Iteration 4 – BDAS

1. **Situation Understanding**
   1. **Situation objectives identification:**

With the rapid development of our world, the quality of human beings’ lives, and living standards are improving to a quite high level. However, the truth is that there are still existing various kinds of problems that influence the world we are living in. In order to solve those problems, the United Nations proposed ’17 Sustainable Development Goals.’ In this case, we focus on the air quality problem, which belongs to multiple goals, such as the 3rd ‘Good health and well-being’ because of the low-quality air is harmful to us; also, the 13th ‘Climate action’ as the impurity in the air may cause climate change – in a bad way. The objective of this case is trying to discover, find and summarize the possible pattern of the air quality data sets, and even predict the future quality of air – in other words, by utilizing those data sets and mining the underlying pattern, we may find a better way to help the world and the public to understand the change of the air that we breathe every second, and take actions to protect the air.

* 1. **Situation Assessment:**

**Sources of Task:**

**Data source**: The datasets are available from Kaggle and the Taiwan Environmental Protection Administration Executive Yuan.

**Hardware source**: Personal laptop, Amazon Web Service.

**Software source**: GitHub Desktop, Putty, Jupyter Notebook.

The current situation our air quality is not that optimal. The air quality has always been a serious problem for us, especially some developing countries. In Taiwan, the air quality indicators, no matter the PM2.5 or PM10 or other chemicals (nitric oxide), all of them indicate that the people are breathing impurity such as ozone, different respirable particulate matters, and so on. Those impurities are no doubt harmful to human beings.

For the purpose of analysis, the air quality monitoring data is indispensable. Usually, the data file data has various kinds of type - the numerical such as numbers and dates, the categorical like locations and weather-type. In general, all the data can be used during the analysis process. However, the risk of this is, the data we used is not always ‘clean,’ and this unclean situation may lead to the inaccurate model generation which is used to make prediction – if the model we have is not accurate, consequently, the predicted results are also not precise, in some special occasion, this inaccuracy may cause serious consequence. The possible solution is, at the very beginning we will execute the operation ‘data cleaning,’ and we will try several different algorithms to make sure that we can generate the optimistic model in a certain way.

* 1. **Data Mining Objectives:**

In general, the goal is to find the potential pattern among the attributes and try to find a way to make a prediction and even get a way to control air pollution.

To help our society to understand the air quality data, we need to analyse the oceans of past monitoring air data. The data mining objectives are mainly focused on finding the different types or kinds of relations between the numerous data attributes. By analysing those data, such as the different numbers of different indicators in various time slots. We may find and summary a pattern that when air pollution is most serious, and which kind of pollutant contributes the most. During the mining process, several different algorithms may be utilised to test for some diverse model, and finally, choose the one who can describe the past data and optimally predict the future air quality.

* 1. **Project Plan:**
     1. **Project Background**

As above, to achieve the United Nations’ sustainable development goals – ‘Good health and well-being,’ and ‘Climate action,’ the society need to pay more attention to the air that is crucial and indispensable for us. Based on this, we need to collect, process and then analyse the air quality monitoring data about the past. After analysis, expectable we will get a pattern of these data, and ideally, we can utilise this model/pattern to predict the future air quality and also, take action on air quality control to provide a better environment for human beings.

* + 1. **Tasks**

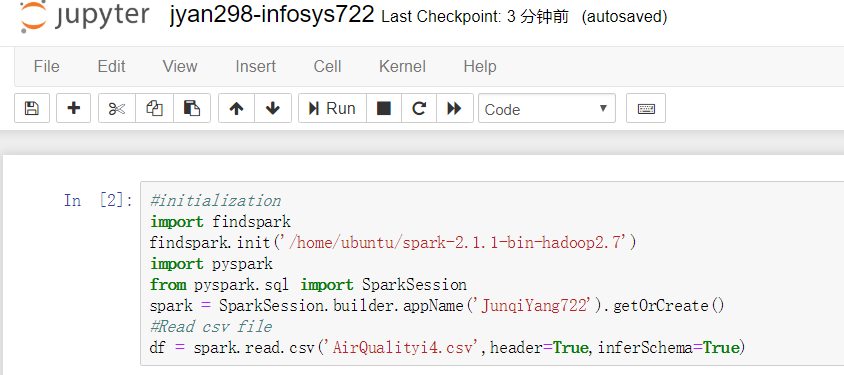
|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Time (ETA)** | **Resources** | **Risks** |
| Situation understanding | 6 hrs. | Data analysts | Environment change, criteria change |
| Data understanding | 12 hrs. | Data analysts | Data problems, technology problems |
| Data preparation | 1 day. | Data analysts, and specific software/tools | Data problems, technology problems |
| Modelling | 1 day. | Data analysts, specific software/tools | No model can fit the requirements |
| Evaluation | 6 hrs. | Data analysts | Environment change, criteria change |
| Deployment | 3 hrs. | Data analysts | Environment change, criteria change |

1. **Data Understanding** 
   1. **Initial Data Collection:**

In this phase, we will collect the data sets that we need to perform data mining operation. There are plenty of ways for us to gather data – for an organisation, the data within the organisation should be collected regarding the current data; also, sometimes we may need to purchase additional data from organisations because our data set is incomplete. If still there are not enough data sets, we need to research and add more data to the data sets.

There are thousands of websites (governments, organisations and so on) provide the data sets to download. In this case, the data can be collected from the official websites or other specialised data sets websites such as Kaggle. For this task, the data sets we collected in this case, it was downloaded via Kaggle.

In this case, we can import the dataset by using pandas or PySpark. Here I import the csv file by pandas first, and before modelling, I will transform the pandas dataframe to PySpark dataframe.





Data Import (Same file, different name)

* 1. **Data Description:**

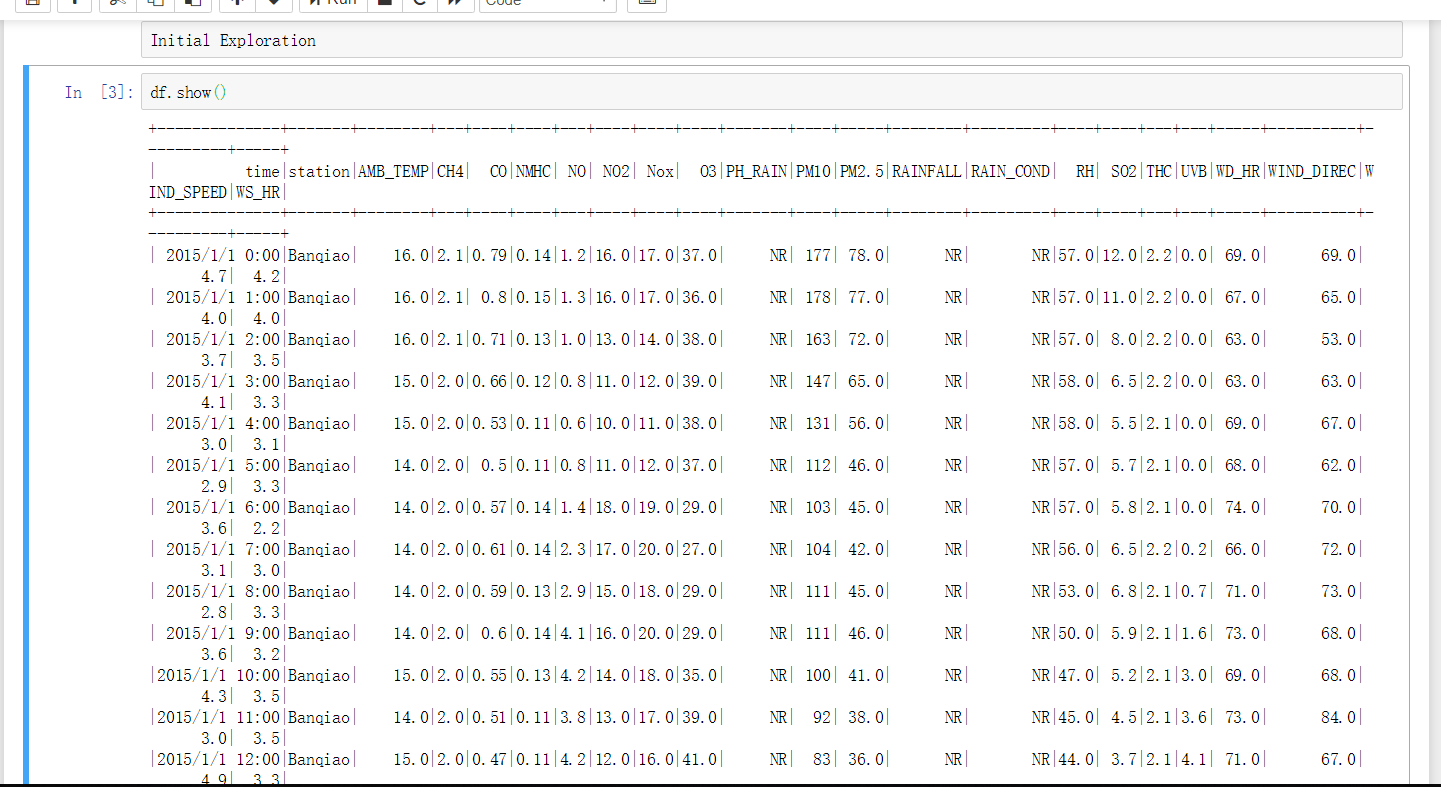
Dataset Scale: 23 attributes, nearly 220k records.

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Type | Role |
| Time | Time records | Continuous | None |
| Station | The station that recorded the data | Categorical | Input |
| AMB\_TEMP | Temperature | Continuous | Input |
| CH4 | CH4 index | Continuous | Input |
| CO | CO index | Continuous | Input |
| NMHC | Non-methane hydrocarbons index | Continuous | Input |
| NO | NO index | Continuous | Input |
| NO2 | NO2 index | Continuous | Input |
| NOx | NOx index | Continuous | Input |
| O3 | O3 index | Continuous | Input |
| PH\_Rain | The PH value | Categorical | Input |
| PM10 | PM10 index | Continuous | None |
| PM2.5 | PM2.5 index | Continuous | Target |
| Rainfall | Rainfall index | Continuous | Input |
| Rain\_Cond | Conductivity index | Continuous | Input |
| RH | Relative-humidity index | Continuous | Input |
| SO2 | SO2 index | Continuous | Input |
| THC | Total-Hydrocarbon index | Continuous | Input |
| UVB | UVB index | Continuous | Input |
| WD\_HR | Wind direction/hour | Continuous | Input |
| Wind\_Direction | Wind direction | Continuous | Input |
| Wind\_Speed | Wind speed | Continuous | Input |
| WS\_HR | Wind speed/hour | Continuous | Input |

Considering the methods of air monitoring, the air quality records will be a large number. The scale of our data set has 23 columns and approximately 220k records and is stored in a CSV file, so I can use the read\_csv() function to load data. The majority of the value types are numerical and categorical, such as date, time, numbers, location, and so forth.

* 1. **Data Exploration:**

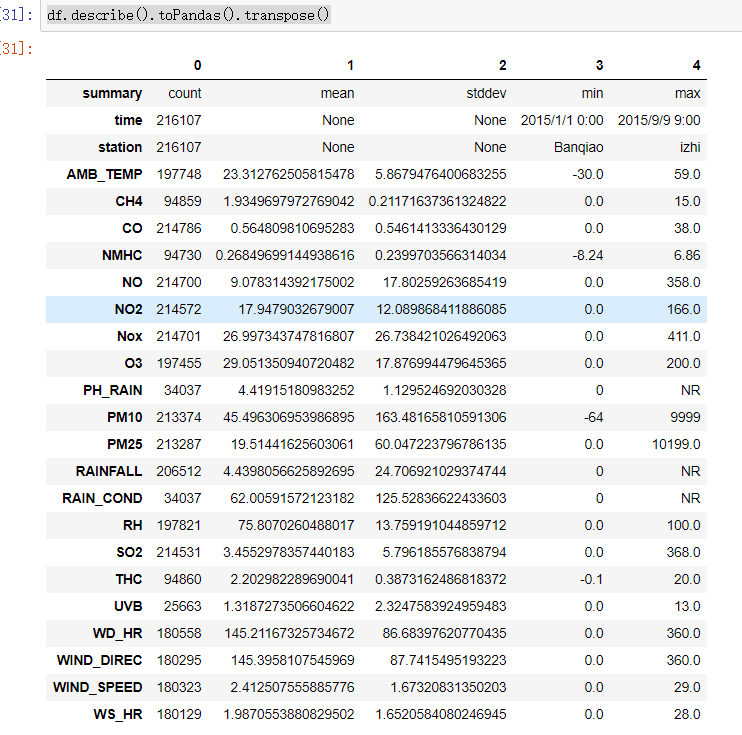
There are a lot of functions can be used to explore the dataset initially. Below is the screenshot of the result.



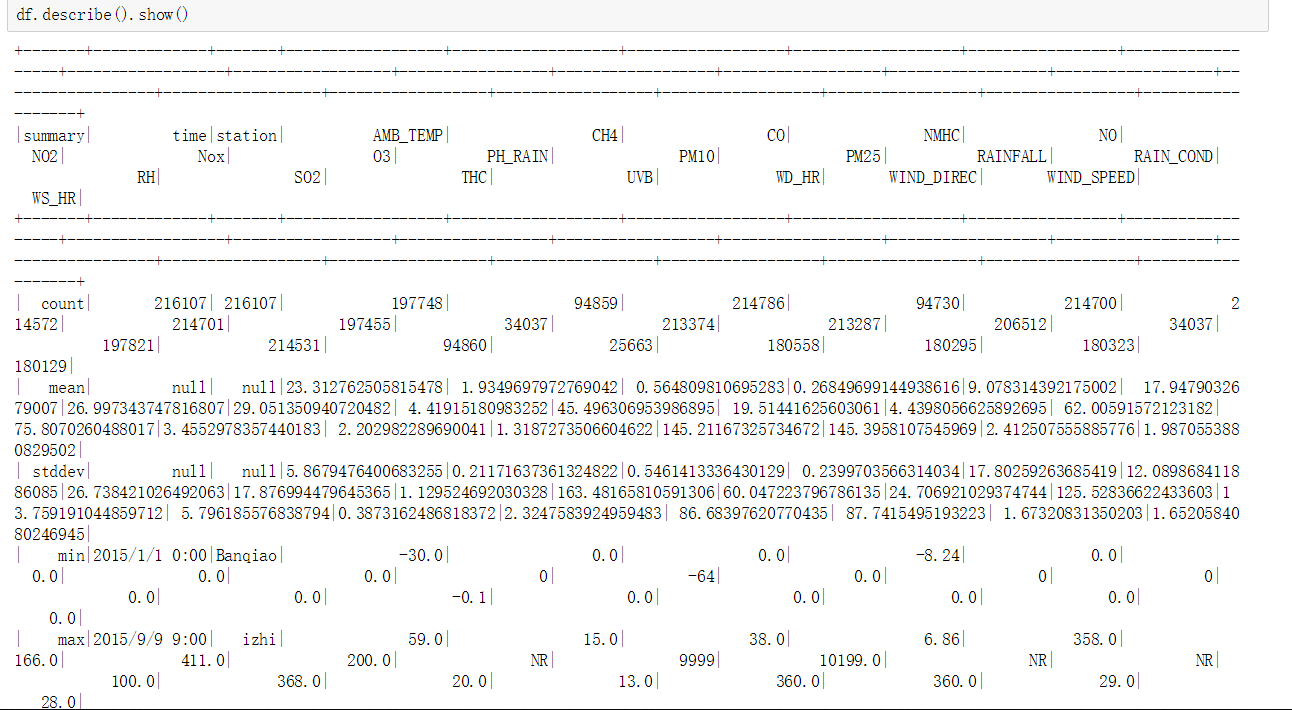
Initial exploration of Data – show() function



