.0|
|[-2.2,-19.6,3.7]| 5.0|
|[-2.6,-19.1,2.5]| 3.0|
|[-1.6,-19.1,3.8]| 3.0|
| [0.1,-19.2,4.1]| 3.0|
| [1.2,-19.3,2.6]| 3.0|
| [1.9,-19.4,3.6]| 3.0|
| [2.9,-20.5,3.7]| 3.0|
| [3.9,-19.7,5.1]| 3.0|
| [5.3,-19.3,4.3]| 3.0|
| [6.0,-19.6,4.4]| 6.0|
| [6.2,-18.6,2.8]| 8.0|
| [5.9,-18.1,3.9]| 9.0|
| [4.3,-18.7,2.8]|10.0|
| [3.1,-18.4,2.1]|11.0|
| [2.3,-18.4,2.8]| 8.0|
+----------------+----+
only showing top 20 rows
```

Fig. 64 : Transformed data in second model

```
train_data1.describe().show()
```

```
+-------+-----------------+
|summary|             PM25|
+-------+-----------------+
|  count|           192604|
|   mean|75.82448235758345|
| stddev|68.02640183742136|
|    min|              2.0|
|    max|            533.0|
+-------+-----------------+
```

```
test_data1.describe().show()
```

```
+-------+-----------------+
|summary|             PM25|
+-------+-----------------+
|  count|            82692|
|   mean| 75.633813428143|
| stddev|68.44003606995318|
|    min|              3.0|
|    max|            500.0|
+-------+-----------------+
```

Fig. 65 : *Description of train and test set in first model*

```
train_data2.describe().show()
```

```
+-------+-----------------+
|summary|             PM25|
+-------+-----------------+
|  count|           193017|
|   mean|75.62382795297825|
| stddev|68.07789301756269|
|    min|              2.0|
|    max|            500.0|
+-------+-----------------+
```

```
test_data2.describe().show()
```

```
+-------+-----------------+
|summary|             PM25|
+-------+-----------------+
|  count|            82279|
|   mean|76.10356834672275|
| stddev|68.32090165256585|
|    min|              2.0|
|    max|            533.0|
+-------+-----------------+
```

Fig. 66 : *Description of train and test set in first model*

- **Generalized linear regression**

In this approach both data models were tested and the code program was used as below.

```
from pyspark.ml.regression import GeneralizedLinearRegression
```

```
glr = GeneralizedLinearRegression(family="gaussian", link="identity",
                                  featuresCol='WP_param',labelCol='PM25', maxIter=35, regParam=0.1)
```

7.3 SEARCH FOR PATTERNS

Pattern_1: The first pattern contain all data; PM10, SO2, NO2, CO, O3, TEMP, PRES and DEWP as a pollutant and climate data Also Fig. 67 as a table explains relationship between target and input clearly. Multi Linear regression model works well. There is high correlation between depended and variable data. Green dolor show target, blue and red one indicate pollution and climate input data respectively.

Pattern_1

```
assembler_p1 = VectorAssembler(inputCols=['PM10','CO','SO2','NO2','O3','TEMP','PRES','DEWP','WSPM'],outputCol="p1")
output_p1 = assembler_p1.transform(dfx)
final_datap1 = output_p1.select("p1",'PM25')
train_datap1,test_datap1 = final_datap1.randomSplit([0.7,0.3])
train_datap1.describe().show()
lrp1 = LinearRegression(featuresCol='p1',labelCol='PM25', predictionCol='prediction')
lrmp1 = lrp1.fit(train_datap1)
print("Coefficients: {} Intercept: {}".format(lrmp1.coefficients,lrmp1.intercept))
test_resultsp1 = lrmp1.evaluate(test_datap1)
print("R2: {}".format(test_resultsp1.r2))
```

station	avg(PM25)	avg(PM10)	avg(SO2)	avg(NO2)	avg(CO)
Changping	70.312328264129	94.08640188488725	15.061356751629388	44.31903533857593	1151.7164407453872
Aotizhongxin	81.86363036303632	109.12068521137829	17.16720528052805	58.97633191890618	1256.5747289014616
Wanshouxigong	84.23851013183594	111.91869201660157	17.152568701171877	55.30718125915527	1362.5316467285156
Wanliu	81.98145851015212	108.76586146112164	17.700088215708004	64.42928958020501	1293.4501534242997
Dongsi	84.93315643747118	110.99191443074693	16.994465686597668	53.717313985760434	1316.524391851803
Shunyi	79.40072530966417	99.3805458038021	14.14410478903093	44.83848316221766	1201.778267205405
Nongzhanguan	84.72078275049827	109.17592861025548	18.560198103521167	58.2134567826297	1327.7843812284834
Tiantan	81.74984014858569	105.62786590749931	14.140742319520141	53.21308315318332	1297.4341868891393
Dingling	66.51251836708619	84.52780297706508	11.944167252283906	27.267578100044716	916.5472113971763
Huairou	70.28566923173962	92.5257915983348	12.27582313611707	32.633827425255454	1032.3982906522012
Gucheng	83.86565345803594	119.08500184592667	15.095190007383708	55.583878547255715	1325.5096911149396
Guanyuan	83.1010507392369	108.97455909245883	17.263085069584356	57.69859070762173	1265.7097604066578

```
#average of CLIMATE properties in each station
dfg2=df.groupBy('station').mean('TEMP','PRES','DEWP','WSPM','pm25')
dfg2.show()
```

station	avg(TEMP)	avg(PRES)	avg(DEWP)	avg(WSPM)	avg(pm25)
Changping	13.401676579712658	1007.9940087512003	1.135298797466421	1.8657568617851341	70.312328264129
Aotizhongxin	13.775610750644972	1011.8003847766261	3.2411063963539273	1.7204714757189838	81.86363036303632
Wanshouxigong	13.773187219529774	1011.5600415548229	2.607440185546893	1.7559844970702887	84.23851013183594
Wanliu	14.020991264246053	1010.7487723336236	3.831086374616461	1.5029705555918151	81.98145851015212
Dongsi	13.623337899612281	1012.8790166898078	2.2777935262706968	1.8725097237787323	84.93315643747118
Shunyi	12.787636529806317	1013.6480448152555	1.5428727561767166	1.8423064184937326	79.40072530966417
Nongzhanguan	13.698473410997504	1012.495573121202	2.5219876789273363	1.858044935676724	84.72078275049827
Tiantan	13.63201256072133	1012.6077624354809	2.4954815333556724	1.8534147306883992	81.74984014858569
Dingling	13.556273950663535	1007.7080069847617	1.4034114866159761	1.850939117102149	66.51251836708619
Huairou	12.276487150356086	1007.805222919518	2.064693452756403	1.6541062192506786	70.28566923173962
Gucheng	13.926933343051434	1008.79284781784	2.632562146197413	1.360549470834365	83.86565345803594
Guanyuan	13.707450852837765	1011.856881205573	3.242757958032418	1.719461922325867	83.1010507392369

Fig. 67: data pattern 1

Pattern_2: this pattern includes only 4 climate elements; TEMP,DEWP,WSPM and PRESS (Fig. 68). Multi linear regression algorithms which were used. In this pattern only 4 kind of data was used and its relationship with target is clear.

Pattern_2

```
assembler_p2 = VectorAssembler(inputCols=['TEMP','PRES','DEWP','WSPM'],outputCol="p2")
output_p2 = assembler_p2.transform(dfx)
final_datap2 = output_p2.select("p2",'PM25')
train_datap2,test_datap2 = final_datap2.randomSplit([0.7,0.3])
train_datap2.describe().show()
lrp2 = LinearRegression(featuresCol='p2',labelCol='PM25', predictionCol='prediction')
lrmp2 = lrp2.fit(train_datap2)
print("Coefficients: {} Intercept: {}".format(lrmp2.coefficients,lrmp2.intercept))
test_resultsp2 = lrmp2.evaluate(test_datap2)
print("R2: {}".format(test_resultsp2.r2))
```

station	avg(PM25)	avg(PM10)	avg(SO2)	avg(NO2)	avg(CO)
Changping	70.312328264129	94.08640188488725	15.061356751629388	44.31903533857593	1151.7164407453872
Aotizhongxin	81.86363036303632	109.12068521137829	17.16720528052805	58.97633191890618	1256.5747289014616
Wanshouxigong	84.23851013183594	111.91869201660157	17.152568701171877	55.30718125915527	1362.5316467285156
Wanliu	81.98145851015212	108.76586146112164	17.700088215708004	64.42928958020501	1293.4501534242997
Dongsi	84.93315643747118	110.99191443074693	16.994465686597668	53.717313985760434	1316.524391851803
Shunyi	79.40072530966417	99.3805458038021	14.14410478903093	44.83848316221766	1201.778267205405
Nongzhanguan	84.72078275049827	109.17592861025548	18.560198103521167	58.2134567826297	1327.7843812284834
Tiantan	81.74984014858569	105.62786590749931	14.140742319520141	53.21308315318332	1297.4341868891393
Dingling	66.51251836708619	84.52780297706508	11.944167252283906	27.267578100044716	916.5472113971763
Huairou	70.28566923173962	92.5257915983348	12.27582313611707	32.633827425255454	1032.3982906522012
Gucheng	83.86565345803594	119.08500184592667	15.095190007383708	55.583878547255715	1325.5096911149396
Guanyuan	83.1010507392369	108.97455909245883	17.263085069584356	57.69859070762173	1265.7097604066578