Schema of dataset

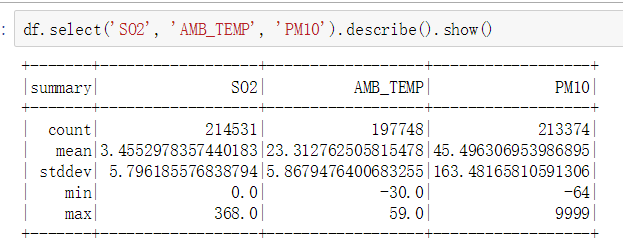


Describe() function



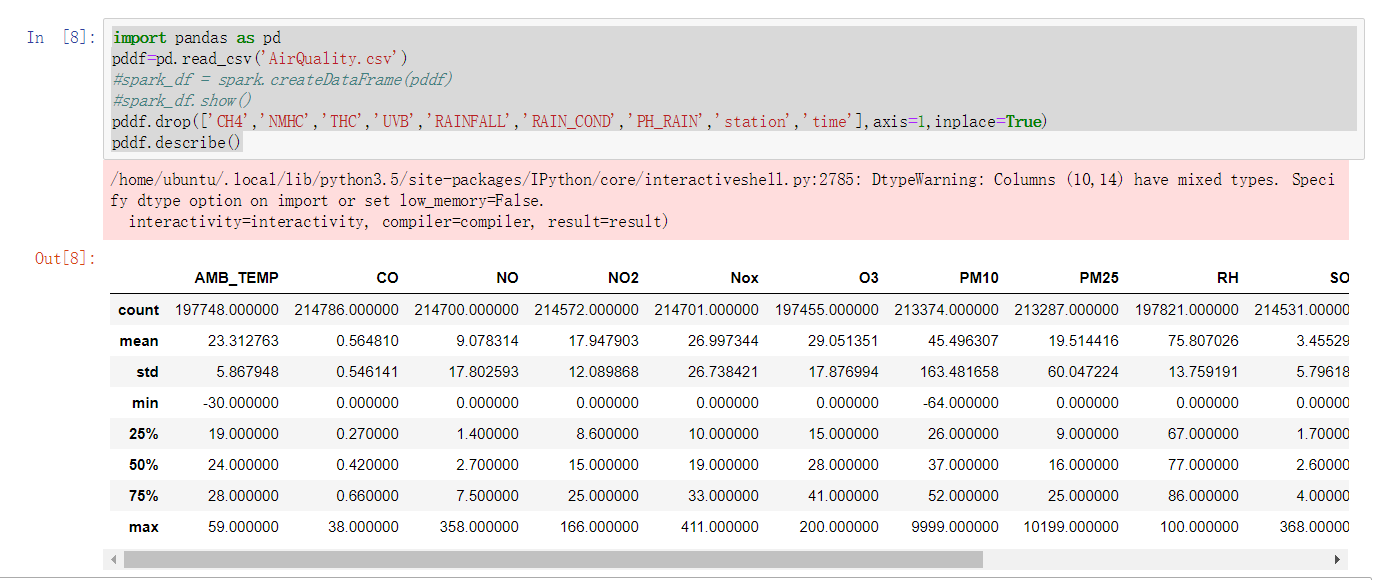
Describe of all attributes

Because this dataset has too many attributes, so the output of the describe() function is a little messy, but I will choose some attribute to apply the describe() function.



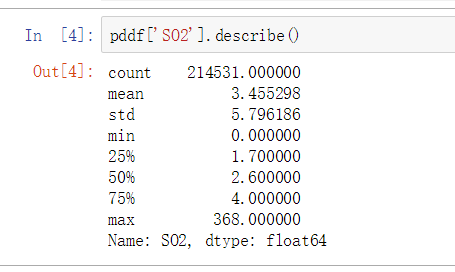
PySpark describe() function

Now I use the pandas library to describe the dataset.



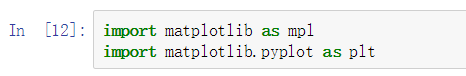
Pandas describe()

We can see the pandas provides us with a clearer view of the dataset. Hence, in the future iteration, using pandas library to explore the dataset initially is a better choice (pyspark.mlib is still used for modelling).



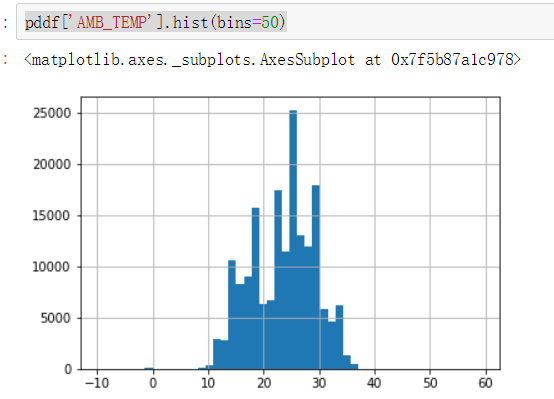
Pandas describe() function of the specific attribute

To further explore the dataset, here I will import the matplotlib to visualise the dataset.



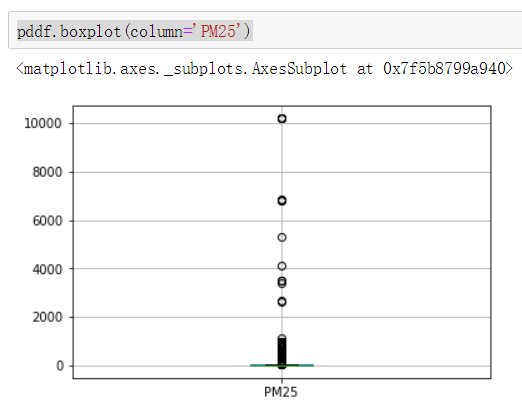
Import matplotlib

The matplotlib is a very powerful library to visualise the data.

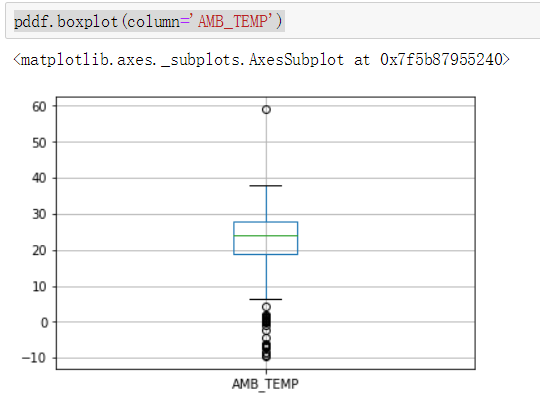


Histogram of AMB\_TEMP

Also, the boxplot can help us to check the quality of our dataset initially.



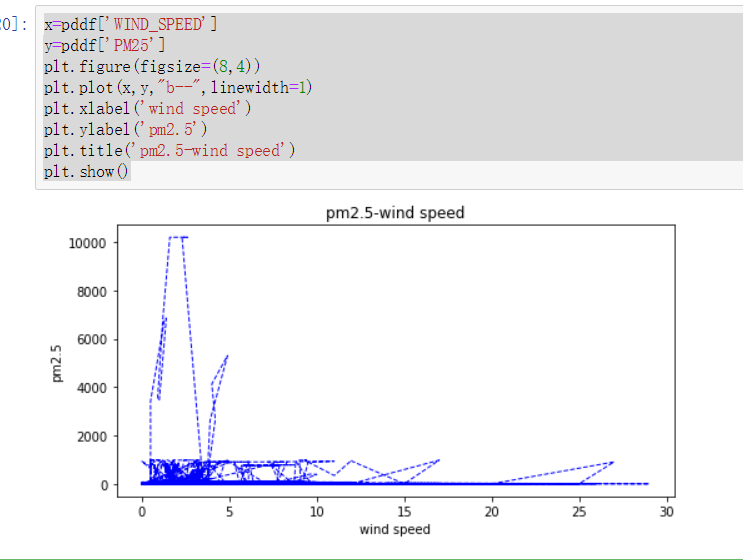
PM2.5 Boxplot

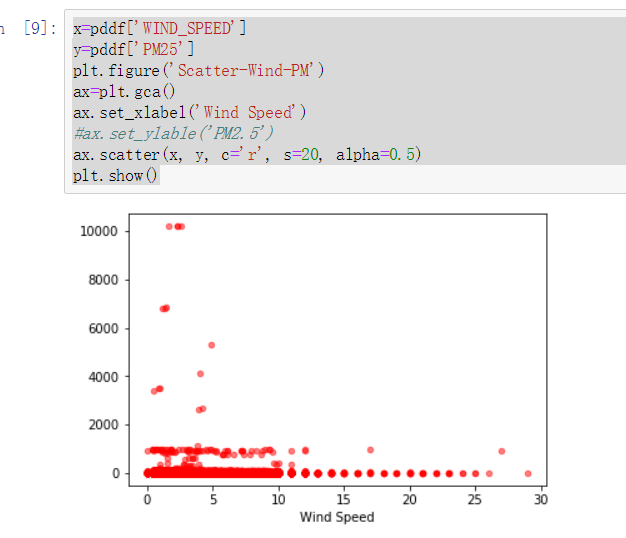


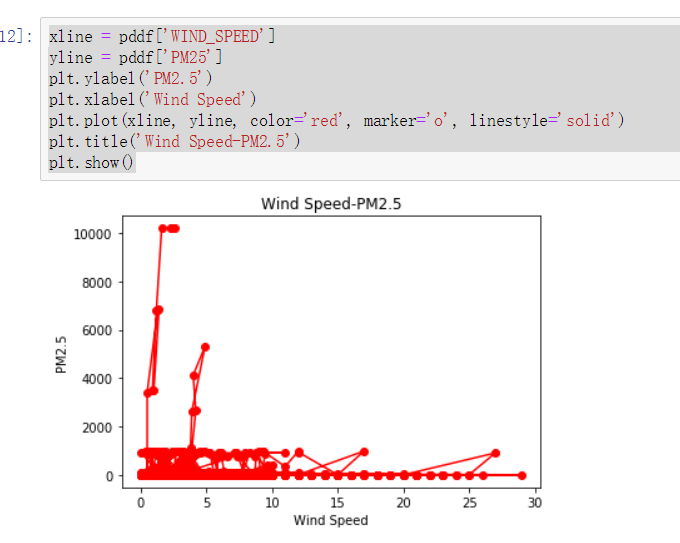
AMB\_TEMP Boxplot

The boxplot can help us visualise the outliers.

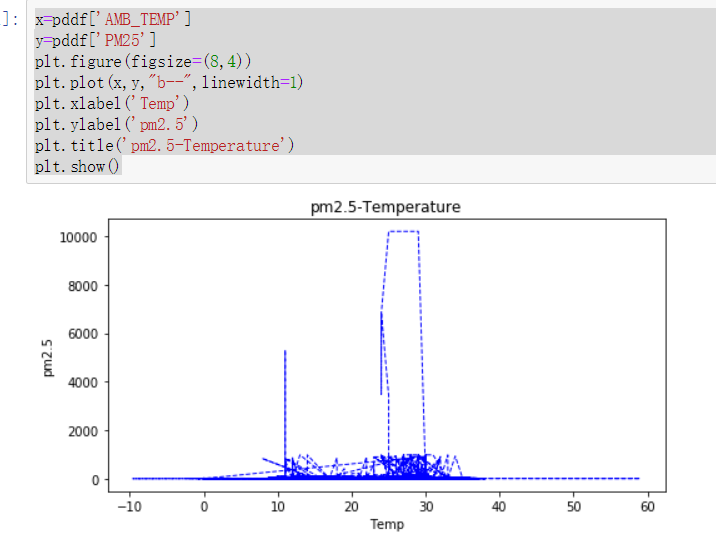
I also try to discover the relationship between wind speed and pm 2.5.



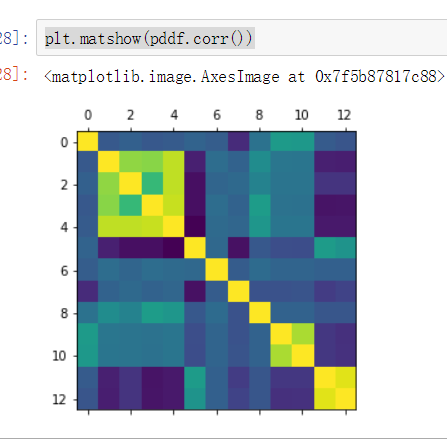




PM 2.5 – Wind Speed



Temperature – PM 2.5

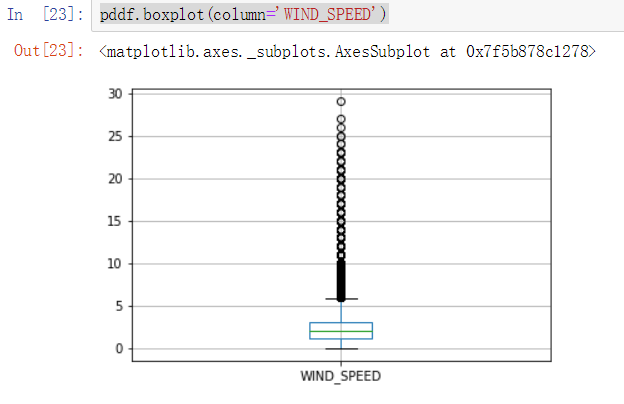
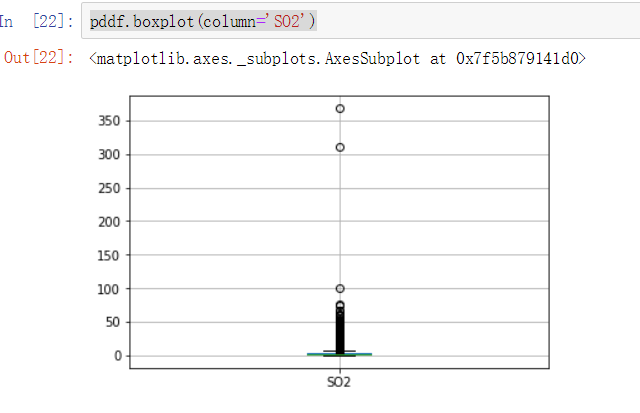
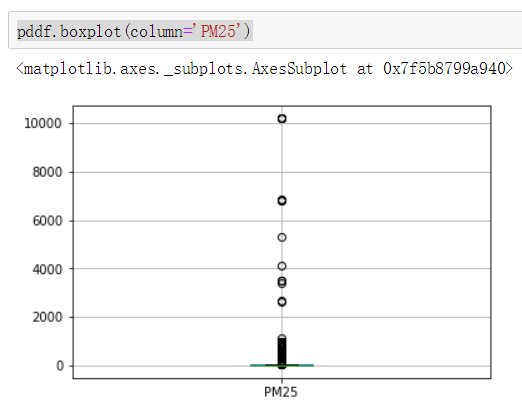
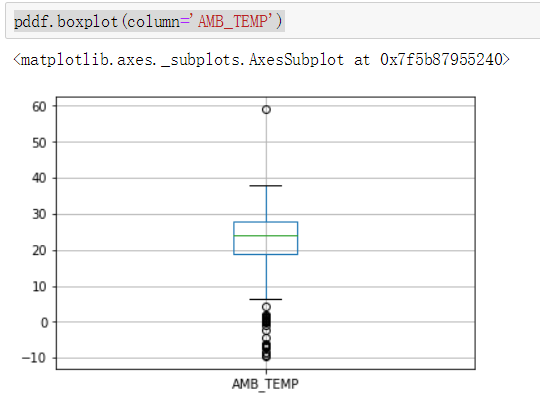


Correlation Heatmap

The initial graphs seem very hard to extract information from it. It indicates us that our data need further cleaning.