```
#average of CLIMATE properties in each station
dfg2=df.groupBy('station').mean('TEMP','PRES','DEWP','WSPM','pm25')
dfg2.show()
```

station	avg(TEMP)	avg(PRES)	avg(DEWP)	avg(WSPM)	avg(pm25)
Changping	13.401676579712658	1007.9940087512003	1.135298797466421	1.8657568617851341	70.312328264129
Aotizhongxin	13.775610750644972	1011.8003847766261	3.2411063963539273	1.7204714757189838	81.86363036303632
Wanshouxigong	13.773187219529774	1011.5600415548229	2.607440185546893	1.7559844970702887	84.23851013183594
Wanliu	14.020991264246053	1010.7487723336236	3.831086374616461	1.5029705555918151	81.98145851015212
Dongsi	13.623337899612281	1012.8790166898078	2.2777935262706968	1.8725097237787323	84.93315643747118
Shunyi	12.787636529806317	1013.6480448152555	1.5428727561767166	1.8423064184937326	79.40072530966417
Nongzhanguan	13.698473410997504	1012.495573121202	2.5219876789273363	1.858044935676724	84.72078275049827
Tiantan	13.63201256072133	1012.6077624354809	2.4954815333556724	1.8534147306883992	81.74984014858569
Dingling	13.556273950663535	1007.7080069847617	1.4034114866159761	1.850939117102149	66.51251836708619
Huairou	12.276487150356086	1007.805222919518	2.064693452756403	1.6541062192506786	70.28566923173962
Gucheng	13.926933343051434	1008.79284781784	2.632562146197413	1.360549470834365	83.86565345803594
Guanyuan	13.707450852837765	1011.856881205573	3.242757958032418	1.719461922325867	83.1010507392369

Fig. 68: Data pattern 2

Pattern_3: this pattern includes PM10, CO, and NO2 as in input (Fig. 69). simple regression regression gives us proper correlation among the algorithms which were used in this pattern. In this pattern the relationship with target is clear.

station	avg(PM25)	avg(PM10)	avg(SO2)	avg(NO2)	avg(CO)
Changping	70.312328264129	94.08640188488725	15.061356751629388	44.31903533857593	1151.7164407453872
Aotizhongxin	81.86363036303632	109.12068521137829	17.16720528052805	58.97633191890618	1256.5747289014616
Wanshouxigong	84.23851013183594	111.91869201660157	17.152568701171877	55.30718125915527	1362.5316467285156
Wanliu	81.98145851015212	108.76586146112164	17.700088215708004	64.42928958020501	1293.4501534242997
Dongsi	84.93315643747118	110.99191443074693	16.994465686597668	53.717313985760434	1316.524391851803
Shunyi	79.40072530966417	99.3805458038021	14.14410478903093	44.83848316221766	1201.778267205405
Nongzhanguan	84.72078275049827	109.17592861025548	18.560198103521167	58.2134567826297	1327.7843812284834
Tiantan	81.74984014858569	105.62786590749931	14.140742319520141	53.21308315318332	1297.4341868891393
Dingling	66.51251836708619	84.52780297706508	11.944167252283906	27.267578100044716	916.5472113971763
Huairou	70.28566923173962	92.5257915983348	12.27582313611707	32.633827425255454	1032.3982906522012
Gucheng	83.86565345803594	119.08500184592667	15.095190007383708	55.583878547255715	1325.5096911149396
Guanyuan	83.1010507392369	108.97455909245883	17.263085069584356	57.69859070762173	1265.7097604066578

Fig. 69: data pattern 3

Pattern_3

```
assembler_p3 = VectorAssembler(inputCols=['PM10','CO','NO2'],outputCol="p3")
output_p3 = assembler_p3.transform(dfx)
final_datap3 = output_p3.select("p3",'PM25')
train_datap3,test_datap3 = final_datap3.randomSplit([0.7,0.3])
train_datap3.describe().show()
lrp3 = LinearRegression(featuresCol='p3',labelCol='PM25', predictionCol='prediction')
lrmp3 = lrp3.fit(train_datap3)
print("Coefficients: {} Intercept: {}".format(lrmp3.coefficients,lrmp3.intercept))
test_resultsp3 = lrmp3.evaluate(test_datap3)
print("R2: {}".format(test_resultsp3.r2))
```

Pattern 4: this pattern used the data from pattern 1. It includes all air pollution and weather data. But Generalized Linear regression was used for prediction of PM25. different family were tested and Gaussian was selected. Also the relationship of the input data with target is clear.

Pattern_4

```
from pyspark.ml.regression import GeneralizedLinearRegression
glrp4 = GeneralizedLinearRegression(family="gaussian", link="identity",
                                    featuresCol='p1',labelCol='PM25', maxIter=35, regParam=0.1)
modelgp4 = glrp4.fit(train_datap1)
```

```
|      station|         avg(PM25)|         avg(PM10)|          avg(SO2)|          avg(NO2)|           avg(CO)|
+-------------+------------------+------------------+------------------+------------------+------------------+
|    Changping|  70.312328264129| 94.08640188488725|15.061356751629388| 44.31903533857593|1151.7164407453872|
| Aotizhongxin|81.86363036303632|109.12068521137829| 17.16720528052805| 58.97633191890618|1256.5747289014616|
|Wanshouxigong|84.23851013183594|111.91869201660157|17.152568701171877| 55.30718125915527|1362.5316467285156|
|       Wanliu|81.98145851015212|108.76586146112164|17.700088215708004| 64.42928958020501|1293.4501534242997|
|       Dongsi|84.93315643747118|110.99191443074693|16.994465686597668|53.717313985760434| 1316.524391851803|
|       Shunyi|79.40072530966417|  99.3805458038021| 14.14410478903093| 44.83848316221766| 1201.778267205405|
| Nongzhanguan|84.72078275049827|109.17592861025548|18.560198103521167|  58.2134567826297|1327.7843812284834|
|      Tiantan|81.74984014858569|105.62786590749931|14.140742319520141| 53.21308315318332|1297.4341868891393|
|     Dingling|66.51251836708619| 84.52780297706508|11.944167252283906|27.267578100044716| 916.5472113971763|
|      Huairou|70.28566923173962|  92.5257915983348| 12.27582313611707|32.633827425255454|1032.3982906522012|
|      Gucheng|83.86565345803594|119.08500184592667|15.095190007383708|55.583878547255715|1325.5096911149396|
|     Guanyuan| 83.1010507392369|108.97455909245883|17.263085069584356| 57.69859070762173|1265.7097604066578|
+-------------+------------------+------------------+------------------+------------------+------------------+
```