* 1. **Data Quality Verification:**

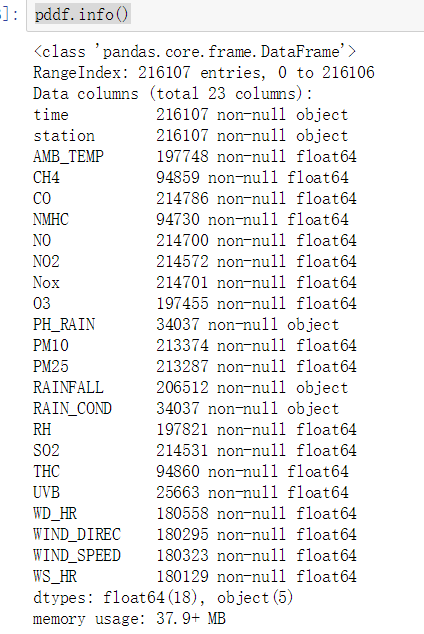
The dataset with a huge amount of records is always not perfect. There are a lot of unexpected data errors within the data set, in our air quality dataset file, there are some missing values (include missing values, ‘NR’ values), and in some attributes, the values are mixed with multiple types of data, and some records seem are not correct. All of those phenomena are not good for our data mining operation. For a better data mining result, the data set needs to be ‘cleaned.’ Within this dataset, there are some attributes have missing values, and in order to further discover the dataset, we can use the boxplot graph to find the outliers.



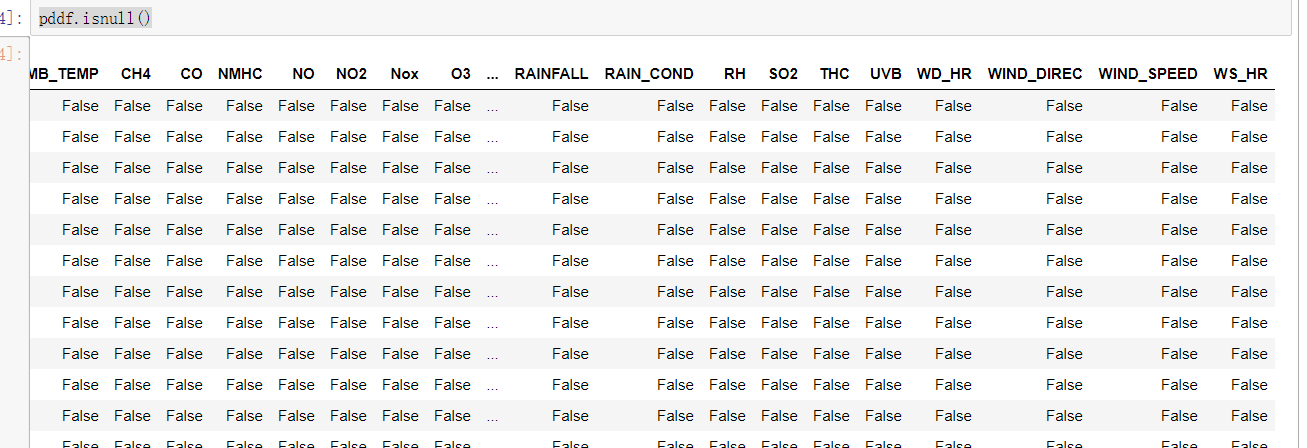
Boxplot of some attributes

From the chart, we can see there are some of the values are abnormal, and some attributes have a lot of outliers.

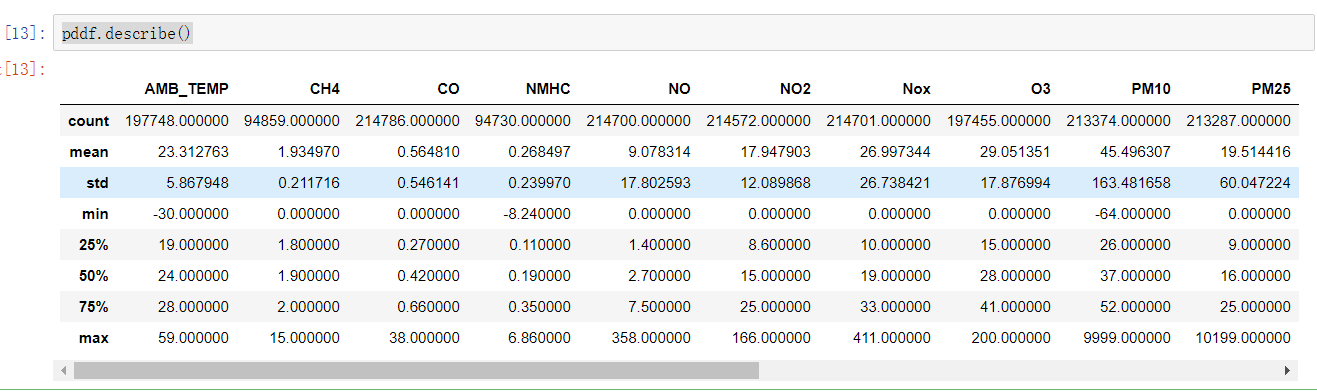
In addition, we need to check whether there are some null values in the dataset. I use the isnull() function, and the info() function to initially check the dataset.



Info() function



IsNull() function



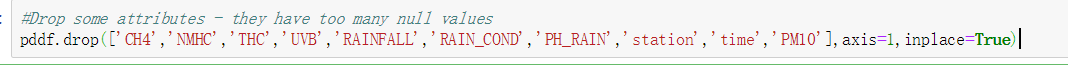
Describe() function

Here the table shows there are null values in some attributes, and there are some unexpected values because some attributes which are supposed to be numeric (float/double) now are identified as objects.

So, we need to deal with these values in the following steps.

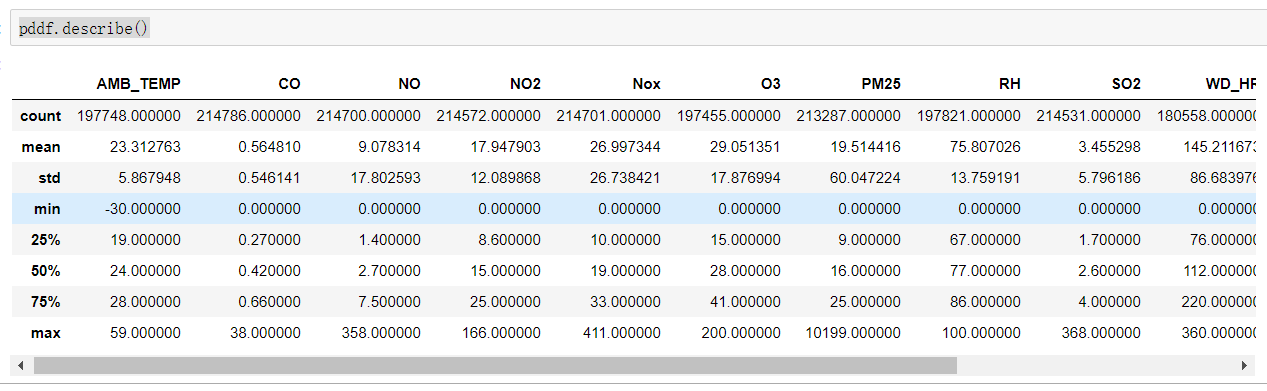
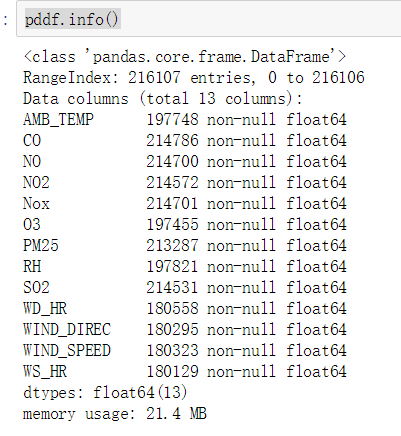
1. **Data Preparation**
   1. **Data Selection**

According to the results I got from the previous steps, I found some of the attributes have too many missing values, if I choose to fill or delete them, it might seriously influence the entire dataset, so it is better to drop these columns.



Drop Columns

Those attributes have too many null values and so I dropped them. The pm10 has a too strong correlation with the target pm2.5 so I dropped it too.

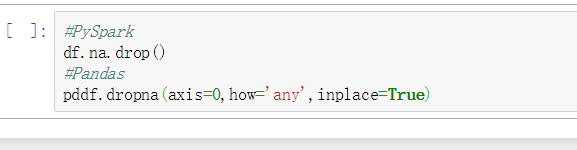


After dropping some columns

* 1. **Data Clean**

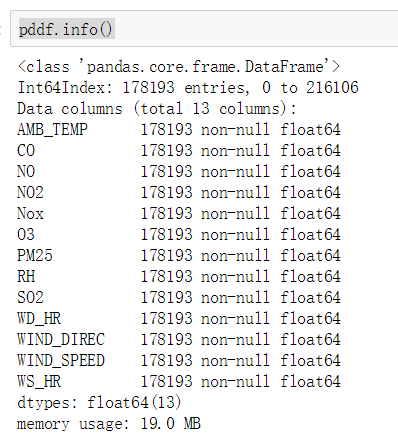
After the data selection phase, now we are going to clean the dataset. From the previous results, we can notice that most of the attributes’ values are valid. Hence, now I will deal with the missing values in the rest attributes.

Both the PySpark and Pandas provides the null value cleaning functions, but it seems the pandas provides us a more comprehensive and convenient data cleaning function.



Pandas and PySpark Clean NA

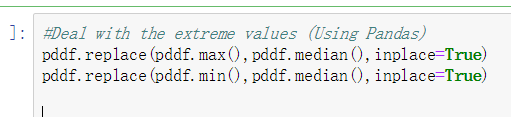
The dropna() function in Pandas and the na.drop() function in the PySpark could be used to clean the missing values.



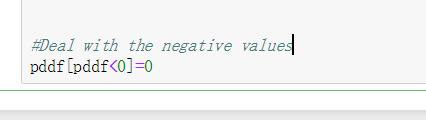
After drop NA

After the null values discard, the record in the dataset has been reduced to 178,193 from 216,707, and all the attributes are numbers now.

Next, I will deal with the extreme values and the negative values. However, this function is not very good as it only replaces one record once, so I will use a better function to further clean the dataset.

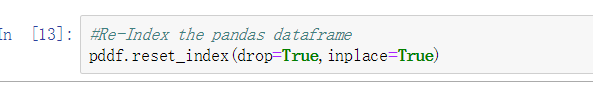


Extreme Values

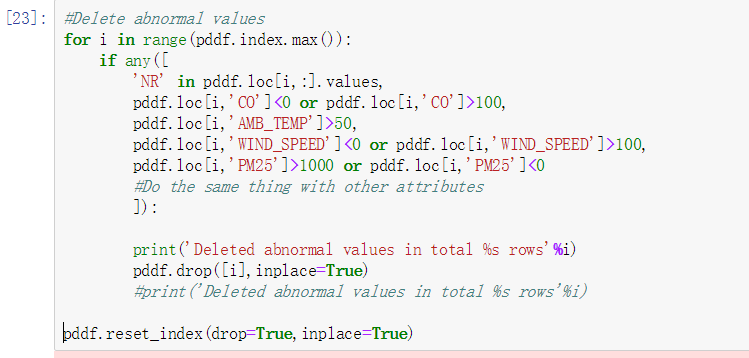


Negative Values

However, there is a better way to deal with these abnormal values. But before using this function, I need to re-index the dataframe first by using the reset\_index() function.

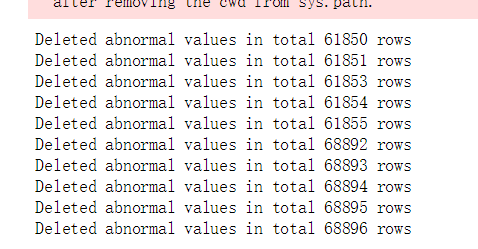


Re-Index



Delete abnormal values

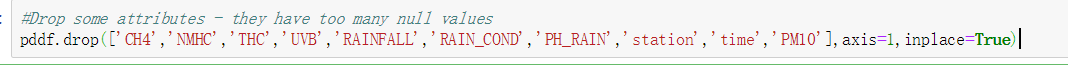
I wrote this function (it requires an ordered index of the dataset) to delete the abnormal values within every columns of the dataset, including the negative values, and some extreme values, but the range is decided according to the requirements of different datasets.

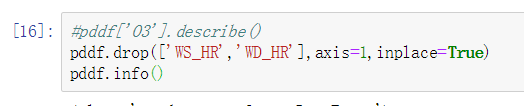


Deleted

* 1. **Data Construction**

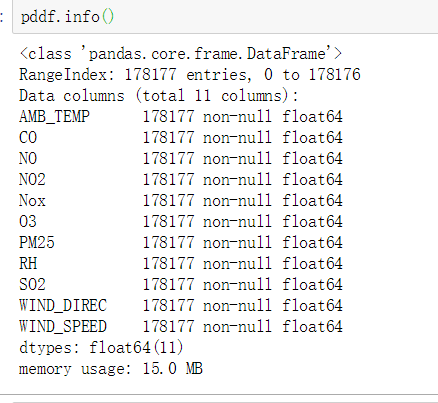
To construct the data, first, I observed that some of the attributes in the initial dataset having too many missing values, as a consequence, they may seriously influence our data mining process so that I will use the drop() function to discard them. As I only have one dataset, so there is no need to ‘merge’ different datasets.





Select columns

Below is the number of columns and rows I got after clean.

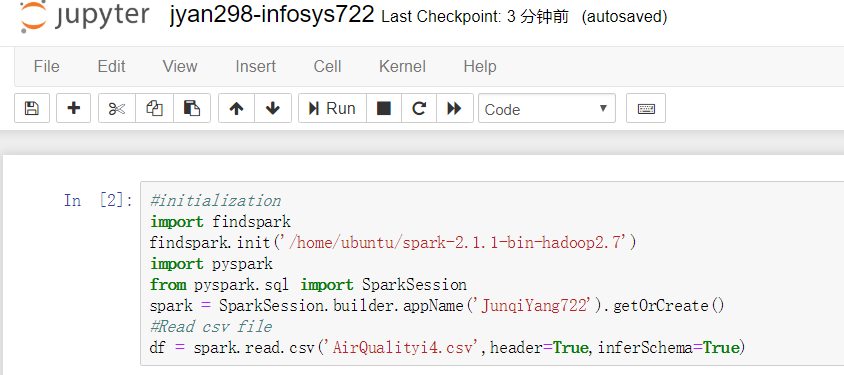


178,177 entries and 11 columns

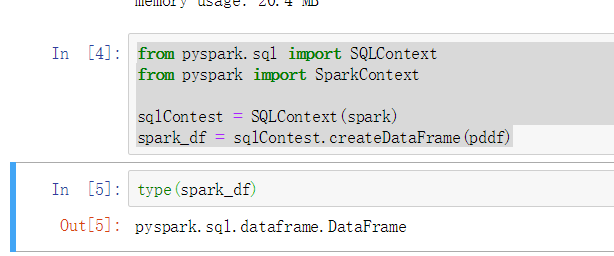
* 1. **Integrate Data Sources**

As we only have one dataset that is stored in one single CSV file, so we do not need to use the ‘Merge’ node which integrates several files together. The source file is imported at the beginning. But the pandas dataframe will be transformed to spark dataframe as the spark.mlib required.