```
#average of CLIMATE properties in each station
dfg2=df.groupBy('station').mean('TEMP','PRES','DEWP','WSPM','pm25')
dfg2.show()
```

```
+-------------+------------------+------------------+------------------+------------------+------------------+
|      station|         avg(TEMP)|         avg(PRES)|         avg(DEWP)|         avg(WSPM)|          avg(pm25)|
+-------------+------------------+------------------+------------------+------------------+------------------+
|    Changping|13.401676579712658|1007.9940087512003| 1.135298797466421|1.8657568617851341|  70.312328264129|
| Aotizhongxin|13.775610750644972|1011.8003847766261|3.2411063963539273|1.7204714757189838|81.86363036303632|
|Wanshouxigong|13.773187219529774|1011.5600415548229| 2.607440185546893|1.7559844970702887|84.23851013183594|
|       Wanliu|14.020991264246053|1010.7487723336236| 3.831086374616461|1.5029705555918151|81.98145851015212|
|       Dongsi|13.623337899612281|1012.8790166898078|2.2777935262706968|1.8725097237787323|84.93315643747118|
|       Shunyi|12.787636529806317|1013.6480448152555|1.5428727561767166|1.8423064184937326|79.40072530966417|
| Nongzhanguan|13.698473410997504| 1012.495573121202|2.5219876789273363| 1.858044935676724|84.72078275049827|
|      Tiantan| 13.63201256072133|1012.6077624354809|2.4954815333556724|1.8534147306883992|81.74984014858569|
|     Dingling|13.556273950663535|1007.7080069847617|1.4034114866159761| 1.850939117102149|66.51251836708619|
|      Huairou|12.276487150356086| 1007.805222919518| 2.064693452756403|1.6541062192506786|70.28566923173962|
|      Gucheng|13.926933343051434|  1008.79284781784| 2.632562146197413| 1.360549470834365|83.86565345803594|
|     Guanyuan|13.707450852837765| 1011.856881205573| 3.242757958032418| 1.719461922325867| 83.1010507392369|
```

Fig. 70: data pattern 4

Pattern_5: this pattern used the data from pattern 3. It includes all air pollution data. But Generalized Linear regression was used for prediction of PM25. different family were tested and Gaussian was selected. Also the relationship of the input data with target is clear.

Pattern_5

```
from pyspark.ml.regression import GeneralizedLinearRegression
glrp5 = GeneralizedLinearRegression(family="gaussian", link="identity",
                                    featuresCol='p3',labelCol='PM25', maxIter=35, regParam=0.1)
modelgp5 = glrp5.fit(train_datap3)
```

station	avg(PM25)	avg(PM10)	avg(SO2)	avg(NO2)	avg(CO)
Changping	70.312328264129	94.08640188488725	15.061356751629388	44.31903533857593	1151.7164407453872
Aotizhongxin	81.86363036303632	109.12068521137829	17.16720528052805	58.97633191890618	1256.5747289014616
Wanshouxigong	84.23851013183594	111.91869201660157	17.152568701171877	55.30718125915527	1362.5316467285156
Wanliu	81.98145851015212	108.76586146112164	17.700088215708004	64.42928958020501	1293.4501534242997
Dongsi	84.93315643747118	110.99191443074693	16.994465686597668	53.717313985760434	1316.524391851803
Shunyi	79.40072530966417	99.3805458038021	14.14410478903093	44.83848316221766	1201.778267205405
Nongzhanguan	84.72078275049827	109.17592861025548	18.560198103521167	58.2134567826297	1327.7843812284834
Tiantan	81.74984014858569	105.62786590749931	14.140742319520141	53.21308315318332	1297.4341868891393
Dingling	66.51251836708619	84.52780297706508	11.944167252283906	27.267578100044716	916.5472113971763
Huairou	70.28566923173962	92.5257915983348	12.27582313611707	32.633827425255454	1032.3982906522012
Gucheng	83.86565345803594	119.08500184592667	15.095190007383708	55.583878547255715	1325.5096911149396
Guanyuan	83.1010507392369	108.97455909245883	17.263085069584356	57.69859070762173	1265.7097604066578

Fig. 71: data pattern 5

8 INTERPRETATION

8.1 STUDY AND DISCUSS THE MINED PATTERNS

In previous section, by using different input 5 pattern have been built. In all approach, Python packages was chosen and different regression algorithm was used.

Pattern-1: in this pattern as a complete pattern, all air pollution and climate parameters was used and correlation in all algorithm are very good. Multi Linear Regression method shows the correlation about 0.86 between original and predicted PM25. The error in (training, testing and validation) are very small and ignorable. This approach is the best among all patterns.

summary	PM25
count	193136
mean	75.88223894043576
stddev	68.20709324114831
min	2.0
max	533.0

```
Coefficients: [0.5930194067555453,0.020409177514042746,0.08846813728806564,-0.009997742390691908,0.07755930324190029,-0.8494709
426465086,0.5043712463841379,1.285755438405562,0.3494120592287339] Intercept: -515.7288581617388
R2: 0.8410966573857191
```

Fig. 72 Multi Linear regression algorithms which have the best correlation with PM25 in model 1 and pattern-1

Pattern-2: in this pattern, only climate data was imported to Multi linear regression model, however based on the evaluation correlation in all algorithm is not good. Multi Linear Regression method shows the correlation about 0.18 between original and predicted PM25 . The error in (training, testing and validation) is considerable. This approach is not recommended.

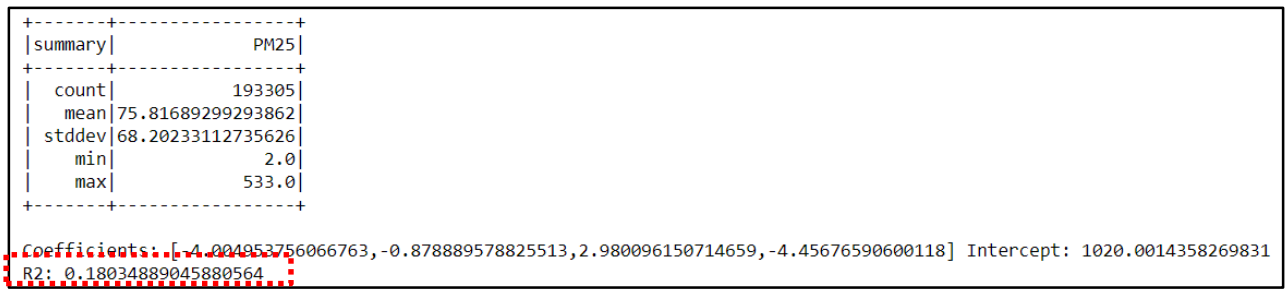


Fig. 73 Three best algorithms which have the best correlation with PM25 in model 1 and pattern 2

Pattern-3: in this pattern as a subsidiary of pattern-1, include only air pollution materials as an important input data and correlation in all algorithm are very good. Multi Linear Regression method shows the correlation about 0.82 between original and predicted PM25

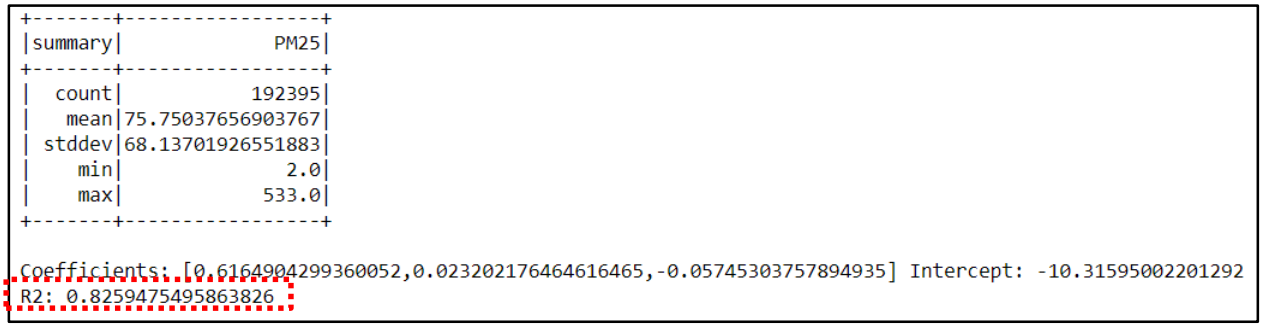


Fig. 74 Multi Linear regression algorithms which have the best correlation with PM25 in pattern-3

Pattern-4: in this pattern as a subsidiary of pattern-1, include air pollution and weather data and correlation in all algorithm are very good. GLR method shows the correlation about 0.6 between original and predicted PM25.

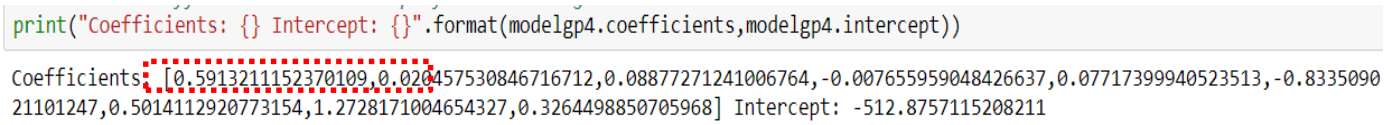


Fig. 75 GLR algorithms which have the best correlation with PM25 in pattern-4

Pattern-5: in this pattern includes only pollution data and correlation in all algorithm is suitable. GLR method shows good correlation about 0.62 between original and predicted PM25.