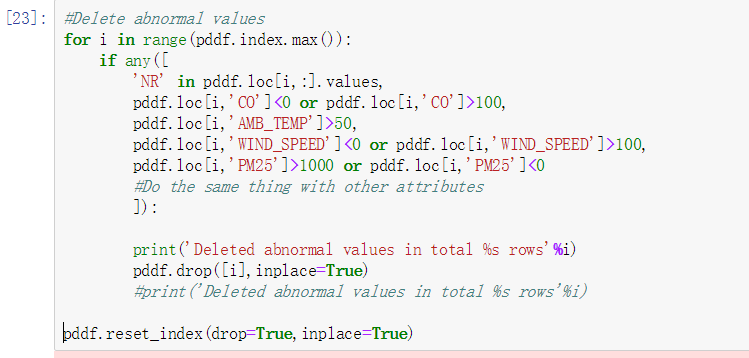
Source File Imported



Transformation

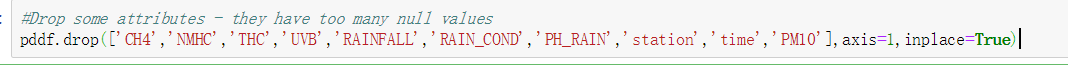
1. **Data Transformation**
   1. **Reduce the Data**

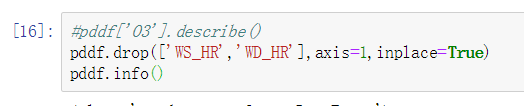
By the observation of the dataset contents, we found that some attributes have too many blank values (‘NR’) or Null values, and compared to the entire dataset these attributes cannot be easily ‘processed’ with the assurance of data quality, also, the Pm10 has too strong correlation with the target Pm2.5, so I decided to drop it too. To make sure the accuracy of further data mining process, we decide to discard these attributes – reducing the dataset.



Abnormal values Dropping

By using the built-in functions, I can easily remove some attributes.

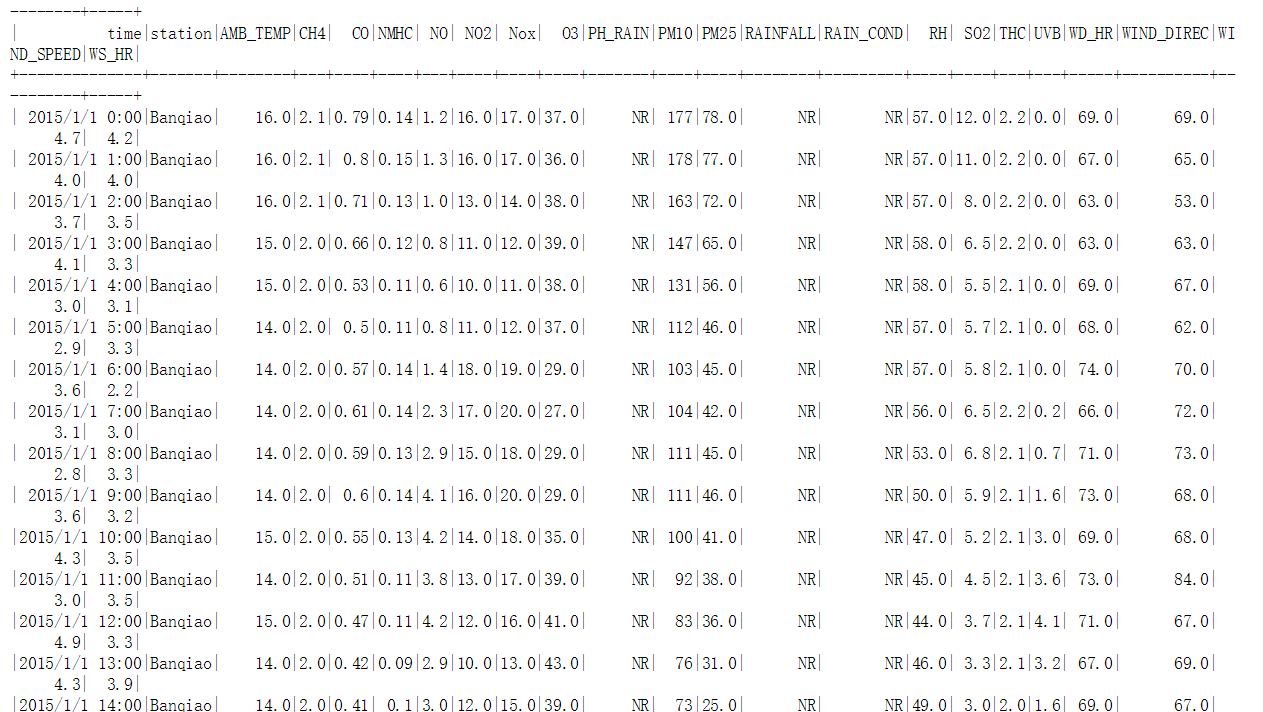




Reduce Columns

Also, we drop the PM10 as it has too strong but useless correlation with our target value, this is a part of feature selection.

In this case, the ‘NR’ value is the blank value which we want to discard. To some extent, we can reduce some noisy data through the data reducing phase.



NR Value Example

* 1. **Project the Data**

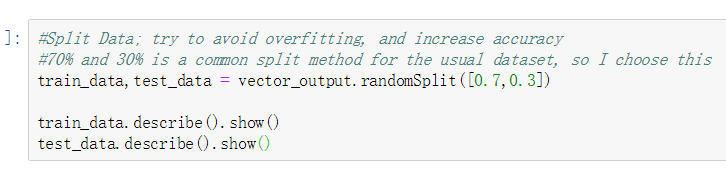
In this phase, I will make the projection of the data. With the randomSplit() function, I can split the data into separate subsets for the training, testing.

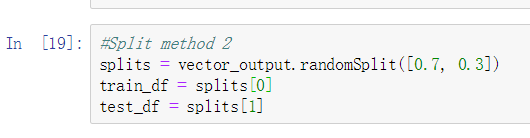
When splitting the dataset, if there is no common practice for determining the size of the training set and the test set, we generally choose 60:40, 70:30 or 80:20. For large scale datasets, 90:10 or even 99:1 is more common. It should also be noted that when the optimal model and parameters are obtained through local verification, the entire data set (training set + verification set + test set) is also trained once to obtain the final model.

*Training Set*: Help us train the model. Simply put, let us determine the parameters of the fitted curve through the data of the training set.

*Test Set*: To test the accuracy of a model that has been trained. However, the test set does not guarantee the correctness of the model. It just indicates that similar data will get similar results with this model. Hence, when we train the model, the parameters are all corrected and fitted according to the data in the existing training set. It is possible that there will appear over-fitting, that is, this parameter only fits the data in the training set more accurately. At this time, there is another data that needs to use the model to predict the result, and the accuracy rate may be very poor.

By using this function, we can avoid the appearing of the ‘overfit’ phenomenon at a certain degree, as we test and validate the dataset, so the dataset will not be constrained in an over-specified environment.





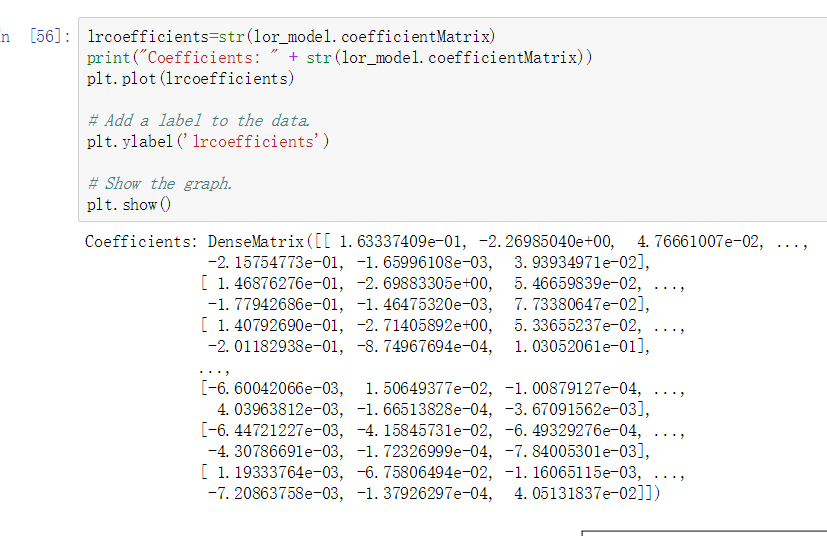
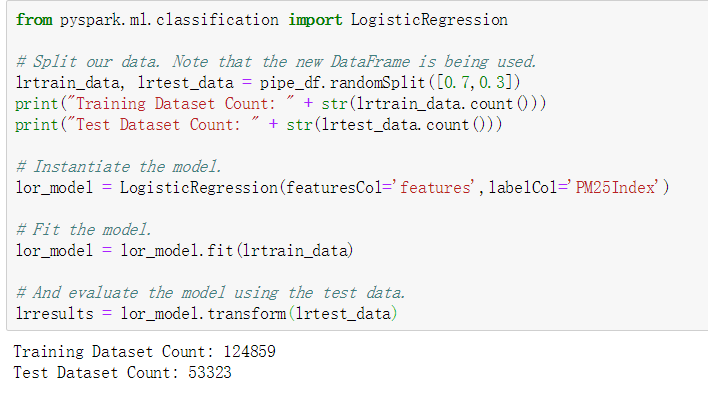
Data Partition

1. **Data-mining Method Selection**
   1. **Match and discuss the objectives of data mining to data mining methods**

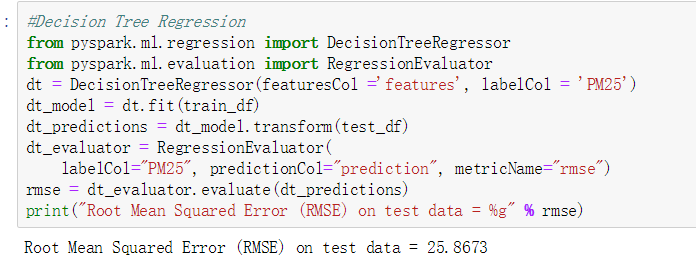
In this case, our goal is discovering the relationship between the various kinds of chemicals in the air and the PM values. Also, we want to know whether we can predict the PM value if we have the other pollutants’ values, they are our objectives.

In general, there are three types of methods – classification, clustering, and regression. The classification is the task of approximating a mapping function from input variables to discrete output variables; its output is usually named label, some classical classification algorithms such as Naïve Bayes, decision tree, and support vector machine. The regression is the task of approximating a mapping from input variables to a continuous output variable, in other words, numeric values. The classical regression algorithms include linear regression, regression tree, and logistic regression. The clustering method is an unsupervised algorithm, if we don’t know our target yet, we can use the clustering, such as K-mean algorithm, Hierarchical clustering, and Gaussian mixture model.

In my dataset, both the target attributes and the other attributes are numeric values, so it is better for this case to use the regression method to do the data mining work. Here I tried the classification, but the result is not very good.



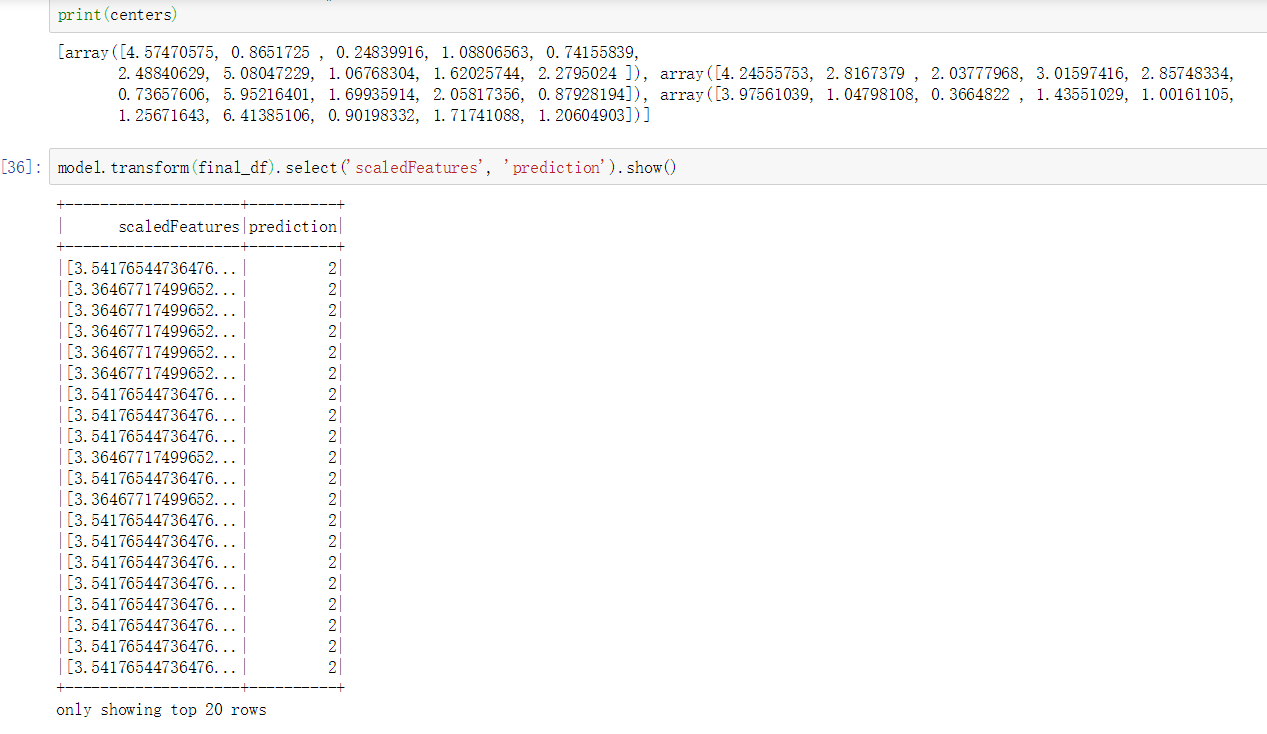
Logistic Regression



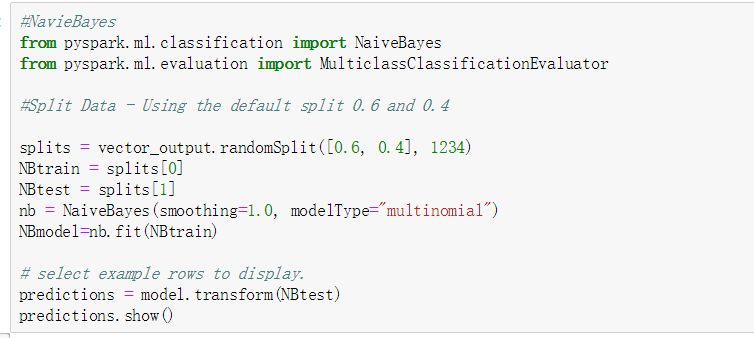
Decision Tree

Also, the clustering algorithm is unsupervised, which means if we have no clear target in terms of the data mining process, the clustering algorithm will be a better choice to find the potential pattern within the dataset. However, in this case, my target is the ‘PM2.5’, as I have a target, so here I will not use the clustering algorithm.