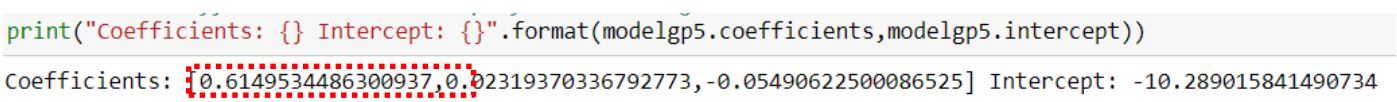


Fig. 76 GLR algorithms which have the best correlation with PM25 in pattern-5

8.2 VISUALIZE THE DATA, RESULTS, MODELS, AND PATTERNS

In previous chapter all relevant data for each pattern was introduced and based on different kind of data was used to increase the correlation between input and target. Pattern-1, 3 and 5 by using the various air pollution and data show very suitable correlation by the graph. Multi Linear Regression algorithm and GLR as a supervised method was the best approach to simulate PM25 (Fig. 77, Fig. 79 and Fig. 80). However pattern 4 and 5 by using only climate data show moderate to weak correlation between input data and target (Fig. 78and Fig. 81).

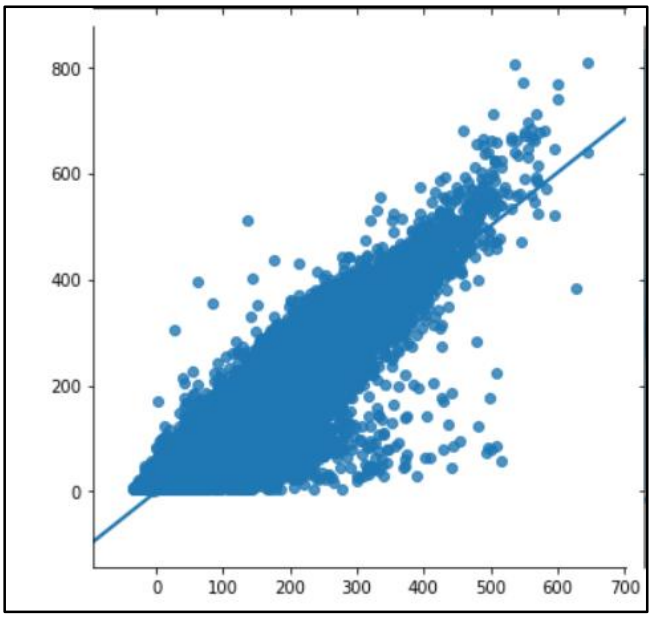


Fig. 77 Correlation between test data and prediction data by multi-regression model in pattern 1

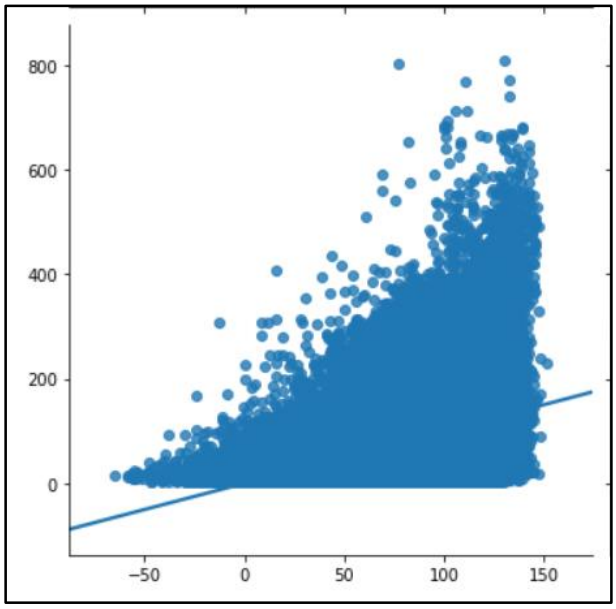


Fig. 78 *Correlation between test data and prediction data by multi-regression model in pattern 2*

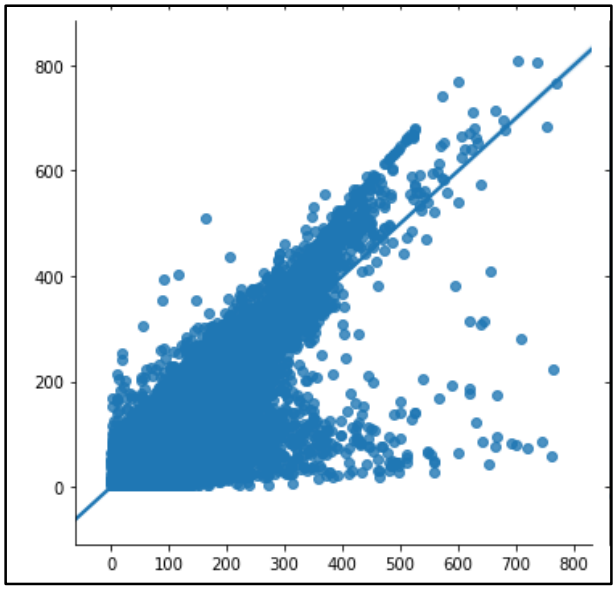


Fig. 79 *Correlation between test data and prediction data by simple linear regression model in pattern 3*

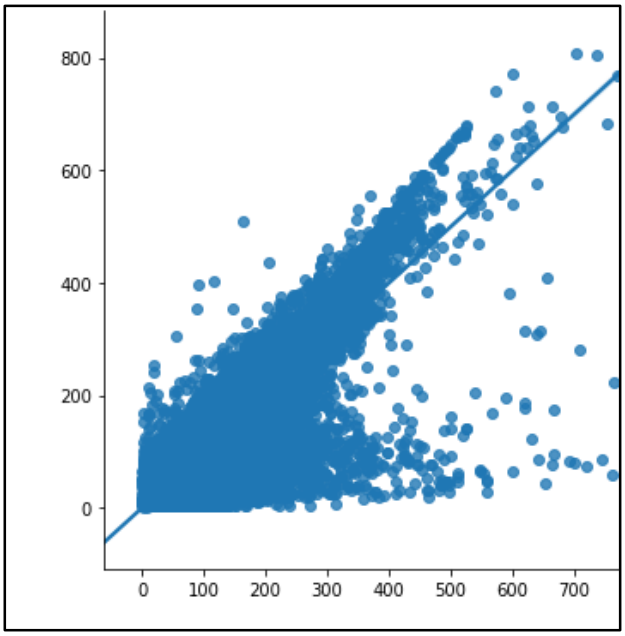


Fig. 80 *Correlation between test data and prediction data by GLR method in pattern 4*

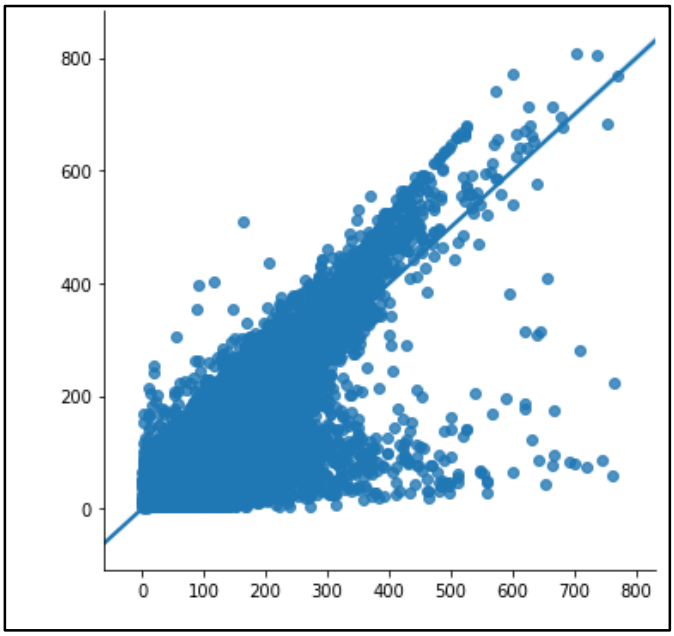


Fig. 81 *Correlation between test data and prediction data by GLR method in pattern 5*

8.3 INTERPRET THE RESULTS, MODELS, AND PATTERNS

Pattern-1, 3 and 5 are very important to model the target. In the first one all data was used as an input. the output by using Multi Linear regression algorithm are shown as a Fig. 77, Fig. 79 and Fig. 80. On the other hand, Fig. 78 and Fig. 81 display the correlation between original and predicted