Clustering – K-Means Example

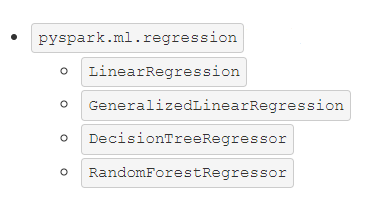
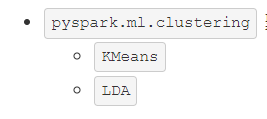
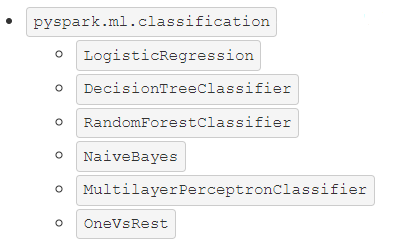


Naïve Bayes Example

Based on the situation of my dataset, all the attributes are numbers. The regression algorithms are good at dealing with numbers. Hence, I will use the regression algorithms for this dataset.

* 1. **Select the appropriate data-mining method(s) based on discussion**

Within the PySpark machine learning package, we can find some machine learning algorithms are available.

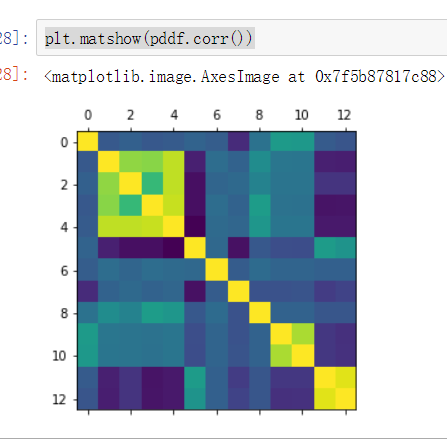


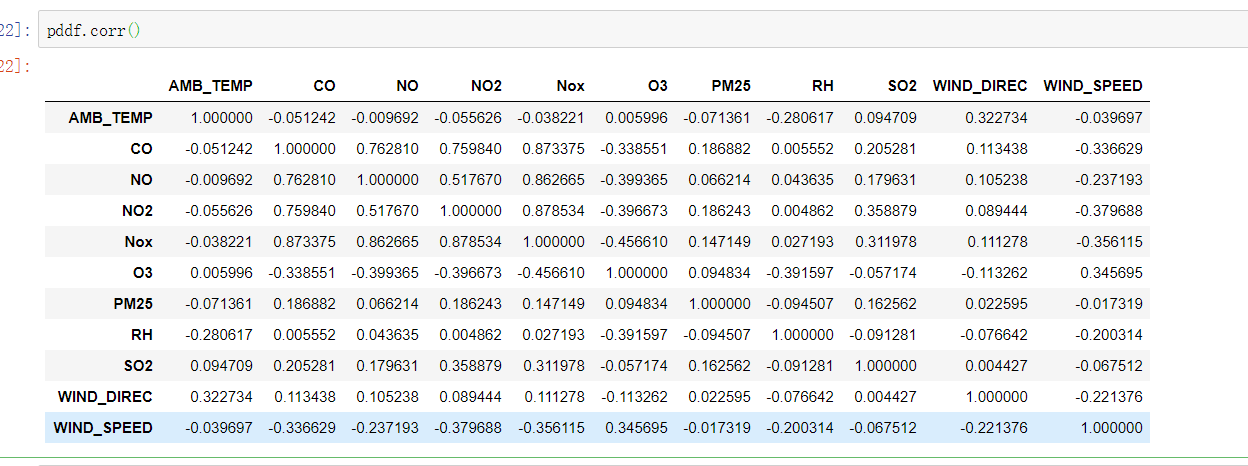
Some of the PySpark Algorithms

There are Linear Regression, Generalized Linear Regression, Decision Tree Regressor, and Random Forest Regression under the regression algorithms. As the discussion above, the clustering and classification are both not very suitable for this dataset, so here I choose the regression algorithms, I will try different regression algorithms and find a better one based on their results.

1. **Data-mining Algorithm(s) Selection**
   1. **Conduct exploratory analysis and select algorithms**

To begin with, here I can use a function which called corr() to analyse the dataset initially.





Correlation

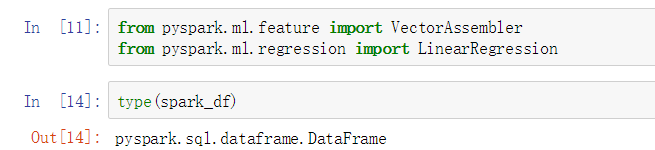
From the screenshot above, we can notice that the attributes have correlations to a certain degree, so we can use the regression algorithms, however, because the classification and regression have some overlapping aspects, so in the future iteration I could also try some classification algorithms later.

* 1. **Select Algorithms**

First, under the ‘Regression’ tag we have several different regression algorithms in total. In this case, I will try the linear regression algorithm, decision tree regression, and G-Boost tree regression. In the future iteration, the other regression like Generalized linear regression, and Random Forest Regression can be used.

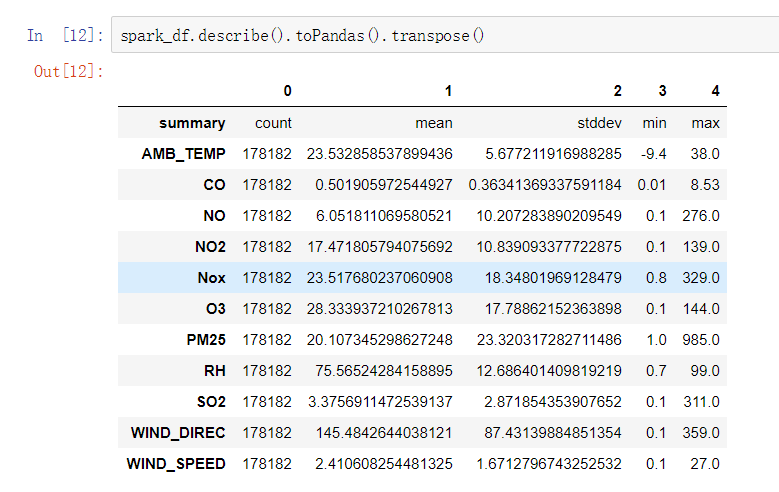
* 1. **Build/Select appropriate model(s) and choose parameters**

Now we will start running our models. To begin with, we need to import the necessary library. Also, the data frame has already been transformed at step 4.

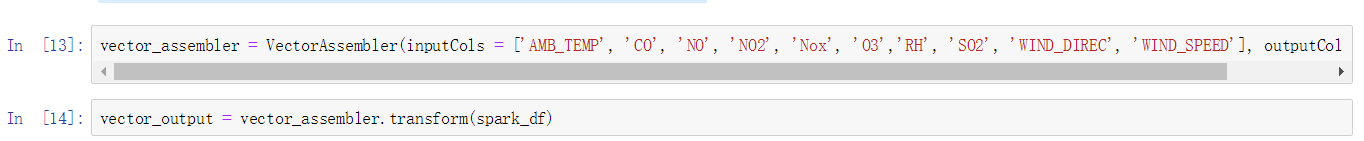


Linear Regression

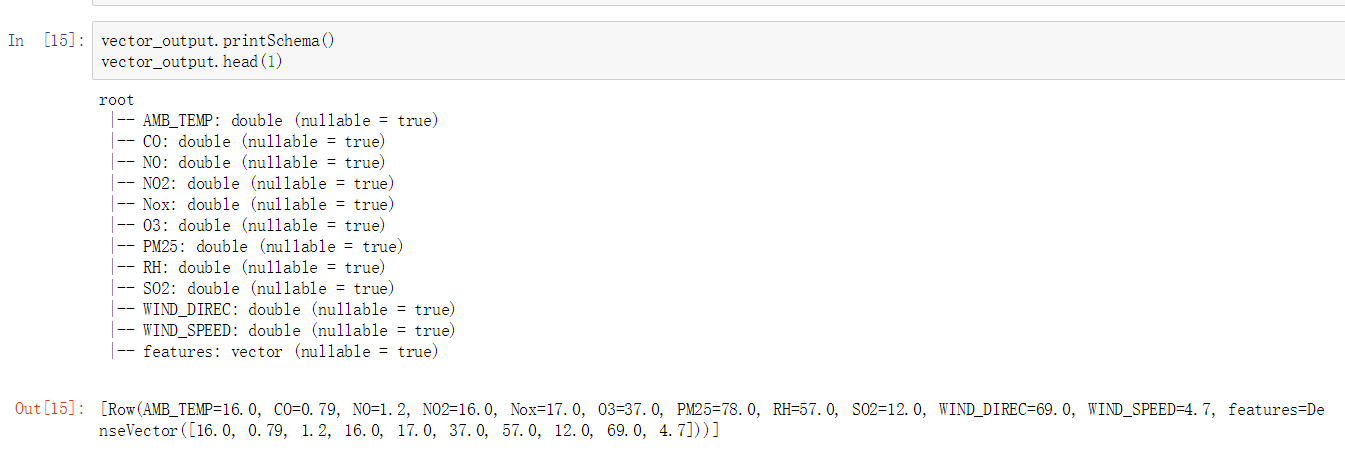
Next, I need to do some preparation work to start the linear regression modelling.



Describe result transformation



Set target and transformation

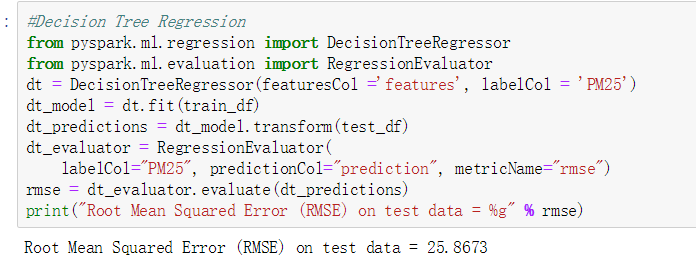


Status Check

The Vector Assembler function is used to select features from the dataset and combine them to a single output, which is required by the modelling functions after. Here the selected features are output as ‘features.’

First, we set the PM 2.5 as our target, and then I set up all the necessary preparation work for modelling. As we can see, a model stream has been built, and we can apply this to other algorithms.

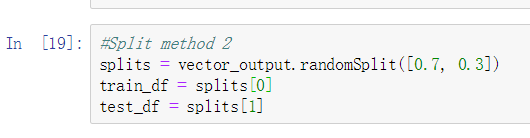
The target setting code can be used for other algorithms like decision tree regression.

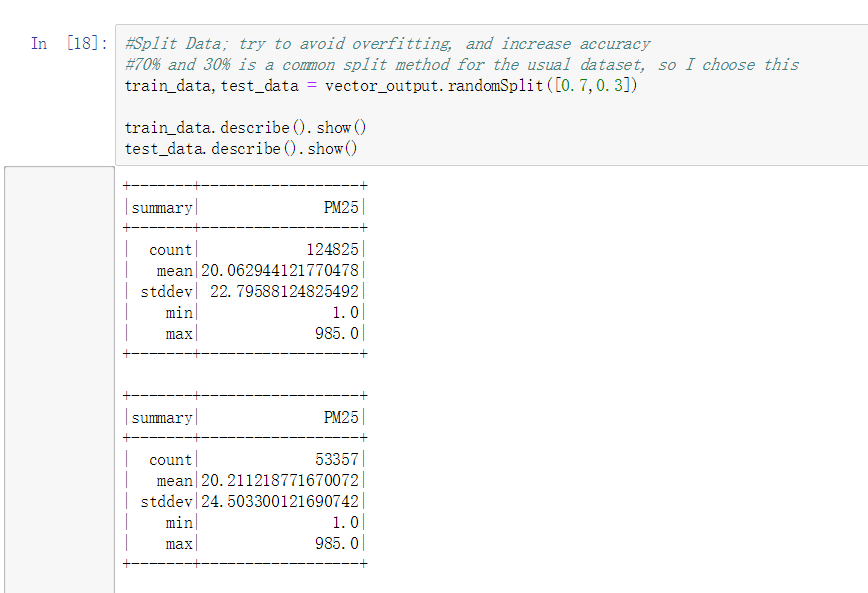


Directly use the previous results

1. **Data Mining**
   1. **Create and justify test designs**

The partition module is used to separate the dataset to different subsets, training, testing and evaluation. By using this, we can avoid the ‘overfit’ phenomenon at a certain degree, that would make the entire design has a more solid result. Before each algorithm is executed, I assign a training subset and a test subset respectively.





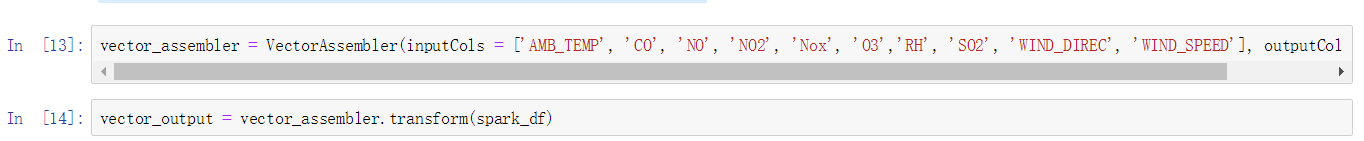
Train and Test Subset

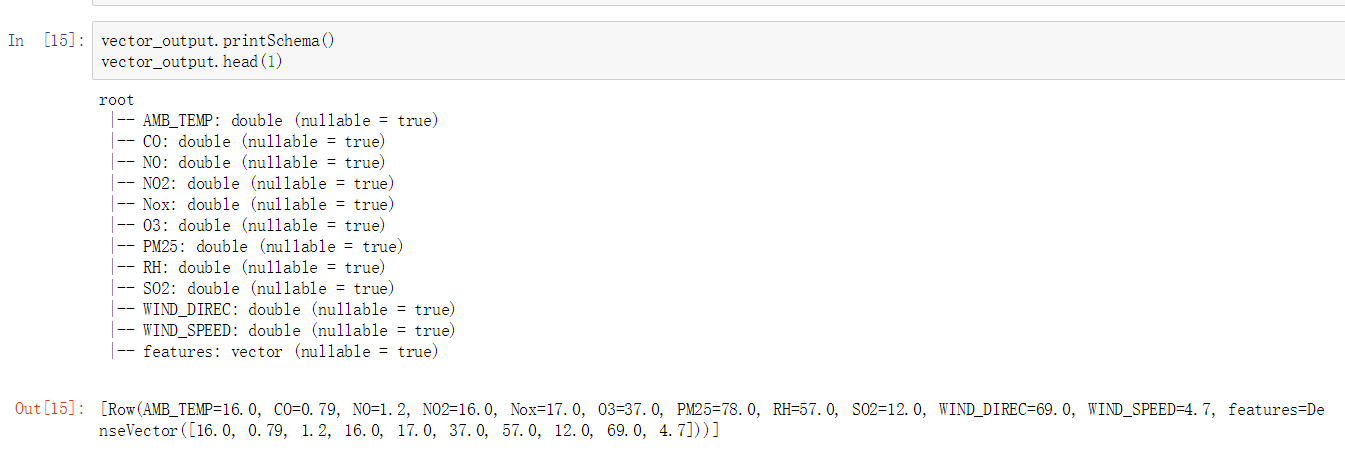
By splitting the dataset, we can avoid the overfitting and test our model, and also improve the model in the future.

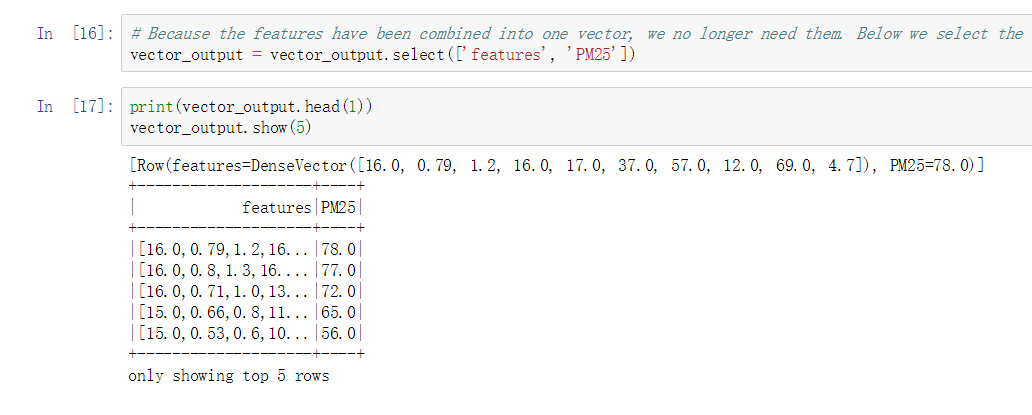
The entire process of this project includes the data import, initially data exploration, data cleaning, model selection, and data splitting.

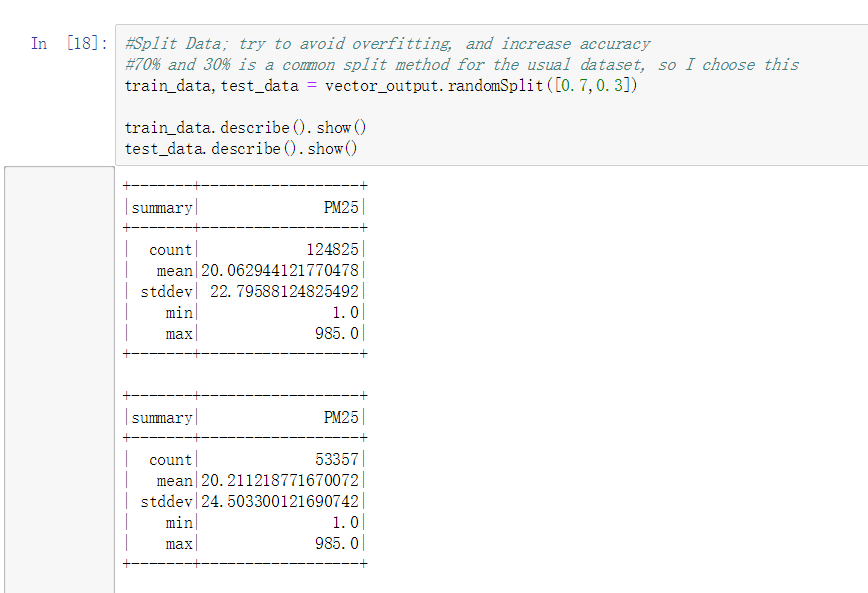
* 1. **Conduct data-mining – Classify, Regression, Clustering**

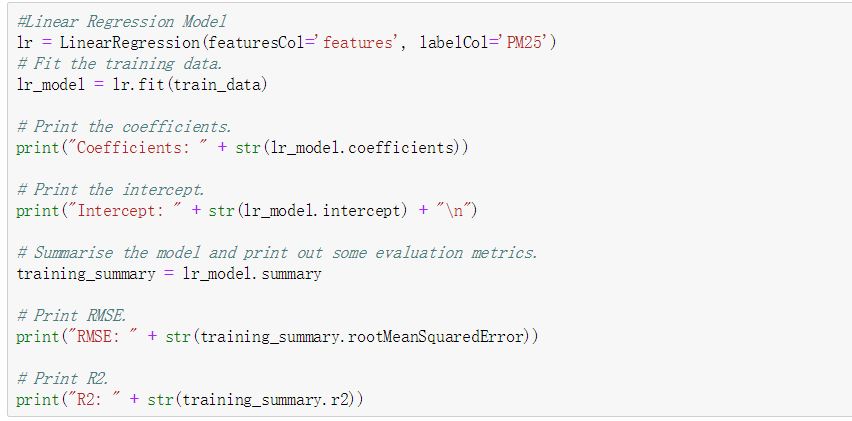
First, I will run the regression, start with the linear regression algorithm.

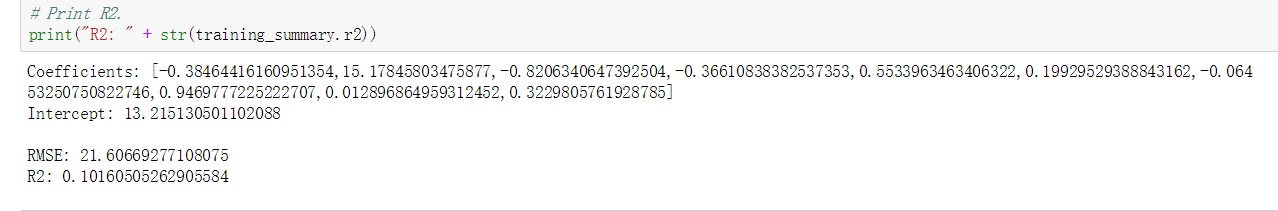


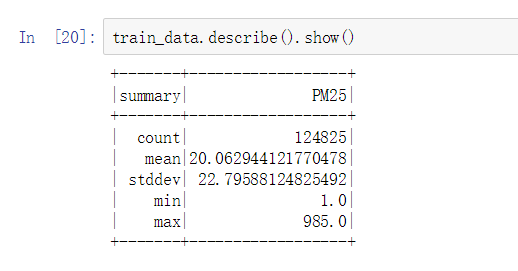






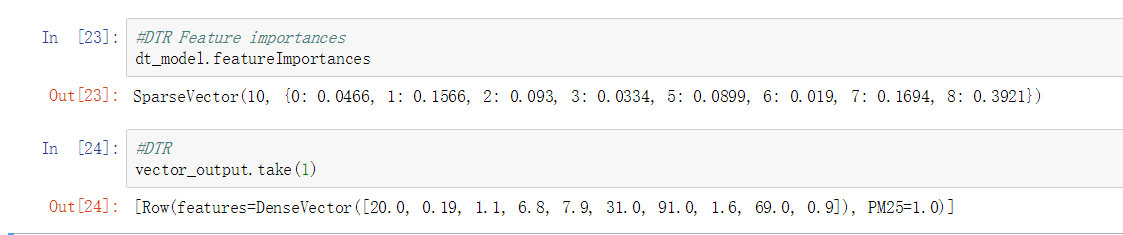
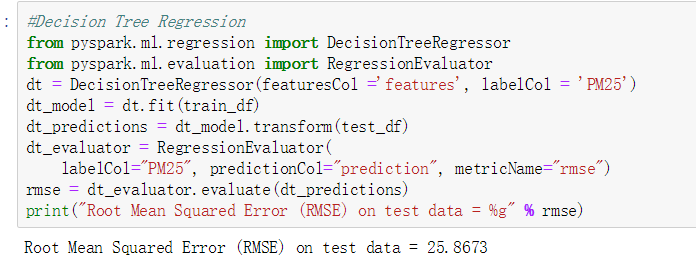




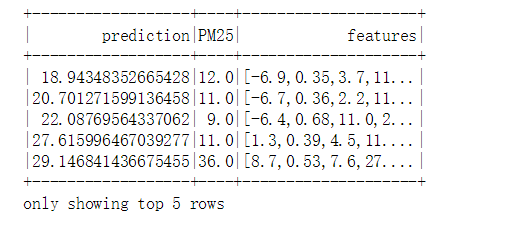
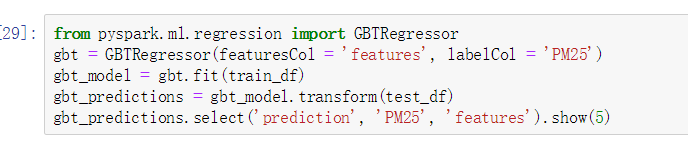


Linear Regression Algorithm

Next is the decision tree regression algorithm.

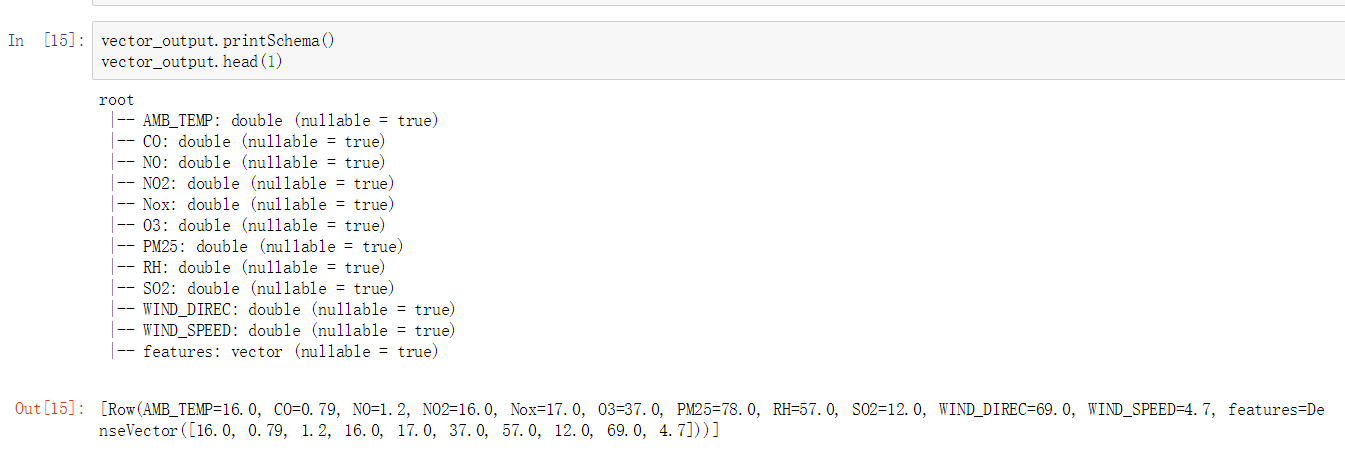


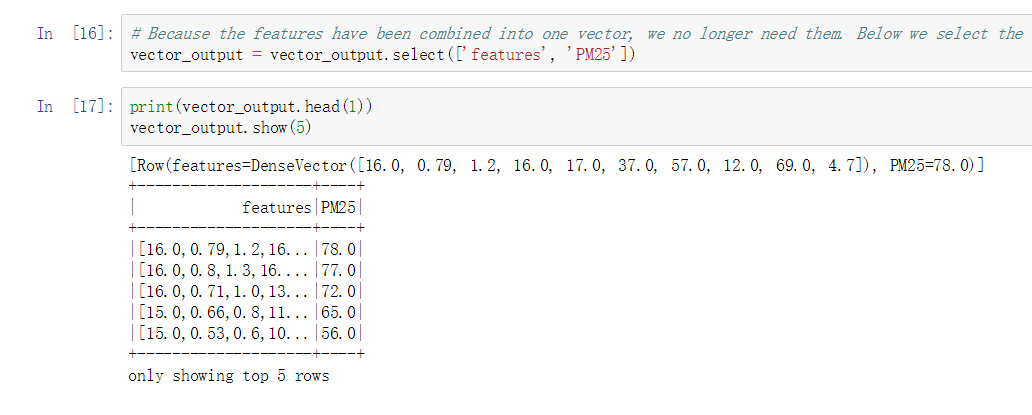
Decision tree

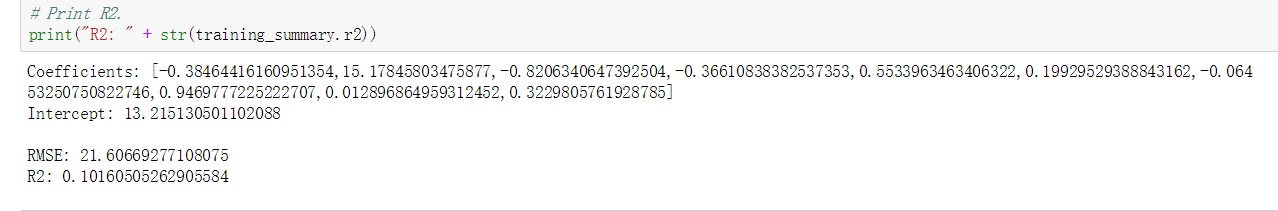
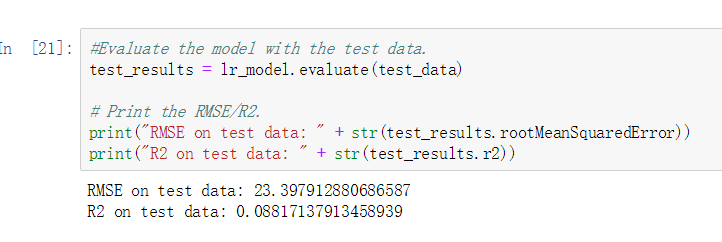


G-Boost Regression

* 1. **Search for patterns**

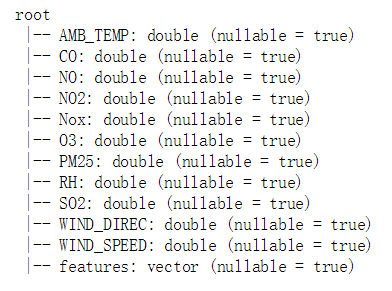




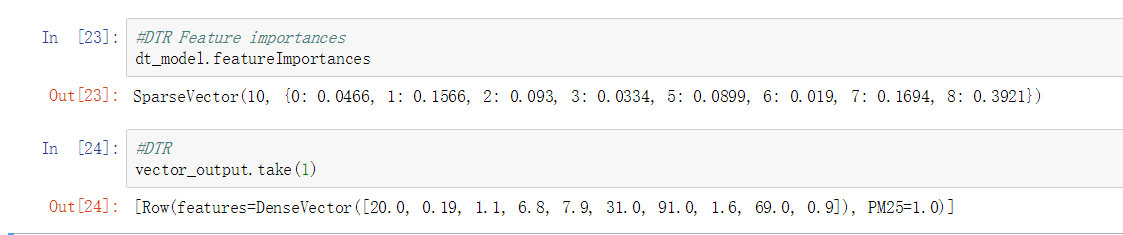


Linear Regression

Here we got a linear regression coefficient result, based on the feature columns, we can get the coefficient values for each attribute.



Attributes



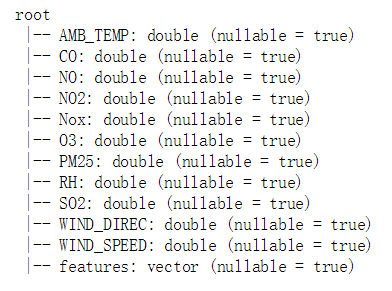
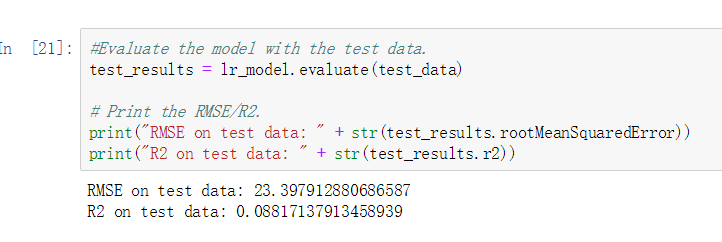
Decision Tree Feature Importance

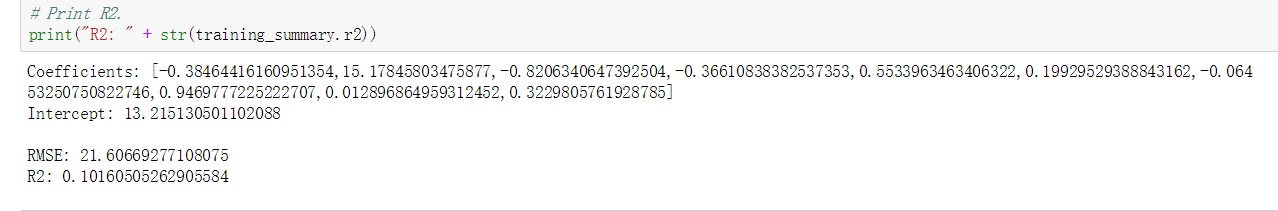
Then we can summarise the result. The temperature (AMB\_TEMP) is negatively correlated with the target pm25, which means if the temperature is high, it is more likely to have a lower value of pm2.5. The chemicals such as NO and NO2 are also negatively correlated with the target.