PM25 by only climate data. The first pattern has the best correlation but it used different kind of data. But the second pattern, the input is only normal climate data such as temp, pressure, wind speed and prediction of air pollution parameters such as PM25 by correlation about 0.2 is not recommended. At the end only pattern 1, 3 and 4 are suitable with high correlation.

8.4 ASSESS AND EVALUATE RESULTS, MODELS, AND PATTERNS

The tables below categorize pattern 1 and 5 together and based on that Multi Linear Regression by using all air pollution and climate data works better than the others (pattern 1 and model 1). However, based on the result in previous chapter, PM25 could be predicted only by climate data with correlation about 0.2. this achievement was very useful and beneficial for all organization to predict the air pollution by some usual weather and air pollution parameters which can be measured easily.

Patterns	*method*	*data*	*correlation*
Pattern_1	Multi Linear Regression	Pollution & climate data	0.84
Pattern_2	Multi Linear Regression	climate data	0.18
Pattern_3	Multi Linear Regression	pollution	0.82
Pattern_4	Generalized Linear regression	Pollution & climate data	0.6
Pattern_5	Generalized Linear regression	pollution	0.62

8.5 ITERATE PRIOR STEPS (1 – 7) AS REQUIRED

In the pattern-1 and 3 used Multi Linear regression. Determining the correct input is very important to reduce the error. Different combination of input data was tested. However, the result shows by using PM10 in every combination the accuracy will be increased so the best combination of input data is including PM10. This investigation showed us the best final and suitable input data is combination of climate and air pollutants together.

Pattern_investigation