However, the rmse (root mean square error) is slightly high, while the R2 is a little low, which indicates the model is not very good now.

1. **Interpretation**
   1. **Study and discuss the mined patterns**

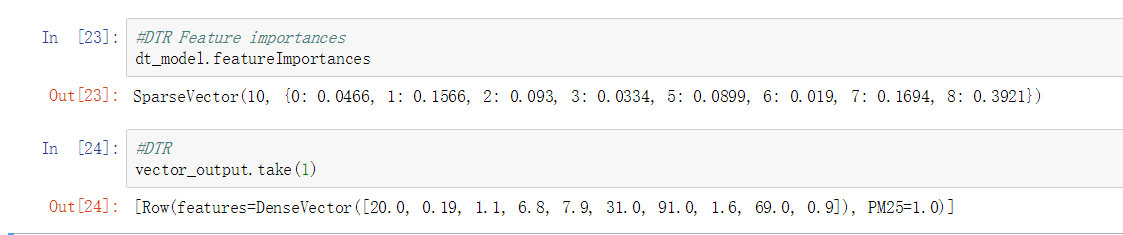
After running all the modules, here we got a lot of results. Now I will connect a ‘Permutation feature importance’ module to discover the different importance of variables.





Based on the coefficient values, we can know which feature is most important for our target pm2.5 value, no matter it is negatively correlated or positive correlated. Then, we can analyse the results to set up a proper environmental policy to relieve air pollution.

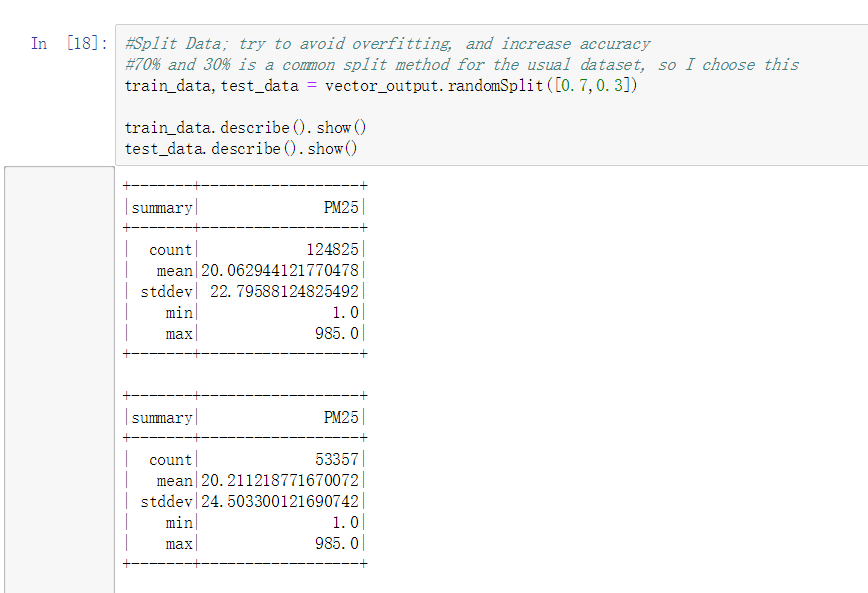
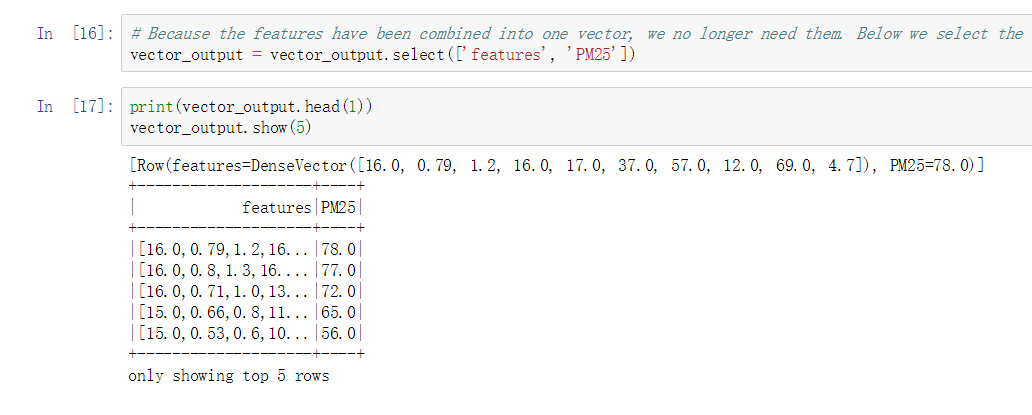
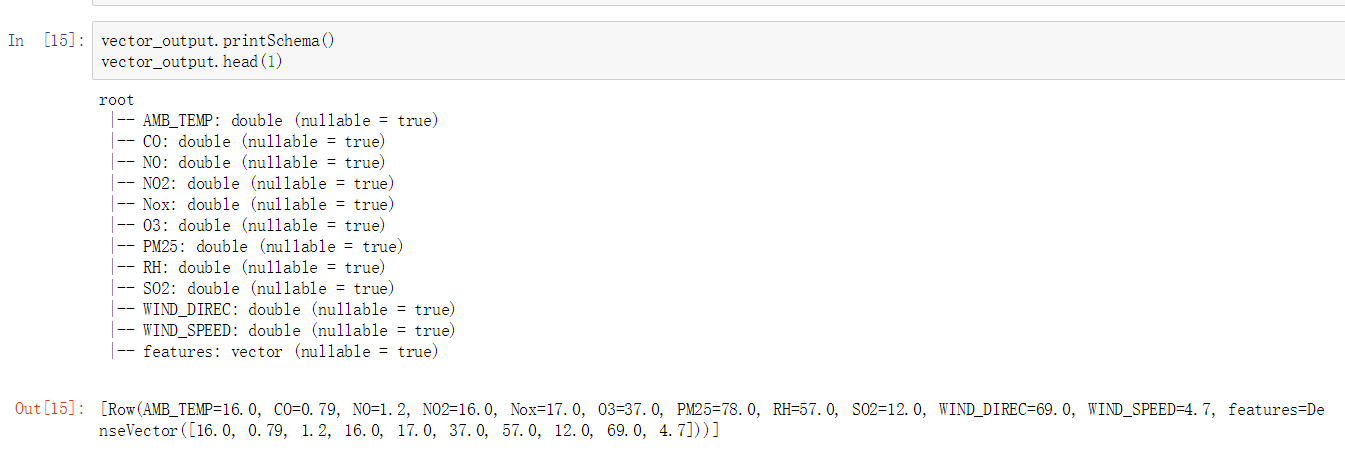
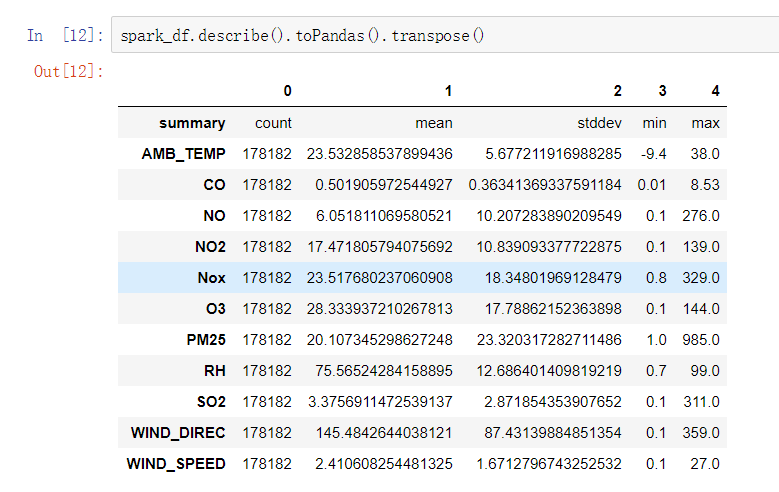
In my opinion, this result is understandable as it is suitable for our common knowledge. Also, if all the pollutants indicators are aggregated together, their result might have a stronger relationship with the PM value. But according to this, we can have a view about which kind of pollutant contributes the most to air pollution.

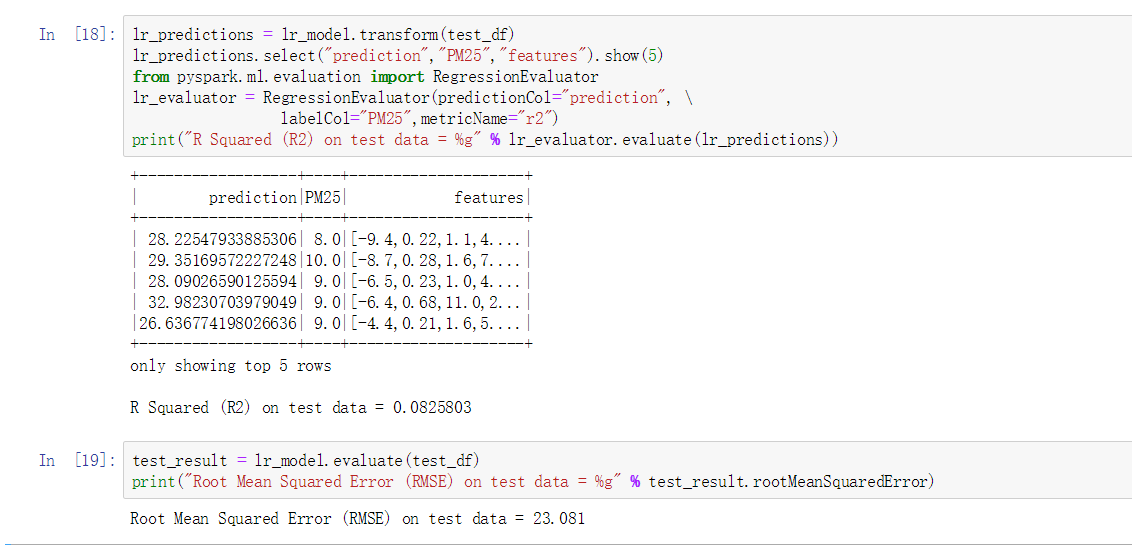


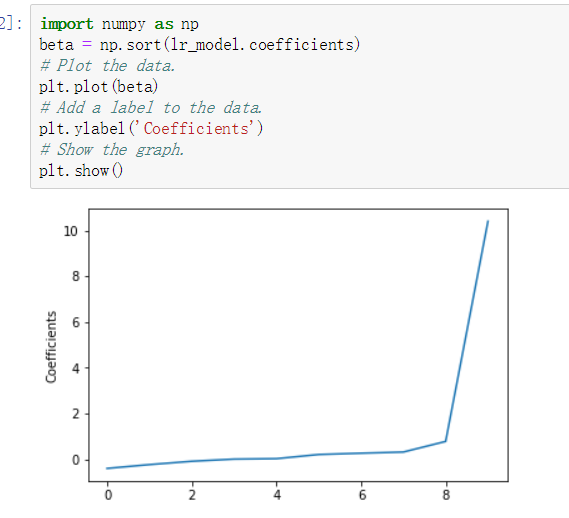
In the result of decision tree regression, we can see the feature importance list, the wind speed is the most important factor, which is easy to understand as the strong wind will blow the air pollution away.

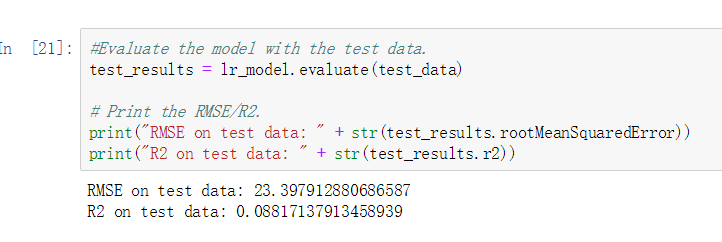
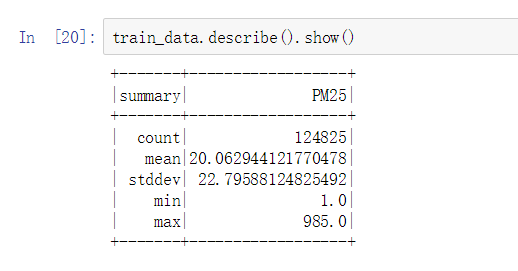
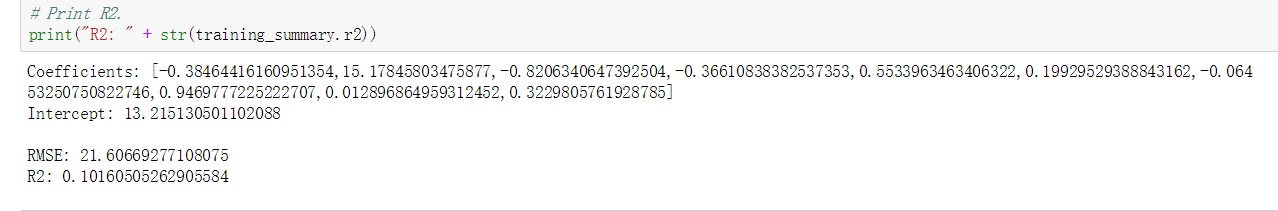
* 1. **Visualize the data, results, models, and patterns**

This section will display the screenshots.

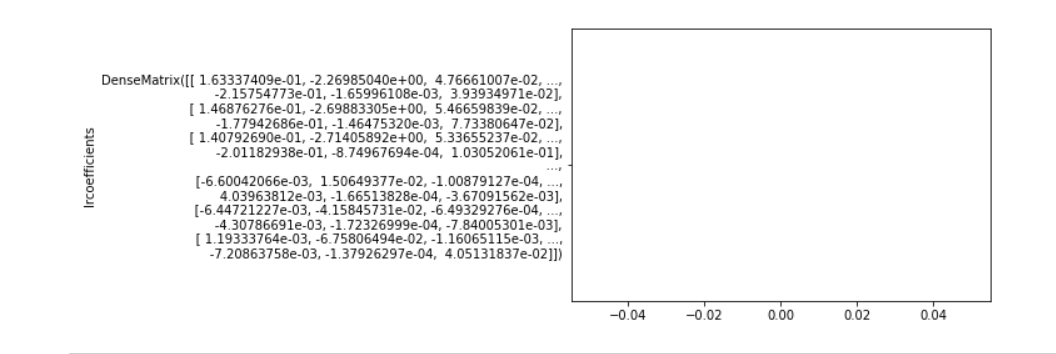
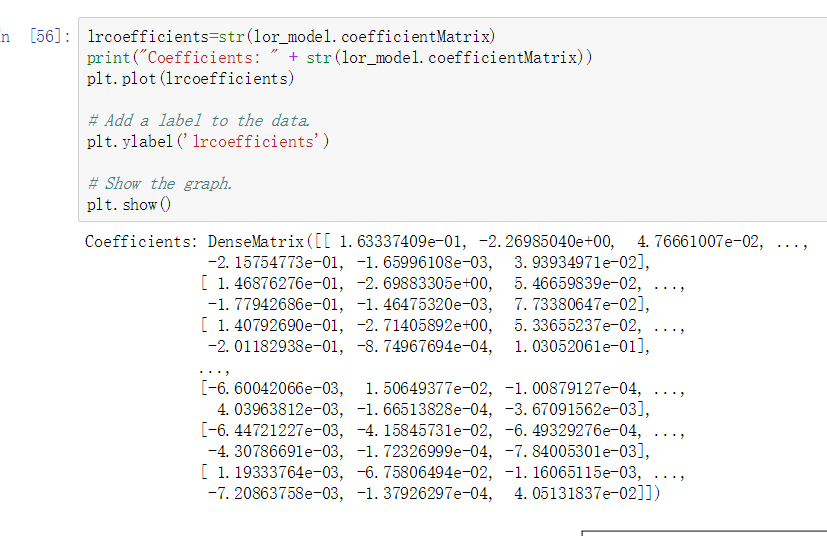
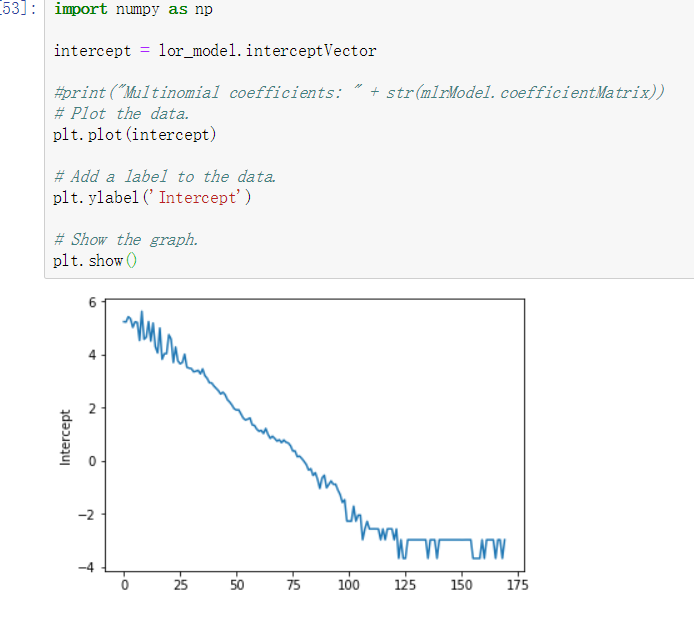
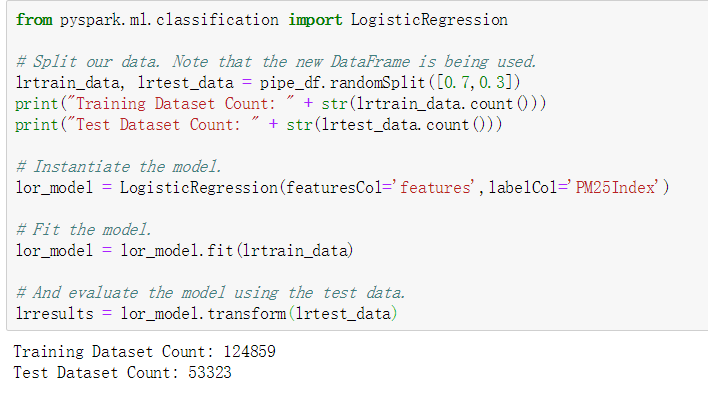




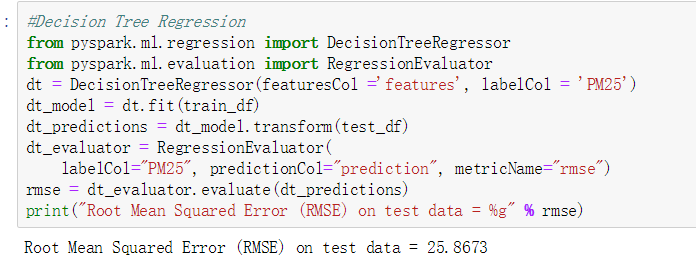




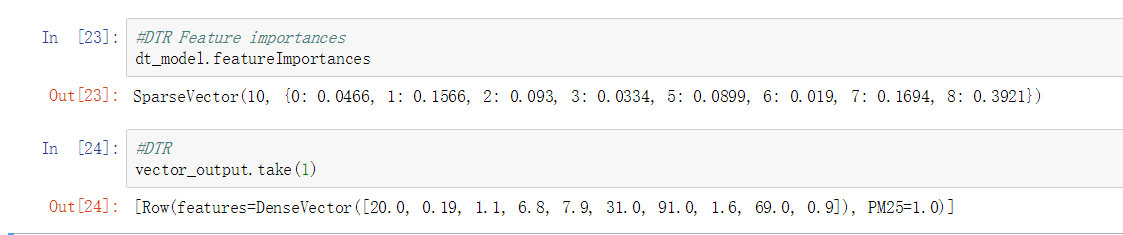
Linear Regression



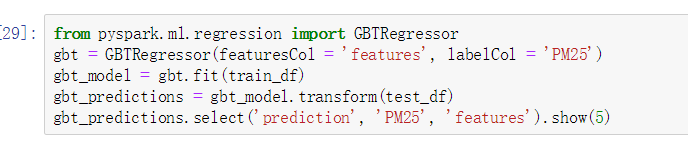
Logistic Regression

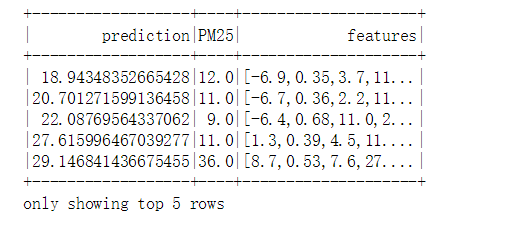
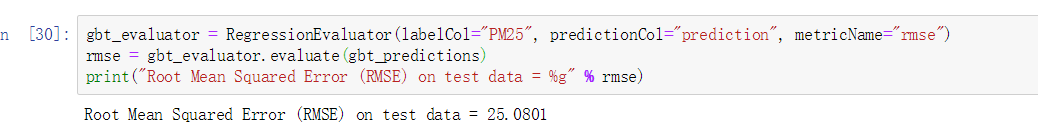


Decision Tree Regression



Decision Tree Feature Importance





G-Boost Regression

* 1. **Interpret the results, models, and patterns**

In this phase, we will interpret these models. From the different models we can infer that, first, if there are more chemical pollutants in the air, especially the CO, SO2, O3, the PM2.5 value will increase to a high level; second, if the temperature is relatively high, the PM2.5 value will decrease.

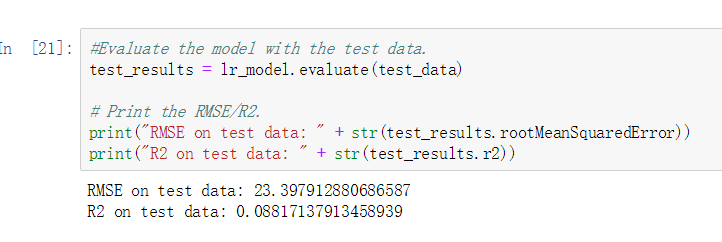
It is reasonable enough for us to understand, and also, it is useful for our environmental control department. For example, we can see that CO has a serious impact on the PM2.5, so the environment department now has a clue – figure out the where does the CO come from, and how it be produced and exhausted into the air. Such model indicates the things that influence our air quality. Also, we can utilise the different ‘contribution’ of air pollution from the various kinds of chemical pollutants to make up a specialised plan, which will solve the air problem based on the importance of predictors one by one.

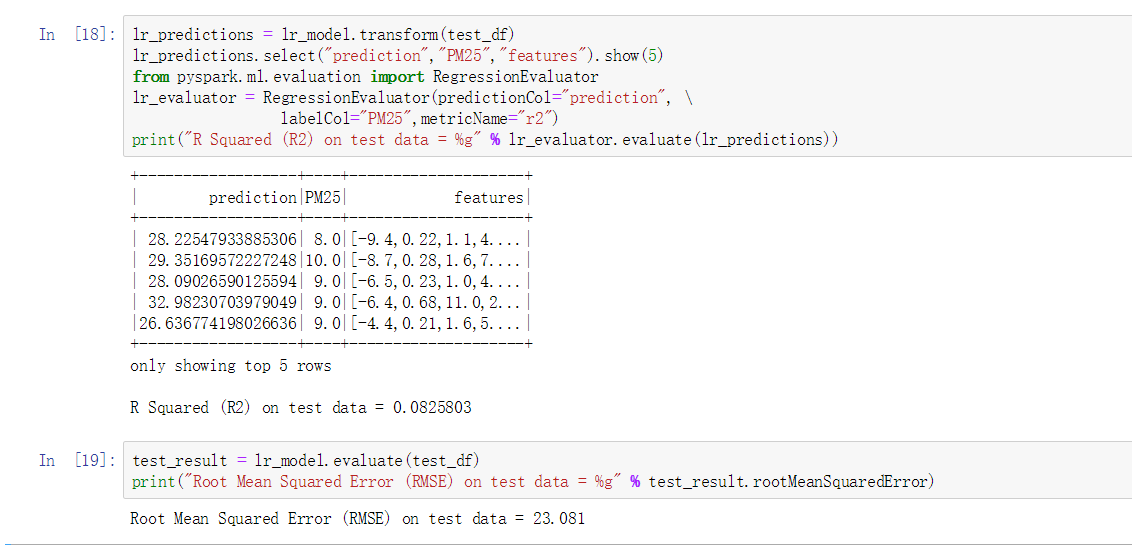
For different algorithms, the importance of different attributes is various. Hence, we may get opposite results by using different algorithms. However, it indicates us that, a data mining process requires us to try several models, and compare the similarities and differences.

* 1. **Assess and evaluate results, models, and patterns**

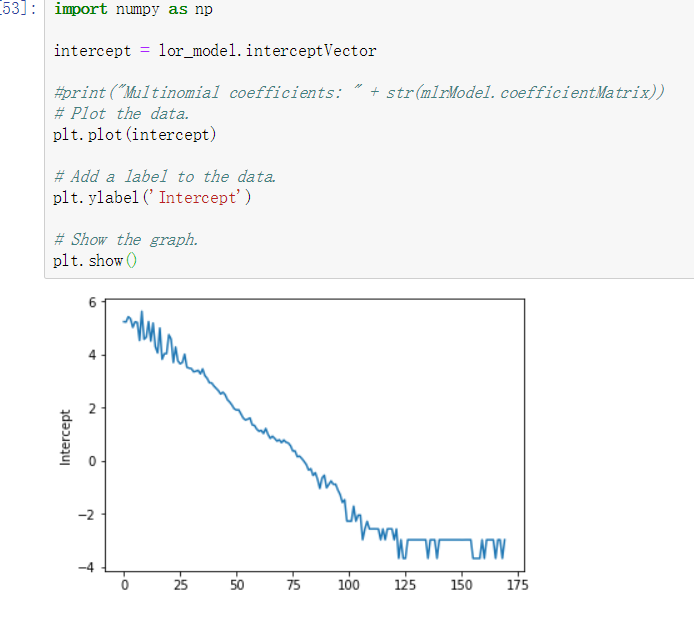
To assess our results, in the spark machine learning library, a method which is named evaluate() can provide such a function for us. Also, for the linear regression, the root mean square error and R2 are useful indicators.

For the regression method, the deviation is usually caused by the noise, outlier, and bias. To evaluate a regression model, R square, root mean square error and mean absolute error all can be used. The precision, recall, and ROC etc. are used to evaluate the classification model. The R square indicates the ability to fit the model, the more it near 1, the better result the model has, while the lower number the rmse has, the better the model will be.

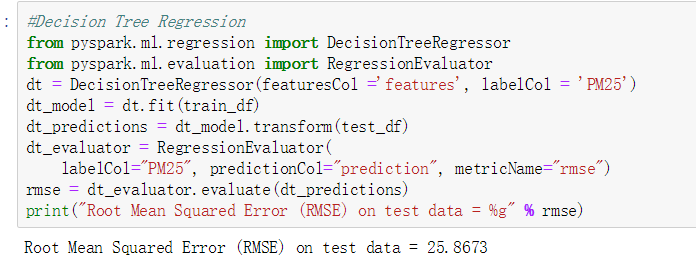




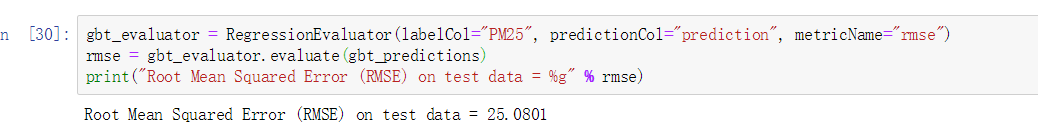
Linear Regression Evaluation



Logistic Regression Intercept



Decision Tree RMSE Evaluate



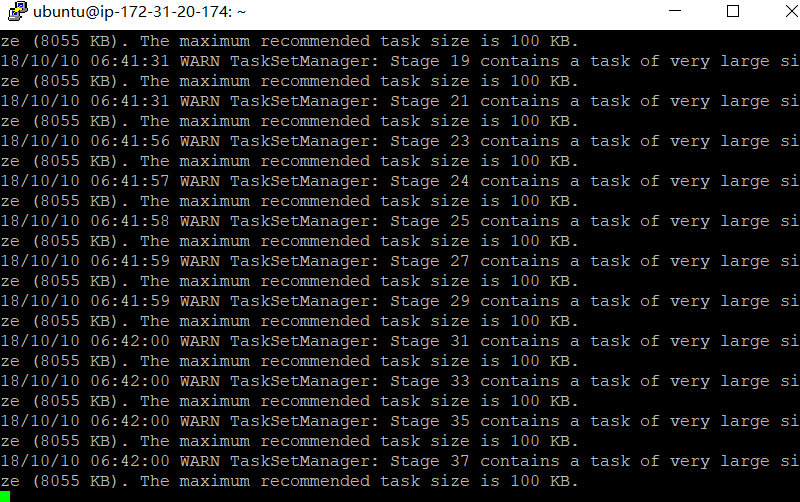
G-Boost Regression Evaluate

From these evaluation results, we can see that all the regression algorithms’ RMSE are slightly high, which indicates that the models are not very accurate. The results indicate us we may need to reconsider the situation of our dataset, re-analyse the dataset, and re-organise the data cleaning process, also, do the data mining process again.

By analysing these models, our environment department may have a better understanding of the air quality and have a clearer view of what are the main pollutants in the air, also, how the external factors such as wind speed, humidity and so on, influence the PM2.5 concentration, and our physical health. Also, by tracking the source of these pollutants, the environmental department may come up some solutions to deal with the ‘pollutant-producers.’ The data mining process reveals the relationship between air quality and chemicals, and it helps people to determine the priority when they want to deal with the pollutants.

* 1. **Iterate prior steps as required**

The data mining process is iterative, as we need to discover and try different algorithms to find the better result. In this case, for example, the first time I run the linear regression, the result is not very good, so I thought if I try to do something with the data, do the result will be better? Hence, if I iteratively try several different models, and adjust the parameters, we may have a better result during the discovery process. Also, in the data cleaning step, it is very often that we need to execute the cleaning operation iteratively as the data cleaning is a very important but also hard part in the data mining process, and then we can get the dataset that we want. Besides, as I do not have a lot of experience with Spark, therefore, when I gradually know more knowledge about Spark, I can do the iteration iteratively, and I can have a better result.



Low Memory Warning (Decision Tree)

Another thing I want to mention is that when I was running the decision tree, the terminal reminded me of the low memory warning (I have cut some records already, but still too large), hence, maybe If I run the code in the distributed environment with enough memory and computing power, the result may be better.

In general, data mining work requires us to do plenty of iterative jobs, and then we can have a robust and accurate model.