```
]: assembler_pv = VectorAssembler(inputCols=['PM10','NO2'],outputCol="pv")
   output_pv = assembler_pv.transform(dfx)
   final_datapv = output_pv.select("pv",'PM25')
   train_datapv,test_datapv = final_datapv.randomSplit([0.7,0.3])
   train_datapv.describe().show()
   lrpv = LinearRegression(featuresCol='pv',labelCol='PM25', predictionCol='prediction')
   lrmpv = lrpv.fit(train_datapv)
   print("Coefficients: {} Intercept: {}".format(lrmpv.coefficients,lrmpv.intercept))
   test_resultspv = lrmpv.evaluate(test_datapv)
   print("R2: {}".format(test_resultspv.r2))
```

```
+-------+-----------------+
|summary|             PM25|
+-------+-----------------+
|  count|           192540|
|   mean| 75.7574010595201|
| stddev|68.05294714621544|
|    min|              2.0|
|    max|            533.0|
+-------+-----------------+

Coefficients: [0.73141578757579,0.20753770186965992] Intercept: -8.930164413505402
R2: 0.783020957549277
```

9 REFERENCE:

Chan, C.K. & Yao, X. (2008). *Air pollution in megacities in China. 42*, 1–42.

Chen, W.Y. & Xu, R.N. (2010). *Clean coal technology development in China*. Energy Policy, *38*, 2123–2130.

Dayan, P. (1999). Unsupervised Learning. The MIT Encyclopedia of the Cognitive Science.

Du, X., Kong & Q., Ge & W., Zhang, S. & Fu, L. (2010). *Characterization of personal exposure concentration of particles for adults and children exposed to high ambient concentrations in Beijing, China*. Journal of Environmental Sciences, 22(11), 1757-1764. DOI:10.1016/s1001-0742(09)60316-8

Elminir, H.K. (2005). *Dependence of urban air pollutants on meteorology*. Sci. Total Environ, 350, 225–237.

Holloway, T. & Spak, S.N. & Barker, D. & Bretl, M. & Moberg, C. & Hayhoe, K. & Van Dorn, J. & Wuebbles, D. (2008). *Change in ozone air pollution over Chicago associated with global climate change.* J. Geophys. Res. Atmos. 113,DOI:10.1029/2007JD009775.

Knowledge Center .*com/support/knowledgecenter/en/SS3RA7_sub/modeler_mainhelp_client_ddita/clementine/understanding_modeltypes.html*

James, G. (2015). An introduction to statistical learning: With applications in R.(pp. 337-372) New York: Springer.

Kurt, A. & Oktay, A.B.(2010). *Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks*. Expert Syst. 37, 7986–7992.

Li, L.J. & Wang, Y. & Zhang, Q. & Li, J.X. & Yang, X.G. & Jin, J. (2008). *Wheat straw burning and its associated impacts on Beijing air quality*. Sci. China Ser. *51*, 403–414.

Microsoft Neural Network Algorithm, (2018). https://docs.microsoft.com/en-us/analysis-services/data-mining/microsoft-neural-network-algorithm?view=asallproducts-allversions

Rohde, R. A., Muller, R. A. (2015). Air Pollution in China: Mapping of Concentrations and Sources. Plos One, 10(8). doi:10.1371/journal.pone.0135749

Samet, J.M.& Zeger, S.L. & Dominici, F. & Curriero, F. & Coursac, I. & Dockery, D.W. & Schwartz, J. & Zanobetti, A.(2000). *The national morbidity, mortality, and air pollution study. Part II: Morbidity and mortality from air pollution in the United States*. Res. Rep. Health Eff. Inst. 94, 5–79.

Sun, Y.L. & Zhuang, G.S. & Tang, A.H. & Wang, Y. & An, Z.S. (2006). *Chemical characteristics of PM25 and PM$_{10}$ in haze-fog episodes in Beijing*. *Environ. Sci. Technol. 40*, 3148–3155.

Wang, Q.Q. & Shao, M. & Liu, Y. & William, K. & Paul, G. & Li, X.H. & Liu, Y.A. & Lu, S.H. (2007) *Impact of biomass burning on urban air quality estimated by organic tracers: Guangzhou and Beijing as cases.* Atmos. Environ, 41, 8380–8390.

Wei Chen. (2015). Air Quality of Beijing and Impacts of the New Ambient Air Quality Standard. 6, 1243-1258. DOI:10.3390/atmos6081243

World Health Organization, (2020). Air Pollution. Retrieved from https://www.who.int/health-topics/air-pollution#tab=tab_2

Yuan, Z. & Zhou, X. & Yang, T. & Tamerius, J. & Mantilla, R. (2017). *Predicting Traffic Accidents Through Heterogeneous Urban Data: A Case Study*. In Proceedings of the 6th International Workshop on Urban Computing, Halifax, NS, Canada.

Zheng, Y. & Liu, F. & Hsieh, H.-P. (2013). *When urban air quality inference meets big data*. In Proceedings of the19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA,