Project Github URL: This repo stores the data mining project of INFOSYS 722: Data Mining and Big Data

Table of Contents

1.BUSINESS AND/OR SITUATION UNDERSTANDING 4

1.1 Identify the objectives of the business and/or situation 4
1.2 Assessment of the Situation 5
1.2.1 Resources of the project 5
1.2.2 Requirements, Assumptions, and Constraints 6
1.2.3 Risks and Contingencies 6
1.2.4 Cost/Benefit Analysis 8
1.3 Data mining objectives 9
1.3.1 Specific Objectives 9
1.3.2 Data Mining Success Criteria 10
1.4 Project plan 11

2.DATA UNDERSTANDING 16

2.1 Collect initial data 16
2.1.1 Data Source 16
2.1.2 Data Collection Methods 17
2.1.3 Challenges Encountered During Data Collection 17
2.2 Describe the data 18
2.3 Explore the data 20
2.4 Data Quality Verification 25
2.4.1 Missing Values 26
2.4.2 Outlier Detection 27
2.4.3 Data Duplication Check 27

3.DATA PREPARATION 28

3.1 Data Selection 28
3.2 Data Clean 30
3.2.1 Cleaning Missing Values 30

3.2.2 Cleaning Outliers *30*
3.3 Data Construction *32*
3.4 Data Integration *35*
3.5 Data Reformatting *35*

4.DATA TRANSFORMATION 36

4.1 Reduce the data *36*
4.1.1 Horizontal Dimensionality Reduction *37*
4.2 Project the data *38*

5.DATA-MINING METHOD SELECTION 41

5.1 - Match and discuss DM methods within the context of the DM objectives. *41*
5.2 Select the appropriate data-mining method(s) based on discussion *42*

6.DATA-MINING ALGORITHM(S) SELECTION 43

6.1 Conduct exploratory analysis and Discuss *43*
6.2 select algorithms *44*
6.3 Build/Select appropriate model(s) and choose parameters *45*

7.DATA MINING 49

7.1 Create and justify test designs *49*
7.2 Conduct data mining *50*
7.3 Search for patterns *54*

8.INTERPRETATION 54

8.1 Study and discuss the mined patterns *54*
8.1.1 Understanding the Patterns *55*
8.1.2 Discussion of Mined Patterns *55*
8.2 Visualize the data, results, models, and patterns *57*
8.3 Interpret the results, models, and patterns *60*
8.3.1 Results *60*
8.3.2 Models *60*
8.3.3 Patterns *60*
8.4 Assess and evaluate results, models, and patterns *61*
8.5 Iterate prior steps as required *61*

Data Mining and Prediction Modeling for Air pollution

RuiLi-562077095

INFOSYS 722: Data Mining and Big Data, University of Auckland

Abstract:

This research project primarily focuses on in-depth exploration, data mining, and accurate prediction of the significant impact that meteorological factors have on air quality. The study delves into the intricate relationship between the two, aiming to gain a comprehensive understanding of how various meteorological elements interact with air pollutants and ultimately affect the quality of the air we breathe. By conducting extensive research and employing advanced analytical techniques, this project endeavors to unravel the underlying mechanisms and patterns that govern this complex interplay. The findings of this research will not only contribute to our theoretical knowledge in the field of environmental science but also have practical implications for effective air quality management, policy formulation, and the development of targeted strategies to mitigate the adverse effects of poor air quality on human health and the environment. Additionally, it will provide valuable insights for stakeholders, including environmental agencies, researchers, and policymakers, enabling them to make informed decisions and take appropriate actions to safeguard air quality and ensure a healthier living environment for all. This comprehensive study represents a significant step forward in our efforts to better understand and address the challenges associated with meteorological influences on air quality, ultimately striving for improved air quality and a sustainable future.

1.Business and/or Situation understanding

1.1 Identify the objectives of the business and/or situation

air pollution is a significant environmental risk and is closely related to people's daily lives. The health hazards caused by air pollution have become a global public health issue, not just in Taiwan (Liu Yiting, February 22, 2018).

Taiwan's rapid economic development has been accompanied by a significant increase in air pollution.

Air pollution has long been a pressing issue in Taiwan, and while recent efforts have shown some improvement, it remains a significant environmental and public health concern. According to the Environmental Protection Administration (EPA) of Taiwan, the annual average PM2.5 concentration in 2022 was 21 micrograms per cubic meter, exceeding the World Health Organization's (WHO) recommended guideline of 5 micrograms per cubic meter [1].

Air pollution in Taiwan can have various negative impacts on human health and daily life. Some possible effects include:

1. Respiratory problems: Exposure to polluted air can trigger or exacerbate respiratory issues, including coughing, shortness of breath, and asthma attacks [2].
2. Increased risk of heart disease: Poor air quality has been linked to the development of cardiovascular diseases such as hypertension, diabetes, stroke, and heart attacks [3].
3. Weakening of the immune system: Prolonged exposure to air pollution can suppress the immune system, making individuals more vulnerable to infections and diseases [4].
4. Negative impact on tourism: Deteriorating air quality can discourage tourists from visiting Taiwan, leading to a decline in the tourism industry, impacting the economy [5].
5. Higher healthcare costs: The treatment of air pollution-related illnesses results in increased healthcare expenses for individuals and society as a whole, straining the healthcare system [6].

Urban air quality in Taiwan is influenced by a complex interplay of factors, including weather conditions (wind, rain, temperature) [7], geographical features (terrain, elevation) [8], and human activities (industrialization, population density) [9]. The island's diverse landscape presents challenges in collecting accurate data

across different regions, potentially hindering data preprocessing and model generation efficiency [10]. Moreover, the varying degrees of urbanization and industrialization across Taiwan lead to regional disparities in air quality, further complicating effective data processing and the generation of reliable air quality reports [11].

In conclusion, a project will be carried out with the following aims:

1. Describe and visualize meteorological data.
2. The relation between meteorological elements and air quality.
3. The impact of different meteorological conditions on air quality.

The project will be regarded as complete if the following standards are fulfilled:

1.Complete Data Description and Visualization:
Successfully describe and visualize meteorological data.

2.Complete Factor Category Analysis
If the impact of meteorological phenomena on air quality,find the key factors,Otherwise,meteorological factors are not the determining factors of air quality.

1.2 Assessment of the Situation

Severe air pollution can damage people's respiratory systems, causing various respiratory diseases, such as asthma and emphysema. It also increases the risk of cardiovascular diseases such as heart disease and stroke. Long-term exposure to polluted air may even lead to the occurrence of cancer. In addition, air pollution can affect people's nervous systems, resulting in problems such as inattention and memory decline. It can be said that air pollution has become a major hidden killer endangering human health, and we must attach great importance to it and take effective measures to deal with it.

1.2.1 Resources of the project

Data source:

Meteorological data and Air quality data from

1. Kaggle(www.kaggle.com) -The data is too old and can only be used for reference.
2. World Air Quality Historical Database(https://aqicn.org/historical/)

3. the Taiwan Environmental Protection Administration Executive Yuan(https://data.moenv.gov.tw)

Hardware source: Macbook
Software source: ANACONDA.NAVIGATOR, Spyder, Jupyter Notebook,Python along with other essential development and collaborative tools.

1.2.2 Requirements, Assumptions, and Constraints

1. Requirement:
Obtain relevant meteorological data (such as precipitation/snowfall amount, temperature, wind speed, etc.) and air pollution data (such as pollutant concentration). Combined with specific geographical environments, seasons, and other factors, further explore the influence mechanism of meteorological factors on air pollution.

2.Assumption:
It is assumed in this analysis that the dataset is accurate and recent, and that the relevant skills and knowledge are consistently being refreshed.

3. Constraints
Possible limitations could encompass data availability, budget constraints, time limitations, as well as regulatory and privacy restrictions.

1.2.3 Risks and Contingencies

Risk	Describe	Solutions	Impact
Data Quality Issues	Some data are missing or confusing	1. Verify data in multiple ways 2. Use data cleansing tools to correct erroneous data.	High

Goal confusion	Forget about goals and get bogged down in redundant data analysis	1. Set clear schedules and goals. 2. Increase self-monitoring and motivation mechanisms. 3. Conduct daily reviews and mark progress charts.	Low
Technical Difficulties	Being unfamiliar with technology or language and unable to use it proficiently	1. Conduct technical assessments and tool selection early in the project. 2. Seek online resources and community support for guidance. 6 3. Consider adopting more mature and widely	Moderate

		supported tools.	
Project Delay	Project cannot be completed on time, Maybe it's because it takes too long to solve the problem, or technical bottlenecks cannot be solved, etc.	1. Set and adhere to a clear timeline. 2. If the problem cannot be solved within the time limit, submit it to the technical community	Moderate

Table 1 Risks and Contingencies

1.2.4 Cost/Benefit Analysis

Cost Analysis

1. Time Cost:

The time cost includes the analysis before project establishment, the selection of materials, and the formulation of plans, all of which require time investment.

The use of tools and the learning of languages also require time.

2.Money Cost

The consultation and download of some documents may cost money.

Benefit Analysis

Enhance one's ability in project decision-making and analysis, face various difficulties encountered in the project, enhance the determination to solve problems, and enhance the ability of project planning.

The mastery of new tools and new programming languages is also a kind of personal improvement.

1.3 Data mining objectives

Identify the relationship between meteorological conditions and air quality through predictive and classificatory analysis.

1.3.1 Specific Objectives

Usage stage:

Through statistical data, it was found that the weather is constantly changing throughout the four seasons, and air pollution is also different

By analyzing and visualizing data, we can identify the factors that have the greatest impact on the air quality index.

Discovery and Analysis Phase:

If uneven distribution is found, data analysis will be based on factors such as temperature, humidity, weather, etc.: This analysis

Enable us to gain a deeper understanding of the key factors affecting air pollution.

Future direction stage:

Based on the analysis results, optimize the factors affecting air pollution:

It will enable more people to understand and improve their living habits, thereby avoiding the impact of air pollution on their bodies.

These three phases establish a clear data mining process and ensure project continuity, consistency, and a close alignment with business objectives.

1.3.2 Data Mining Success Criteria

1. Methods for Model Assessment

- Accuracy: The model is expected to precisely mirror the patterns and relationships existing within the data.
- Interpretability: The model needs to be readily comprehensible to facilitate the decision-making procedure.
- Applicability: The model ought to be practicable for actual use in business settings.

2. Benchmarks for Evaluating Success

- Performance Benchmark: Evaluate the model's performance against existing solutions or industry standards, ensuring it meets or surpasses established metrics.
- Business Impact Benchmark: The project should demonstrate quantifiable business value or impact, contributing positively to key performance indicators or organizational goals.

3.Subjective Measurements and the Arbiter of Success

- User Satisfaction: Assess if the model effectively meets the needs and expectations of both internal and external customers, ensuring their satisfaction with its performance and usability.

- Alignment with Objectives: Evaluate whether the model aligns with the specific goals set at the onset of the project, ensuring it fulfills the intended purposes and delivers the expected outcomes.

- Arbiters of Success: Clearly define the assessment parties and criteria that determine project success, including project managers and relevant business stakeholders, to establish clear benchmarks for evaluating performance and impact.

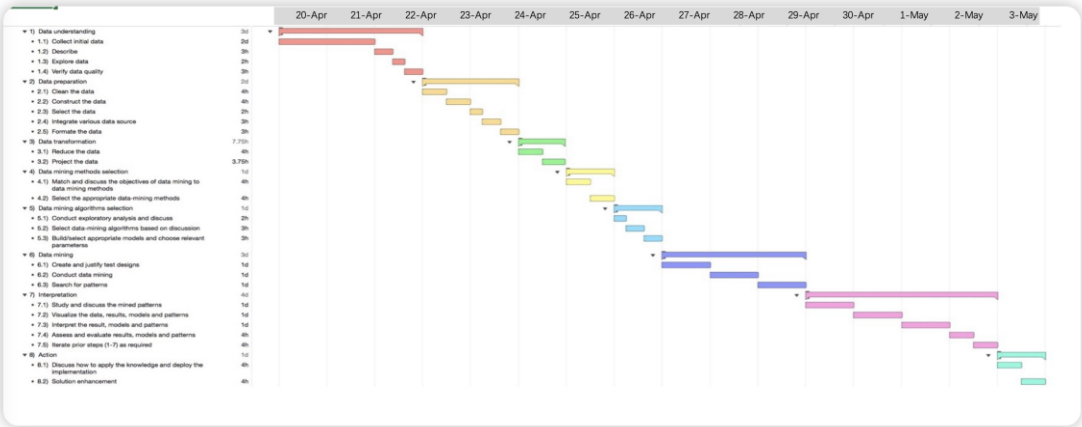
4. Successful Deployment of Model Results a Part of Data Mining Success:

Air is an essential element for human beings, which is related to human survival and health. However, with the rapid development of industrialization and urbanization, the problem of air pollution has become increasingly serious, which poses a great threat to human health.

If the model is successful, it can help guide people's lifestyles and reduce the impact of pollution on people in some weather conditions

1.4 Project plan

This project of analyzing meteorological effect on air quality may take around 2 weeks, followed by the visualized data mining roadmap:



1.4.1 Project Phases

Phase	**Time (allocated in %)**	**Risks**
1. Business and/or Situation understanding (10%)	10%	1. Disalignment from business objectives, indistinct project scope.
2. Data understanding (10%)	10%	2. Incoherent data, poor data quality.
3. Data preparation (15%)	15%	3. Insufficient data preparation, integration difficulties.
4. Data transformation (5%)	5%	4. Loss of significant information, projection mistakes.
5. Data-mining method(s) selection (10%)	10%	5. Incorrect method choosing, disalignment with the objectives.
6. Data-mining algorithm(s) selection (15%)	15%	6. Unsuitable algorithm selection, parameter adjusting.

7. Data Mining (15%)	15%	7. Model execution breakdown, pattern recognition failure.
8. Interpretation (20%)	20%	8. Misconception of the results, assessment errors.

Phase	**Time (allocated in %)**	**Risks**
Iteration 1 - Proposal (Steps 1)	10%	Disalignment from business objectives, indistinct project scope.
Iteration 2 ISAS (Steps 2 – 5)	25%	- Incoherent data, poor data quality. (Step 2) - Insufficient data preparation, integration difficulties. (Step 3) - Loss of significant information, projection mistakes. (Step 4) - Incorrect method choosing, disalignment with the objectives. (Step 5)
Iteration 3 OSAS (Steps 6 – 7)	25%	Unsuitable algorithm selection, parameter adjusting. (Step 6) Model execution breakdown, pattern recognition failure (Step 7)
Iteration 4 BDAS (Steps	20%	Misconception of the results,

8)		assessment errors. (Step 8)
Research Paper (Details of Steps 1 – 9)	20%	Insufficient detail, lack of actionable insights, failure to meet the deadline

1.4.2 Project Daily Plan

Phase	**Time**	**Stage**	**Start**	**End**
Business understanding	7 days	1.1 Define the Objects of the Situation	28/04/2024	02/04/2024
		1.2 Situation Assessment	03/04/2024	04/04/2024
		1.3 Data Mining Objectives	04/04/2024	04/04/04
		1.4 Project Plan	04/04/2024	04/04/2024
Data understanding	5 days	2.1 Collecting Initial Data	05/04/2024	06/04/2024
		2.2 Describe Data	07/04/2024	07/04/2024
		2.3 Data Exploration	07/04/2024	07/04/2024
		2.4 Data Quality Verification	07/04/2024	09/04/2024

Data preparation	9 days	3.1 Data Selection	08/04/2024	9/04/2024
		3.2 Cleaning the data	10/04/2024	11/04/2024

		3.3 Construct New Features	11/04/2024	11/04/2024
		3.4 Data Integration	12/04/2024	13/04/2024
		3.5 Formatting Data	14/04/2024	15/04/2024
Data Transformation	4 days	4.1 Data Reduction	16/04/2024	17/04/2024
		4.2 Data Projection	18/04/2024	19/04/2024
Data Mining Method Selection	3 days	5.1 Discuss the Objectives of Data Mining	20/04/2024	21/04/2024
		5.2 Select the Appropriate Data Mining method	22/04/2024	22/04/2024
Data Mining Algorithm Selection	5 days	6.1 Exploratory Analysis and Discuss	23/04/2024	24/04/2024
		6.2 Select Data Mining Algorithms	25/04/2024	25/04/2024
		6.3 Build Appropriate Models with Algorithms	26/04/2024	27/04/2024
Data Transformation	4 days	4.1 Data Reduction	16/04/2024	17/04/2024
		4.2 Data Projection	18/04/2024	19/04/2024
Data Mining Method Selection	3 days	5.1 Discuss the Objectives of Data Mining	20/04/2024	21/04/2024
		5.2 Select the Appropriate Data Mining method	22/04/2024	22/04/2024
Data Mining Algorithm Selection	5 days	6.1 Exploratory Analysis and Discuss	23/04/2024	24/04/2024

		6.2 Select Data Mining Algorithms	25/04/2024	25/04/2024
		6.3 Build Appropriate Models with Algorithms	26/04/2024	27/04/2024
Data Mining	8 days	7.1 Create and Justify Test Designs	28/04/2024	31/04/2024
		7.2 Conduct data mining	01/05/2024	02/05/2024
		7.3 Search for patterns	03/05/2024	04/05/2024
Interpretation	8 days	8.1 Study and Discuss the Mined Patterns	05/05/2024	06/05/2024
		8.2 Visualize the Data, Results,	07/05/2024	08/05/2024
		8.3 Interpret the Results, Models, and Patterns	09/05/2024	10/05/2024
		8.4 Assess and Evaluate Results, Models, and Patterns	11/05/2024	12/05/2024
		8.5 Iterate Prior Steps	05/05/2024	12/05/2024
Action	14 days	9.1 Discuss how to apply the knowledge and deploy the implementation	13/05/2024	15/05/2024
		9.2 Discuss how to monitor the implementation	15/05/2024	20/05/2024
		9.3 Discuss how to maintain the implementation	20/05/2024	24/05/2024

		9.4 How to enhance the solution in the future?	25/05/2024	26/05/2024
Action	14 days	9.1 Discuss how to apply the knowledge and deploy the implementation	13/05/2024	15/05/2024

2.Data understanding

2.1 Collect initial data

2.1.1 Data Source

Air quality data is primarily collected and aggregated from Kaggle:
This data science community hosts several Taiwanese air quality datasets, potentially useful for supplementary analysis or specific research purposes.
Link:
https://www.kaggle.com/datasets/lionvp/taipeis-air-quality/data

Here is a polished version: Since the data source is collected from various sources, their formats need to be standardized and then combined together for subsequent data analysis. The raw data set is named “original-data.csv" in the folder.

2.1.2 Data Collection Methods

1. The CWB (Central Weather Bureau) is Taiwan's meteorological authority, tasked with collecting, disseminating, and managing climate data for the region. Meteorological data collection involved a variety of methods:

   Meteorological Station Observations:

   Meteorological stations serve as fundamental units for ground-based weather observation. They are crucial for conducting weather observations and research, issuing weather forecasts, warnings, and services. Taiwan boasts 61 meteorological stations, comprising 52 ground-based stations and 9 upper-air observation stations. These stations utilize various instruments and equipment to observe atmospheric elements, including temperature, pressure, humidity, wind speed, wind direction, precipitation, sunshine duration, evaporation, and others.

   Radar Observation:

   Weather radars are tools that utilize radar waves to detect atmospheric phenomena like clouds, precipitation, and wind. Taiwan has 16 weather radars, covering the entire island. These radars enable real-time monitoring of weather conditions, providing crucial information for weather forecasting, warnings, and disaster mitigation.

   Satellite Remote Sensing:

   Satellite remote sensing is a technique that employs artificial satellites equipped with remote sensing instruments to gather information about Earth's surface. Taiwan can access data on the atmosphere, oceans, and land from multiple meteorological and environmental satellites.

2. Air pollution data in Taiwan can be accessed through Taiwan Environmental Protection Agency (EPA) Air Quality Monitoring Network:

   The Taiwan EPA operates an air quality monitoring network with numerous air quality monitoring stations established across the island, providing real-time monitoring of air quality conditions. The monitoring data encompasses the concentration of pollutants like fine particulate matter (PM2.5), fine particulate matter (PM10), ozone (O3), nitrogen dioxide (NO2), carbon monoxide (CO), and sulfur dioxide (SO2).

2.1.3 Challenges Encountered During Data Collection

Although the data collection went smoothly, we also encountered some problems:

- Data Consistency Issues: Inconsistency in data formats and standards arose due to the data originating from multiple sources.
- Missing and outlier values were identified in the initial data inspection.

- Unnecessary Redundant Data: There is some non-utilitarian data occupying a large amount of storage in the file, and some data are unusable due to the deficiency of information or fields.

2.2 Describe the data

The scale of our data set has 26 columns and approximately 1000 records, and it stored in a CSV file, so I can upload this file to the 'Dataset' module on the spyder.

Here are data fields within the dataset:

TM3	Maximum temperature (°C)
Tm4	Minimum temperature (°C)
SLP	Atmospheric pressure at sea level (hPa)
H	Average relative humidity (%)
PP	Total rainfall and / or snowmelt (mm)
VV	Average visibility (Km)
V	Average wind speed (Km/h)
VM	Maximum sustained wind speed (Km/h)
VG	Maximum speed of wind (Km/h)
RA	Indicate whether there was rain or drizzle (In the monthly average, the total days it rained)
SN	Indicate if it snowed (In the monthly average, the total days it snowed)

TS	Indicates whether there was storm (In the monthly average, Total days with thunderstorm)
FG	Indicates whether there was fog (In the monthly average, Total days with fog)
AQI Quality	
AQI	
AQI Ranking	
PM2.5	
PM10	
So2	
No2	
Co	
O3	

An air quality index (AQI) is a numerical value that represents the degree of pollution of the current air or the expected level of pollution in the future. With the increase of the AQI, an increasingly larger proportion of the population is likely to encounter increasingly severe adverse impacts on health. Different countries possess their own air quality indices, which correspond to different national air quality standards.

Six atmospheric pollutants that AQI has considered are also included in the data ,namely sulfur dioxide (SO2), nitrogen dioxide (NO2), suspended particulates smaller than 10 μm in aerodynamic diameter (PM10), suspended particulates smaller than 2.5 μm in aerodynamic diameter (PM2.5), carbon monoxide (CO), and ozone (O3) measured at the monitoring stations throughout each city. In this way, the AQI level plays an important role as a comprehensive indicator that reveals potential air pollution problem.

2.3 Explore the data

In order to simply explore data within a spark environment, we have to set up a spark instance and establish a spark data frame based on that instance.

```
#initialization

import findspark

findspark.init()

import pyspark

import pandas as pd

import matplotlib as mpl

import matplotlib.pyplot as plt

from pyspark.sql import SparkSession

spark = SparkSession.builder.appName('rli795').getOrCreate()

#Read csv file

df = spark.read.csv('Original-data.csv',header=True,inferSchema=True)
```

Original meteorological data is shown below:

Original-data

Date	Day	T	TM	Tm	SLP	H	PP	VV	V	VM	VG	RA	SN	TS	FG	AQI Quality	AQI	AQI Ranking	PM2.5	PM10	So2	No2	Co	O3
1/01/21	1	-3.2	2.5	-10	-	25	0	6.4	9.1	18	-					Good	61	84	41	66	23	49	1.09	29
2/01/21	2	-1.5	5	-8	-	30	0	6	8	14.4	-					Good	69	172	48	78	30	58	1.28	23
3/01/21	3	-3.1	3.6	-8	-	45	0	5	5.4	14.4	-					Moderately Polluted	194	322	148	193	66	100	3.24	8
4/01/21	4	2.2	8.6	-7	-	37	0	4.3	5.6	10.7	-					Heavily Polluted	217	286	173	224	74	119	3.67	12
5/01/21	5	1.7	11	-6	-	22	0	4.2	20	39.6	50.4					Moderately Polluted	154	216	106	156	57	70	2.89	30
6/01/21	6	-1.1	6	-8	-	23	0	9.5	10.7	25.2	28.7					Good	53	90	34	51	22	40	0.8	37
7/01/21	7	-2.6	3.3	-11	-	39	0	6.1	4.3	10.7	-					Good	85	215	61	89	38	64	1.77	14
8/01/21	8	-2.5	5	-8	-	50	0	6.3	7.8	18	-					Moderately Polluted	162	315	124	151	49	91	2.98	7
9/01/21	9	-0.9	7.5	-10	-	32	0	8	9.8	32.4	43.2					Good	99	170	73	92	32	61	1.57	28
10/01/21	10	4.8	13	-8	-	22	0	6.1	15	35.9	53.9					Moderately Polluted	191	294	146	174	54	94	2.8	17
11/01/21	11	0.6	6.5	-7	-	36	0	7.6	6.5	10.7	-					Good	56	66	37	55	18	46	0.89	33
12/01/21	12	-2.3	4.2	-7	-	55	0	5.5	6.3	14.4	-					Moderately Polluted	154	313	116	145	42	72	2.18	12
13/01/21	13	-2.6	3.1	-7	-	69	0	4	5.6	10.7	-					Heavily Polluted	203	348	155	183	54	91	2.81	7

Delete unused data columns such as the "AQI Ranking" column, which displays Taipei's air quality ranking compared to other cities in Tanwan.

The column of 'AQI quality', which is actually derived from the AQI, can be utilized as the ordinal representation of the AQI value and might result in a more favorable visualization output in the subsequent process. As I mentioned previously, this report will analyze the impact of meteorological factors on air quality through both classification and prediction approaches. Given that the 'AQI quality' is an excellent nominal target for classification, it will be retained in the data set.

```
df.show(n=10)
df.columns
```

```
[8]: df.show(n=10)
     df.columns

+------+---+----+---+---+---+---+---+---+----+----+----+----+----+----+----+--------+--------------------+---+-----------+-----+----+---+---+-
---+---+
| Month|Day|   T|TM3|Tm4|SLP|  H| PP| VV|   V|  VM|  VG|  RA|  SN|  TS|  FG|    Date|        AQI Quality|AQI|AQI Ranking|PM2.5|PM10|So2|No2|
Co| O3|
+------+---+----+---+---+---+---+---+---+----+----+----+----+----+----+----+--------+--------------------+---+-----------+-----+----+---+---+-
---+---+
|Jan-21|  1|-3.2|2.5|-10|  -| 25|  0|6.4| 9.1|  18|   -|NULL|NULL|NULL|NULL| 1/01/21|               Good| 61|         84|   41|  66| 23| 49|
1.09| 29|
|Jan-21|  2|-1.5|  5| -8|  -| 30|  0|  6|   8|14.4|   -|NULL|NULL|NULL|NULL| 2/01/21|               Good| 69|        172|   48|  78| 30| 58|
1.28| 23|
|Jan-21|  3|-3.1|3.6| -8|  -| 45|  0|  5| 5.4|14.4|   -|NULL|NULL|NULL|NULL| 3/01/21|Moderately Polluted|194|        322|  148| 193| 66|100|
3.24|  8|
|Jan-21|  4| 2.2|8.6| -7|  -| 37|  0|4.3| 5.6|10.7|   -|NULL|NULL|NULL|NULL| 4/01/21|   Heavily Polluted|217|        286|  173| 224| 74|119|
3.67| 12|
|Jan-21|  5| 1.7| 11| -6|  -| 22|  0|4.2|  20|39.6|50.4|NULL|NULL|NULL|NULL| 5/01/21|Moderately Polluted|154|        216|  106| 156| 57| 70|
2.89| 30|
|Jan-21|  6|-1.1|  6| -8|  -| 23|  0|9.5|10.7|25.2|28.7|NULL|NULL|NULL|NULL| 6/01/21|               Good| 53|         90|   34|  51| 22| 40|
0.8| 37|
|Jan-21|  7|-2.6|3.3|-11|  -| 39|  0|6.1| 4.3|10.7|   -|NULL|NULL|NULL|NULL| 7/01/21|               Good| 85|        215|   61|  89| 38| 64|
1.77| 14|
|Jan-21|  8|-2.5|  5| -8|  -| 50|  0|6.3| 7.8|  18|   -|NULL|NULL|NULL|NULL| 8/01/21|Moderately Polluted|162|        315|  124| 151| 49| 91|
2.98|  7|
|Jan-21|  9|-0.9|7.5|-10|  -| 32|  0|  8| 9.8|32.4|43.2|NULL|NULL|NULL|NULL| 9/01/21|               Good| 99|        170|   73|  92| 32| 61|
1.57| 28|
|Jan-21| 10| 4.8| 13| -8|  -| 22|  0|6.1|  15|35.9|53.9|NULL|NULL|NULL|NULL|10/01/21|Moderately Polluted|191|        294|  146| 174| 54| 94|
2.8| 17|
+------+---+----+---+---+---+---+---+---+----+----+----+----+----+----+----+--------+--------------------+---+-----------+-----+----+---+---+-
---+---+
only showing top 10 rows
```

```
[8]: ['Month',
      'Day',
      'T',
      'TM3',
      'Tm4',
      'SLP',
      'H',
      'PP',
      'VV',
      'V',
      'VM',
      'VG',
      'RA',
      'SN',
      'TS',
      'FG',
      'Date',
      'AQI Quality',
      'AQI',
      'AQI Ranking',
      'PM2.5',
      'PM10',
      'So2',
      'No2',
      'Co',
      'O3']
```

Initial exploration of Data - show() function

```
df.printSchema()
```

```
[5]: df.printSchema()

root
 |-- Month: string (nullable = true)
 |-- Day: integer (nullable = true)
 |-- T: string (nullable = true)
 |-- TM3: string (nullable = true)
 |-- Tm4: string (nullable = true)
 |-- SLP: string (nullable = true)
 |-- H: string (nullable = true)
 |-- PP: string (nullable = true)
 |-- VV: string (nullable = true)
 |-- V: string (nullable = true)
 |-- VM: string (nullable = true)
 |-- VG: string (nullable = true)
 |-- RA: string (nullable = true)
 |-- SN: string (nullable = true)
 |-- TS: string (nullable = true)
 |-- FG: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- AQI Quality: string (nullable = true)
 |-- AQI: integer (nullable = true)
 |-- AQI Ranking: integer (nullable = true)
 |-- PM2.5: integer (nullable = true)
 |-- PM10: integer (nullable = true)
 |-- So2: integer (nullable = true)
 |-- No2: integer (nullable = true)
 |-- Co: double (nullable = true)
 |-- O3: integer (nullable = true)
```

Schema of dataset

df.describe().toPandas().transpose()

```
df.describe().toPandas().transpose()
[41]:
```

	0	1	2	3	4
summary	count	mean	stddev	min	max
Month	1096	None	None	Apr-21	Sep-23
Day	1096	15.732664233576642	8.805388067206271	1	31
T	1096	13.469597069597075	11.178302876535037	-	9.9
TM3	1096	19.27673992673994	11.41065196120622	-	9.7
Tm4	1096	6.925274725274726	11.448850999133077	-	9.9
SLP	1096	1026.5	4.666904755831149	-	1029.8
H	1096	52.55270394133822	21.46688765559118	-	98
PP	1096	1.987713997985899	10.518343879840211	-	9.91
VV	1096	6.349064748201441	2.7517395315858146	-	9.8
V	1096	10.480036630036645	5.1602269397916025	-	9.8
VM	1096	23.005402930403207	10.4054425106767	-	7.2
VG	1096	44.45928030303046	13.546862547100993	-	97
RA	295	None	None	-	o
SN	44	None	None	-	o
TS	91	None	None	-	o
FG	83	None	None	-	o
Date	1095	None	None	1/01/21	9/12/23
AQI Quality	1095	None	None	Excellent	Severely Polluted
AQI	1095	102.25022831050228	70.79867843303101	18	478
AQI Ranking	1095	228.95159817351598	120.90666182089399	1	367
PM25	1095	68.5607305936073	63.87000989509525	5	476
PM10	1095	92.45205479452055	73.49028485381326	0	772
So2	1095	9.325114155251141	11.055876834442596	1	81
No2	1095	45.20182648401826	22.983429297105697	10	152
Co	1095	1.1215159817351597	0.9636445726533793	0.22	8.11
O3	1095	58.35342465753425	38.478278312325024	2	177

Describe() function

```
df.describe().show()
```

```
+-------+------+-----------------+-----------------+-----------
```

```
pddf['So2'].describe()
```

```
count    1095.000000
mean        9.325114
std        11.055877
min         1.000000
25%         2.000000
50%         5.000000
75%        12.000000
max        81.000000
Name: So2, dtype: float64
```

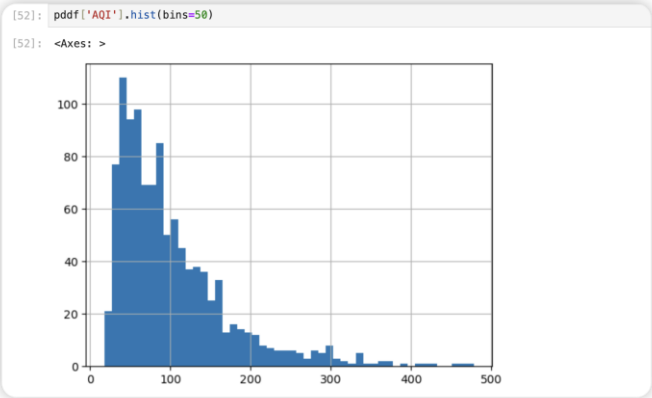
Pandas describe() function of the specific attribute

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

```
import matplotlib as mpl
import matplotlib.pyplot as plt
```

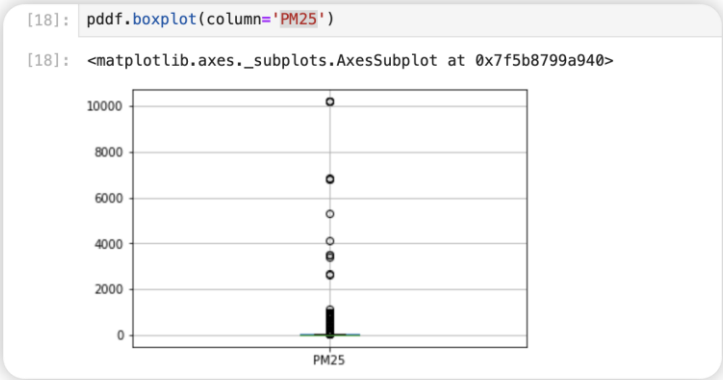
Import matplotlib

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



Histogram of AQI

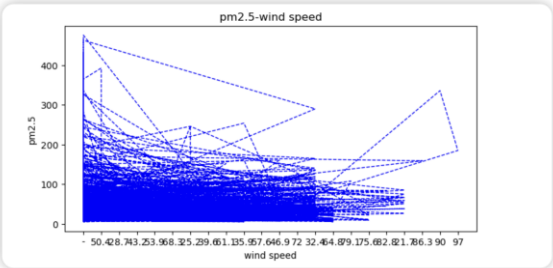
Also, the boxplot can help us to check the quality of our dataset initially.



PM2.5 Boxplot

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

```
x=pddf['VG']
y=pddf['PM25']
plt.figure(figsize=(8,4))
plt.plot(x,y,"b--",linewidth=1)
plt.xlabel('wind speed')
plt.ylabel('pm2.5')
plt.title('pm2.5-wind speed')
plt.show()
```



2.4 Data Quality Verification

Prior to initiating data mining, it is of paramount importance to guarantee data quality. Validating data quality not only ensures the correctness of our analytical outcomes but also lays a solid groundwork for subsequent endeavors in data mining.

2.4.1 Missing Values

```
df.printSchema()
df.toPandas().info()
print(df.toPandas().isnull().sum())
```

```
root
 |-- Month: string (nullable = true)
 |-- Day: integer (nullable = true)
 |-- T: string (nullable = true)
 |-- TM3: string (nullable = true)
 |-- Tm4: string (nullable = true)
 |-- SLP: string (nullable = true)
 |-- H: string (nullable = true)
 |-- PP: string (nullable = true)
 |-- VV: string (nullable = true)
 |-- V: string (nullable = true)
 |-- VM: string (nullable = true)
 |-- VG: string (nullable = true)
 |-- RA: string (nullable = true)
 |-- SN: string (nullable = true)
 |-- TS: string (nullable = true)
 |-- FG: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- AQI Quality: string (nullable = true)
 |-- AQI: integer (nullable = true)
 |-- AQI Ranking: integer (nullable = true)
 |-- PM25: integer (nullable = true)
 |-- PM10: integer (nullable = true)
 |-- So2: integer (nullable = true)
 |-- No2: integer (nullable = true)
 |-- Co: double (nullable = true)
 |-- O3: integer (nullable = true)
```

```
Data columns (total 26 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Month        1096 non-null   object
 1   Day          1096 non-null   int32
 2   T            1096 non-null   object
 3   TM3          1096 non-null   object
 4   Tm4          1096 non-null   object
 5   SLP          1096 non-null   object
 6   H            1096 non-null   object
 7   PP           1096 non-null   object
 8   VV           1096 non-null   object
 9   V            1096 non-null   object
 10  VM           1096 non-null   object
 11  VG           1096 non-null   object
 12  RA           295 non-null    object
 13  SN           44 non-null     object
 14  TS           91 non-null     object
 15  FG           83 non-null     object
 16  Date         1095 non-null   object
 17  AQI Quality  1095 non-null   object
 18  AQI          1095 non-null   float64
 19  AQI Ranking  1095 non-null   float64
 20  PM25         1095 non-null   float64
 21  PM10         1095 non-null   float64
 22  So2          1095 non-null   float64
 23  No2          1095 non-null   float64
 24  Co           1095 non-null   float64
 25  O3           1095 non-null   float64
```

```
Month             0
Day               0
T                 0
TM3               0
Tm4               0
SLP               0
H                 0
PP                0
VV                0
V                 0
VM                0
VG                0
RA              801
SN             1052
TS             1005
FG             1013
Date              1
AQI Quality       1
AQI               1
AQI Ranking       1
PM25              1
PM10              1
So2               1
No2               1
Co                1
O3                1
dtype: int64
```

Figure: Data Missing Value

The columns such as SLP, PP, VV, and VG are incomplete. This is because in the raw data, it uses "-" to denote the null value, and this has been demonstrated by the output presented above on the right

side. The columns of RA, SN, TS, and FG have a large number of missing values.

The records of air quality have 2 null values, and simultaneously, most of the meteorological records have 12 null values. This indicates that there are some entire rows that are completely missing in the data set, for example:

15-Dec	30	-0.4	4	-7	-	41	0	-	14.8	32.4	46.9					
15-Dec	31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16-Jan	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

Some of the meteorological data is missing, while the air quality records exist. These records all need to be preprocessed.

It is noticeable that the proportion of fully completed fields and complete records in our data set is relatively high, suggesting that only some simple additional processing of this data is required. Nevertheless, the variety of data types may imply that we will need to conduct further processing of the data in the aspect of feature engineering.

2.4.2 Outlier Detection

After excluding missing values due to data collection issues, we cross-validated

. These outliers were considered reasonable, potentially stemming from specific circumstances in data.

2.4.3 Data Duplication Check

```
count_all = df.groupBy(df.columns)\
.count()\
.where("count > 1")
print(count_all)
# no duplicates
# emrge with the original data set, retaining only duplicate rows
duplicates = df.join(count_all, on = df.columns, how="inner")
# show duplicates rows
duplicates.show(5)
```

```
DataFrame[Month: string, Day: int, T: string, TM3: string, Tm4: string, SLP: string, H: string, PP: string, VV: string, V: string, VM: strin
g, VG: string, RA: string, SN: string, TS: string, FG: string, Date: string, AQI Quality: string, AQI: int, AQI Ranking: int, PM25: int, PM1
0: int, So2: int, No2: int, Co: double, O3: int, count: bigint]
+-----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Month|Day|  T|TM3|Tm4|SLP|  H| PP| VV|  V| VM| VG| RA| SN| TS| FG|Date|AQI Quality|AQI|AQI Ranking|PM25|PM10|So2|No2| Co| O3|count|
+-----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
+-----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Data Duplication Check

Based on the results of the check for duplicate data in all data fields, it can be seen that our data does not have completely duplicated records.

3.Data Preparation

3.1 Data Selection

After attaining a more profound comprehension of our data, the selection of effective data assumes particular significance. The selected data should be highly relevant to our goals, taking into account factors such as data quality, technical constraints, and other considerations.

1.Remove unnecessary data:

The objective of our report is to discover the association between meteorological elements and air quality. Regarding some factors, the maximum value does not possess an actual value, whereas the average value is more representative and precise. In such situations, these values can be removed to achieve clear outcomes. For instance, columns like TM3, Tm4, VM, and VG represent the maximum values of corresponding factors like temperature or wind speed.

```
#Drop some columns - Reduce the dataset
pddf.drop(['TM3', 'Tm4' , 'VM', 'VG'],axis=1,inplace=True)
```

Drop Columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1096 entries, 0 to 1095
Data columns (total 22 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Month        1096 non-null   object
 1   Day          1096 non-null   int64
 2   T            1096 non-null   object
 3   SLP          1096 non-null   object
 4   H            1096 non-null   object
 5   PP           1096 non-null   object
 6   VV           1096 non-null   object
 7   V            1096 non-null   object
 8   RA           295 non-null    object
 9   SN           44 non-null     object
 10  TS           91 non-null     object
 11  FG           83 non-null     object
 12  Date         1095 non-null   object
 13  AQI Quality  1095 non-null   object
 14  AQI          1095 non-null   float64
 15  AQI Ranking  1095 non-null   float64
 16  PM25         1095 non-null   float64
 17  PM10         1095 non-null   float64
 18  So2          1095 non-null   float64
 19  No2          1095 non-null   float64
 20  Co           1095 non-null   float64
 21  O3           1095 non-null   float64
dtypes: float64(8), int64(1), object(13)
memory usage: 188.5+ KB
```

	Day	AQI	AQI Ranking	PM25	PM10	So2	No2	Co	O3
count	1096.000000	1095.000000	1095.000000	1095.000000	1095.000000	1095.000000	1095.000000	1095.000000	1095.000000
mean	15.732664	102.250228	228.951598	68.560731	92.452055	9.325114	45.201826	1.121516	58.353425
std	8.805388	70.798678	120.906662	63.870010	73.490285	11.055877	22.983429	0.963645	38.478278
min	1.000000	18.000000	1.000000	5.000000	0.000000	1.000000	10.000000	0.220000	2.000000
25%	8.000000	52.000000	115.000000	24.500000	42.000000	2.000000	29.000000	0.580000	28.500000
50%	16.000000	83.000000	264.000000	51.000000	76.000000	5.000000	40.000000	0.850000	53.000000
75%	23.000000	130.500000	344.000000	90.000000	120.000000	12.000000	56.000000	1.235000	82.000000
max	31.000000	478.000000	367.000000	476.000000	772.000000	81.000000	152.000000	8.110000	177.000000

After dropping some columns

2. Modify data:

I changed all "-" tags in the CSV file to null, and for binary columns, I changed all "o" tags to "1" and all empty records to "0".

```
pddf = pddf.replace("-", pd.NA)
binary_columns = ["RA", "SN", "TS", "FG"]
for col in binary_columns:
    pddf[col] = pddf[col].replace("o", 1).fillna(0)
```

By selecting these key variables, we can conduct a more focused analysis and ensure that our model is based on the most relevant data.

3.2 Data Clean

In the data mining process, data cleaning is a crucial step. To ensure the accuracy and reliability of our analysis results, it's essential to identify and address issues within the data.

Issues to Be Identified:

Missing Data: There may be missing values in our data set that require either imputation or removal.

Outliers: There might be outliers or anomalies in the data, which might be due to data entry mistakes or other reasons.

3.2.1 Cleaning Missing Values

When it comes to handling missing values, selecting the appropriate methods is of utmost importance, as this has a direct impact on the quality of the stroke data set and the precision of subsequent analyses.

In this report, we take 1000 records as the baseline number for analysis, because if there is not enough data, the results of the report may be misleading. In this case, due to the lack of data quality, these rows may need to be deleted in the later process.

```
#Drop Null Values
pddf.dropna(axis=0,how='any',inplace=True)
```

Pandas Clean NA

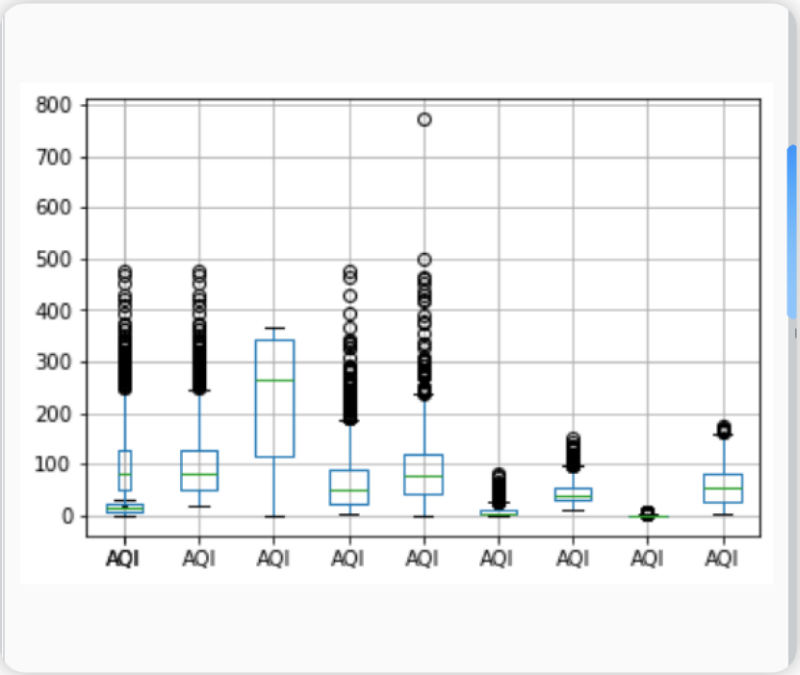
3.2.2 Cleaning Outliers

I composed this function (which requires an ordered index of the dataset) for deleting the abnormal values within each column of the

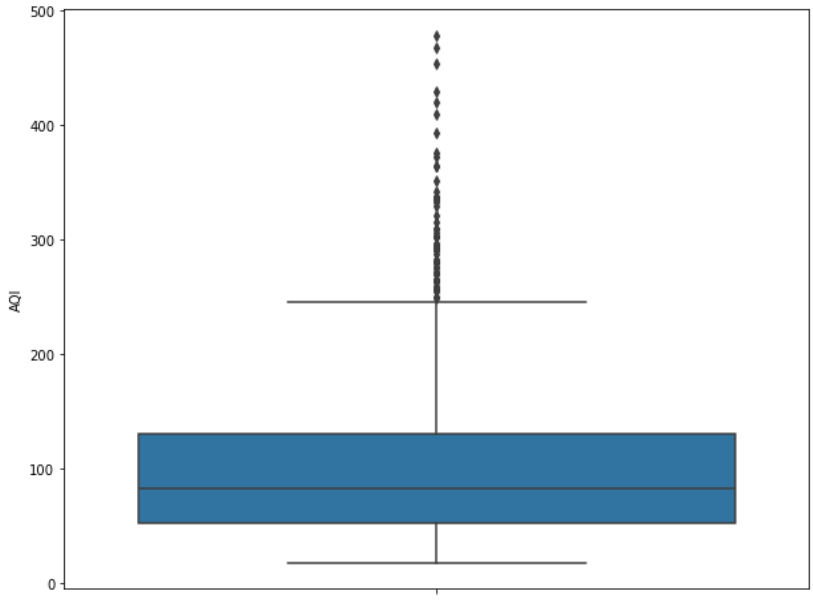
dataset, including the negative values as well as some extreme values, and the range is determined in accordance with the requirements of different datasets.

```
for i in range(pddf.index.max()):
    if any([
        'NR' in pddf.loc[i,:].values,
        pddf.loc[i,'Co']<0 or pddf.loc[i,'Co']>100,
        pddf.loc[i,'V']<0 or pddf.loc[i,'V']>100,
        pddf.loc[i,'PM25']>1000 or pddf.loc[i,'PM25']<0,
        pddf.loc[i,'So2']<0 or pddf.loc[i,'So2']>1000,
        pddf.loc[i,'No2']<0 or pddf.loc[i,'No2']>1000
        #Do the same thing with other attributes
        ]):

        #print('Deleted abnormal values in total %s rows'%i)
        pddf.drop([i],inplace=True)
        print('Deleted abnormal values in total %s rows'%i)
```

Delete abnormal values



At this stage, outliers and extreme values are also found.



Based on the results of the box plot, we can discover that AQI data with values exceeding 250 are regarded as outliers. These are the AQI qualities classified as "heavily polluted" and "severely polluted". Whether directly removing or replacing them with the mean may have a significant impact on our subsequent classification process. In this situation, the data remains unchanged.

3.3 Data Construction

During the process of data mining and analysis, merely relying on the existing features in the original dataset might sometimes be insufficient to fulfill our requirements. By constructing new features or variables, we can capture the characteristics of the data from diverse perspectives or dimensions, thereby enhancing the predictive power and interpretability of the model.

Here are the newly created features:

1. Season: Categorize months into spring, summer, autumn, winter to analyze seasonal trends.
2. Temperature Range: Calculate the difference between TM3 (max temperature) and Tm4 (min temperature) as a measure of temperature variability.
3. Precipitation Intensity: Classify PP (total rainfall/snowmelt) into categories (e.g., none, light, moderate, heavy) to assess the effect of rainfall intensity.
4. Combined Weather Metrics:
   Temperature-Humidity Index (THI): A measure of how hot it feels based on temperature and humidity.

Wind Chill Factor: A measure of how cold it feels based on temperature and wind speed.

```
from pyspark.sql.functions import when, col


# Create season column
df = df.withColumn("Season", when(col("Month").isin("12", "1", "2"), "Winter")
                                .when(col("Month").isin("3", "4", "5"), "Spring")
                                .when(col("Month").isin("6", "7", "8"), "Summer")
                                .otherwise("Autumn"))

# Create temperature range column
df = df.withColumn("TempRange", col("TM3") - col("Tm4"))

# Create temperature range column
df = df.withColumn("TempRange", col("TM3") - col("Tm4"))

# Create precipitation intensity column
df = df.withColumn("PrecipIntensity",
                   when(col("PP") == 0, "None")
                   .when(col("PP") <= 5, "Light")
                   .when(col("PP") <= 20, "Moderate")
                   .otherwise("Heavy"))

# Create Temperature-Humidity Index (THI) column
df = df.withColumn("THI", 0.8 * col("T") + col("H") / 100 * (col("T") - 14.4) + 46.4)

# Create Wind Chill Factor column (assuming wind speed in km/h)
df = df.withColumn("WindChill", 13.12 + 0.6215 * col("T") - 11.37 * col("V")**0.16 + 0.3965 *
col("T") * col("V")**0.16)


df.printSchema()
```

```
root
 |-- Month: string (nullable = true)
 |-- Day: integer (nullable = true)
 |-- T: string (nullable = true)
 |-- TM3: string (nullable = true)
 |-- Tm4: string (nullable = true)
 |-- SLP: string (nullable = true)
 |-- H: string (nullable = true)
 |-- PP: string (nullable = true)
 |-- VV: string (nullable = true)
 |-- V: string (nullable = true)
 |-- VM: string (nullable = true)
 |-- VG: string (nullable = true)
 |-- RA: string (nullable = true)
 |-- SN: string (nullable = true)
 |-- TS: string (nullable = true)
 |-- FG: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- AQI Quality: string (nullable = true)
 |-- AQI: integer (nullable = true)
 |-- AQI Ranking: integer (nullable = true)
 |-- PM25: integer (nullable = true)
 |-- PM10: integer (nullable = true)
 |-- So2: integer (nullable = true)
 |-- No2: integer (nullable = true)
 |-- Co: double (nullable = true)
 |-- O3: integer (nullable = true)
 |-- Season: string (nullable = false)
 |-- TempRange: double (nullable = true)
 |-- PrecipIntensity: string (nullable = false)
 |-- THI: double (nullable = true)
 |-- WindChill: double (nullable = true)
```

New Feature Results

The added features can assist us in conducting more thorough subsequent analyses.

Below are potential approaches for further data processing: Correlation Analysis: First, we can conduct a correlation analysis to determine the relationship between the new features and stroke. This will help us understand which features are most correlated and therefore likely to impact stroke the most.

Feature Importance: When building predictive models, we can assess the importance of each feature. This will further guide us in selecting which features to focus on during model optimization.

Data Visualization: By visualizing the new features, we can gain a more intuitive understanding of their distributions and their relationships with the target variable.

After creating these new features, we can use correlation analysis or other statistical methods to assess the relationship between these new features and the target variableto determine if they add value to the model.

3.4 Data Integration

In order to generate an integrated dataset, the meteorological information and air quality indicators of the same date need to be combined together. Some modifications need to be made before the data mining, for example, the date is recorded in both data source, and we may keep only 1 of them as the id of the data.

Checking for Missing Values and Outliers:

```
from pyspark.sql.functions import col, count, when

# Calculate missing values for each column
missing_value_counts = df.select(
    [count(when(col(c).isNull(), c)).alias(c) for c in df.columns]
).collect()[0].asDict()

# Print the results in a readable format
for column, count in missing_value_counts.items():
    print(f"{column}: {count} missing values")
```

Merging Two Datasets:

```
merged_df = pd.concat([df1, df2])
```

3.5 Data Reformatting

Following the process of data integration and purification, we advance to the data formatting stage. This step is of paramount importance as different data sources might have diverse data formats that could have an impact on subsequent data analysis and mining endeavors. After the data cleaning and integration, the current data format of ours is relatively well-structured, enabling us to move forward with the subsequent analysis and mining procedures.

```
df.describe()
df.describe().show()
```

```
[62]: df.describe()
      df.describe().show()

+-------+------+------------------+------------------+------------------+------------------+------------------+------------------+---------------
----+-----------------+-----------------+-----------------+-----------------+-----+----+----+----+-------+-----------------+---------------
----+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----
-+-----------------+-----------------+-----------------+-----------------+
|summary| Month|               Day|                 T|               TM3|               Tm4|               SLP|                 H|
PP|               VV|                 V|                VM|               VG|  RA|  SN|  TS|  FG|   Date|      AQI Quality|                A
QI|       AQI Ranking|              PM25|              PM10|               So2|               No2|                 Co|                 O3|Season|
TempRange|PrecipIntensity|               THI|         WindChill|
+-------+------+------------------+------------------+------------------+------------------+------------------+------------------+---------------
----+-----------------+-----------------+-----------------+-----------------+-----+----+----+----+-------+-----------------+---------------
----+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----
-+-----------------+-----------------+-----------------+-----------------+
|  count|  1096|              1096|              1096|              1096|              1096|              1096|              1096|
1096|              1096|              1096|              1096|              1096| 295|  44|  91|  83|   1095|              1095|
1095|              1095|              1095|              1095|              1095|              1095|              1095|              1095|  109
6|              1092|              1096|              1091|              1092|
|   mean|  NULL|15.732664233576642|13.469597069597075|19.27673992673994| 6.925274725274726|           1026.5|52.55270394133822| 1.98771399798
5899| 6.349064748201441|10.480036630036645|23.005402930403207| 44.45928030303046|NULL|NULL|NULL|NULL|   NULL|             NULL|102.2502283105
0228|228.95159817351598| 68.56073059360073|92.45205479452055| 9.325114155251141| 45.20182648401826|1.1215159817351597| 58.35342465753425|  NUL
L|12.351465201465201|           NULL|57.329450045829475| 12.80177260592697|
| stddev|  NULL| 8.805388067206271|11.178302876535037|11.41065196120622|11.448850999133077|4.666904755831149|21.46688765559118|10.51834387984
0211|2.7517395315858146|5.1602269397916025|  10.4054425106767|13.546862547100993|NULL|NULL|NULL|NULL|   NULL|             NULL| 70.7986784330
3101|120.90666182089399|63.87000989509525|73.49028485381326|11.055876834442596|22.983429297105697|0.9636445726533793|38.478278312325024|  NUL
L| 4.037762588836899|           NULL|14.839183656854582|13.383870524560866|
|    min|Apr-21|                1|                -|                -|                -|                -|                -|
-|                -|                -|                -|                -|   -|   -|   -|   -|-|1/01/21|        Excellent|                1
8|                1|                5|                0|                1|               10|             0.22|                2|Autumn|
1.0|          Heavy|             25.58|-27.26617913359144|
|    max|Sep-23|               31|               9.9|               9.7|               9.9|            1029.8|               98|
9.91|               9.8|               9.8|               7.2|               97|   o|   o|   o|   o|9/12/23|Severely Polluted|
478|               367|               476|               772|               81|               152|              8.11|               177|Autumn
|              23.0|           None| 83.960000000000001|35.922327738402066|
```

There is no considerable deviation in the data and it doesn't need to be revised.

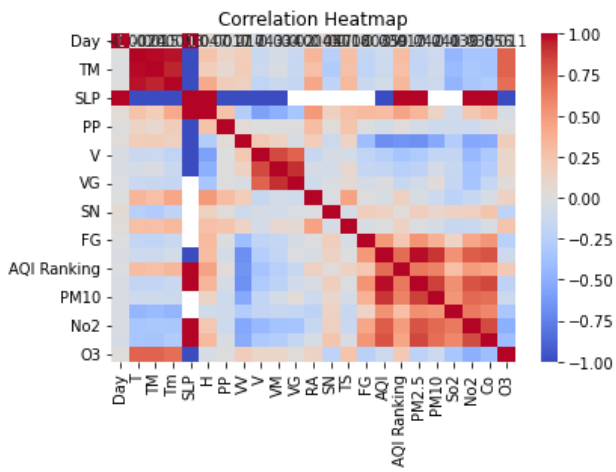
4.Data transformation

4.1 Reduce the data

Data transformation is a crucial part in the process of data preprocessing, with the purpose of converting the raw data into a format that is more suitable for data mining. In our data set, although we have multiple features such as TM, Tm, VM, not all of these features are directly related to the predictive target.

4.1.1 Horizontal Dimensionality Reduction

```
      Day    T    TM    Tm  SLP     H   PP   VV     V    VM    VG  RA  SN  TS  \
0       1 -3.2   2.5 -10.0  NaN  25.0  0.0  6.4   9.1  18.0   NaN   0   0   0
1       2 -1.5   5.0  -8.0  NaN  30.0  0.0  6.0   8.0  14.4   NaN   0   0   0
2       3 -3.1   3.6  -8.0  NaN  45.0  0.0  5.0   5.4  14.4   NaN   0   0   0
3       4  2.2   8.6  -7.0  NaN  37.0  0.0  4.3   5.6  10.7   NaN   0   0   0
4       5  1.7  11.0  -6.0  NaN  22.0  0.0  4.2  20.0  39.6  50.4   0   0   0
  ...  ...   ...   ...  ...   ...  ...  ...   ...   ...   ...  ..  ..  ..
1091   27 -3.3   2.0  -9.0  NaN  71.0  0.0  7.4   6.7  14.4   NaN   0   0   0
1092   28 -3.2   3.0 -10.0  NaN  77.0  0.0  5.3   4.1  10.7   NaN   0   0   0
1093   29 -3.4  -1.4  -9.0  NaN  91.0  NaN  1.8   5.2  10.7   NaN   0   1   0
1094   30 -1.4   7.0 -10.0  NaN  39.0  0.0  1.8   9.4  32.4  57.6   0   0   0
1095   31  NaN   NaN   NaN  NaN   NaN  NaN  NaN   NaN   NaN   NaN   0   0   0

      FG    AQI  AQI Ranking  PM2.5   PM10   So2    No2    Co    O3
0      0   61.0         84.0   41.0   66.0  23.0   49.0  1.09  29.0
1      0   69.0        172.0   48.0   78.0  30.0   58.0  1.28  23.0
2      0  194.0        322.0  148.0  193.0  66.0  100.0  3.24   8.0
3      0  217.0        286.0  173.0  224.0  74.0  119.0  3.67  12.0
4      0  154.0        216.0  106.0  156.0  57.0   70.0  2.89  30.0
  ..    ...          ...    ...    ...   ...    ...   ...   ...
1091   0  101.0        176.0   75.0   91.0  10.0   69.0  1.45   7.0
1092   1  140.0        240.0  105.0  120.0  11.0   68.0  1.86  10.0
1093   1  221.0        292.0  172.0  194.0   6.0   85.0  2.90   6.0
1094   1  157.0        254.0  116.0  129.0   4.0   62.0  2.03  22.0
1095   0   55.0         40.0   29.0   62.0   5.0   47.0  0.83  21.0
```



Repetitive date columns (Month, Day), as well as data with too much null values (SLP, VG, VV, PP) or with useless value (AQI ranking) for the goal. Then, we can remove all the rows with null values.

```
# delete columns:
```

```
columns_to_drop = ['TM', 'Tm', 'VM', 'VG', 'SLP', 'VV', 'PP', 'Month', 'Day', 'AQI Ranking']
df = df.drop(*columns_to_drop)    #
df.show(10)
print(f"Shape of the DataFrame: ({df.count()}, {len(df.columns)})")
```

After the reduction of the data, the result of the dataset is shown below:

```
+----+---+---+---+----+----+----+----+--------+--------------------+---+----+----+---+---+----+---+------+---------+---------------+-----
-+-------------------+
|   T|TM3|Tm4|  H|   V|  RA|  SN|  TS|  FG|    Date|        AQI Quality|AQI|PM25|PM10|So2|No2|  Co| O3|Season|TempRange|PrecipIntensity|   TH
I|          WindChill|
+----+---+---+---+----+----+----+----+----+--------+--------------------+---+----+----+---+---+----+---+------+---------+---------------+-----
-+-------------------+
|-3.2|2.5|-10| 25| 9.1|NULL|NULL|NULL|NULL| 1/01/21|               Good| 61|  41|  66| 23| 49|1.09| 29|Autumn|     12.5|           None| 39.4
4| -6.863825411400413|
|-1.5|  5| -8| 30|   8|NULL|NULL|NULL|NULL| 2/01/21|               Good| 69|  48|  78| 30| 58|1.28| 23|Autumn|     13.0|           None| 40.4
3| -4.500009281966013|
|-3.1|3.6| -8| 45| 5.4|NULL|NULL|NULL|NULL| 3/01/21|Moderately Polluted|194| 148| 193| 66|100|3.24|  8|Autumn|     11.6|           None|36.04
5| -5.308181321035066|
| 2.2|8.6| -7| 37| 5.6|NULL|NULL|NULL|NULL| 4/01/21|   Heavily Polluted|217| 173| 224| 74|119|3.67| 12|Autumn|     15.6|           None|43.64
6| 0.6578712292730902|
| 1.7| 11| -6| 22|  20|NULL|NULL|NULL|NULL| 5/01/21|Moderately Polluted|154| 106| 156| 57| 70|2.89| 30|Autumn|     17.0|           None|44.96
6| -3.097101898801483|
|-1.1|  6| -8| 23|10.7|NULL|NULL|NULL|NULL| 6/01/21|               Good| 53|  34|  51| 22| 40| 0.8| 37|Autumn|     14.0|           None|41.95
5|-4.8144683128221395|
|-2.6|3.3|-11| 39| 4.3|NULL|NULL|NULL|NULL| 7/01/21|               Good| 85|  61|  89| 38| 64|1.77| 14|Autumn|     14.3|           None| 37.6
9|  -4.15649139057647|
|-2.5|  5| -8| 50| 7.8|NULL|NULL|NULL|NULL| 8/01/21|Moderately Polluted|162| 124| 151| 49| 91|2.98|  7|Autumn|     13.0|           None| 35.9
5| -5.604826628534782|
|-0.9|7.5|-10| 32| 9.8|NULL|NULL|NULL|NULL| 9/01/21|               Good| 99|  73|  92| 32| 61|1.57| 28|Autumn|     17.5|           None|40.78
4| -4.335102620486404|
| 4.8| 13| -8| 22|  15|NULL|NULL|NULL|NULL|10/01/21|Moderately Polluted|191| 146| 174| 54| 94| 2.8| 17|Autumn|     21.0|           None|48.12
8| 1.5023612881236312|
+----+---+---+---+----+----+----+----+----+--------+--------------------+---+----+----+---+---+----+---+------+---------+---------------+-----
-+-------------------+
only showing top 10 rows
```

Data must be reduced through the selection of features relevant to the predictor (horizontal reduction) and/or vertical reduction. This could be achieved through the use of feature selection/logical processes.

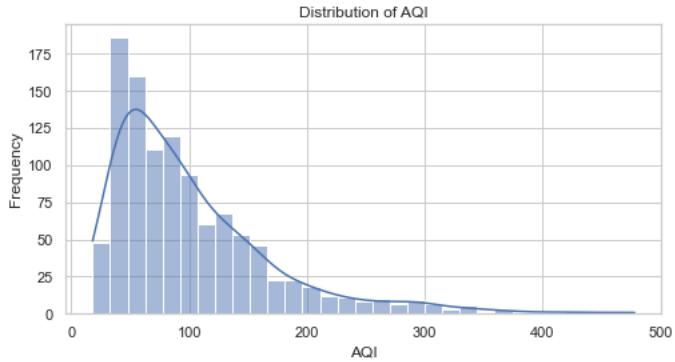
4.2 Project the data

Data projection not only functions to reduce the complexity of data but also to expose patterns or relationships that may be concealed within the data. In the era of big data, we frequently come across datasets that have high dimensionality, which makes it more difficult to understand and analyze the data intuitively. By using suitable data projection techniques, we can convert high-dimensional data into a lower-dimensional space, thereby facilitating the observation and interpretation of the data.

We are able to divide AQI into different clusters based on the categorization of the AQI as indicated below.

AQI	Air Pollution Category
0–50	Excellent
51–100	Good
101–150	Lightly Polluted
151–200	Moderately Polluted
201–300	Heavily Polluted
>300	Severely Polluted

In this way, the distribution of AQI in various level can be directly visualized by each variable, after it is divided into 6 groups, the graph appears to be more representative:



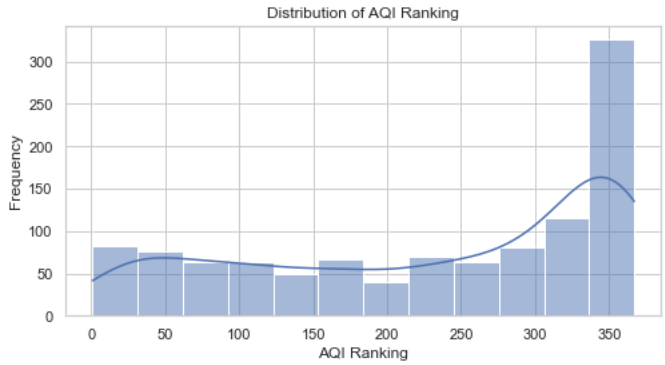


Figure 24 Data Value Distribution

modify the type of each factors using “dtypes” function:

```python
from pyspark.sql.functions import col
from pyspark.sql.types import FloatType, BooleanType

# Print initial schema and column types
print("Original DataFrame Schema:")
df.printSchema()

# Cast columns to desired types
df = (
    df.withColumn("T", col("T").cast(FloatType()))
    .withColumn("H", col("H").cast(FloatType()))
    .withColumn("V", col("V").cast(FloatType()))
    .withColumn("RA", col("RA").cast(BooleanType()))
    .withColumn("SN", col("SN").cast(BooleanType()))
    .withColumn("TS", col("TS").cast(BooleanType()))
    .withColumn("FG", col("FG").cast(BooleanType()))
)

# Print updated schema and column types
print("\nModified DataFrame Schema:")
df.printSchema()
```

```
Original DataFrame Schema:
root
 |-- T: string (nullable = true)
 |-- TM3: string (nullable = true)
 |-- Tm4: string (nullable = true)
 |-- H: string (nullable = true)
 |-- V: string (nullable = true)
 |-- RA: string (nullable = true)
 |-- SN: string (nullable = true)
 |-- TS: string (nullable = true)
 |-- FG: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- AQI Quality: string (nullable = true)
 |-- AQI: integer (nullable = true)
 |-- PM25: integer (nullable = true)
 |-- PM10: integer (nullable = true)
 |-- So2: integer (nullable = true)
 |-- No2: integer (nullable = true)
 |-- Co: double (nullable = true)
 |-- O3: integer (nullable = true)
 |-- Season: string (nullable = false)
 |-- TempRange: double (nullable = true)
 |-- PrecipIntensity: string (nullable = false)
 |-- THI: double (nullable = true)
 |-- WindChill: double (nullable = true)
```

```
Modified DataFrame Schema:
root
 |-- T: float (nullable = true)
 |-- TM3: string (nullable = true)
 |-- Tm4: string (nullable = true)
 |-- H: float (nullable = true)
 |-- V: float (nullable = true)
 |-- RA: boolean (nullable = true)
 |-- SN: boolean (nullable = true)
 |-- TS: boolean (nullable = true)
 |-- FG: boolean (nullable = true)
 |-- Date: string (nullable = true)
 |-- AQI Quality: string (nullable = true)
 |-- AQI: integer (nullable = true)
 |-- PM25: integer (nullable = true)
 |-- PM10: integer (nullable = true)
 |-- So2: integer (nullable = true)
 |-- No2: integer (nullable = true)
 |-- Co: double (nullable = true)
 |-- O3: integer (nullable = true)
 |-- Season: string (nullable = false)
 |-- TempRange: double (nullable = true)
 |-- PrecipIntensity: string (nullable = false)
 |-- THI: double (nullable = true)
 |-- WindChill: double (nullable = true)
```

5.Data-mining Method Selection

5.1 - Match and discuss DM methods within the context of the DM objectives.

The data mining methods are generally categorized into two types, namely, supervised learning and unsupervised learning. In this report, considering that we already know the output (AQI) and the input (meteorological factors), the supervised learning method would be the suitable choice for conducting further studies.

Typically, there are two forms of supervised learning: classification and prediction. Regression and classification share many similarities, and the major distinction lies in the range of values their outputs possess. In the case of classification, the output is usually several values, which are referred to as labels; for regression, the output can be any real number.

For our project, since we have two types of outcomes—AQI (a continuous value) and AQI Quality (a label), we will employ both methods.

The objective of this report is to explore the association between meteorological conditions and air quality. Many of the meteorological factors in the data, such as temperature and humidity, as well as the target—AQI, are continuous values. To model continuous-valued functions, prediction is utilized here for regression analysis. If a regression relationship exists, some of these factors can be potentially used to predict air quality.

In addition, the dataset also encompasses binary values, such as whether it rains on a particular day. For these factors, a classifier will be established to better predict categorical labels, such as "snow" or "no snow", for meteorological data.

Consequently, both prediction and classification are regarded as suitable data mining methods for this report.

5.2 Select the appropriate data-mining method(s) based on discussion

In a data mining project, the selection of appropriate methods is of utmost importance as it directly influences our capability to successfully accomplish data mining objectives and fulfill the success criteria. The following are the data mining methods that we have selected for this project, along with their corresponding associations with the data mining objectives:

1. Decision Tree Analysis

Method Description: Decision trees are a widely used classification and regression technique. It involves recursively partitioning the dataset into subsets based on specific feature values. Each decision tree node represents a decision-making point corresponding to a feature, and it branches down the tree to reach leaf nodes that offer predictions []. One of the advantages of this method is that it provides clear and intuitive insights into decisions, and it is capable of handling both numeric and categorical data.

Applicable Objectives: The relationship between AQI and other factors, such as the relationship between AQI and pm2.5, and the relationship between AQI and O3.

2. Random Forest

Random Forest is an ensemble approach that improves the overall prediction accuracy and steadiness by aggregating the predictions from multiple decision trees. Each tree is trained on a randomly chosen subset of the data and only takes into account a subset of randomly selected features for each node splitting. Subsequently, the predictions of all the trees are either averaged or voted on to produce the final prediction.

Applicable Objectives: In depth analysis of the relationship between AQI and other factors

3. LGB (LightGBM)

LightGBM is a tree-based gradient boosting framework that demonstrates outstanding performance in handling large-scale datasets. In comparison with conventional GBM, LightGBM trains models at a faster pace and consumes less memory.

Applicable Objectives: Effective prediction of air quality

These three methods are widely applied in the domains of data mining and machine learning, and they have shown their efficacy and dependability in diverse situations.

6.Data-mining Algorithm(s) Selection

6.1 Conduct exploratory analysis and Discuss

In this part, the selection of suitable algorithms is a crucial decision-making activity. This choice has a direct bearing on the success of the project, because the performance, adaptability, and

interpretability of the algorithms can exert a significant influence on the results. Hence, it is of utmost importance to have a thorough comprehension of the features and applicable scenarios of each algorithm to guarantee that our choice fulfills the project's needs.

Firstly, defining the objectives of data mining is of utmost importance. Usually, different objectives frequently necessitate different methods to be achieved. For classification problems, algorithms that are able to distinguish between various categories are needed, while for regression problems, algorithms that can predict continuous values are required. Moreover, association rule mining and cluster analysis have their respective specific algorithms.

The nature of the data plays a crucial role in algorithm selection. Distribution, scale, dimensionality, and the presence of noise or outliers all significantly impact which algorithms perform best. Some algorithms are particularly adept at handling massive datasets, while others are better suited for high-dimensional data with many variables. Additionally, algorithms can be sensitive to data quality, with noise and outliers potentially hindering their performance.

Selecting data mining algorithms is a multifaceted process. A thorough understanding of each algorithm's capabilities, coupled with a clear grasp of project goals, is critical. This informed decision-making ensures not only accurate results but also practical benefits from your data mining endeavors.

The variables in this project encompass continuous values such as temperature and binary values like whether it rains on a specific day. The targets are also categorized into two possibilities: the continuous AQI value or the categorical AQI Quality level. Hence, algorithms for both classification and prediction methods need to be taken into account.

6.2 select algorithms

Our data mining project will leverage both regression and classification algorithms. Binomial classification, ideal for datasets

with two labels (positive/negative), won't be suitable here due to the multinomial nature of our AQI data (more than two categories). However, common regression algorithms are well-suited for our continuous, float-based target variable.

Therefore, we've chosen the following algorithms:

- Classification: Decision Tree, Logistic Regression, Random Forest Classifier
- Prediction: Linear Regression

6.3 Build/Select appropriate model(s) and choose parameters

In this part, it is likely that continuous parameters will yield a better outcome when paired with a continuous target, and binary parameters will perform better with a categorical target. Hence, we will attempt to utilize Linear Regression to analyze the temperature, humidity, rainfall, and wind speed, while implementing decision tree, logistic regression, and random forest for binary variables such as rain, storm, fog, and snow.

```
[75]: from pyspark.ml.feature import VectorAssembler
      from pyspark.ml.regression import LinearRegression

[76]: spark_df.describe().toPandas().transpose()
```

Linear Regression

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

```
[88]: spark_df.describe().toPandas().transpose()
```

[88]:

	0	1	2	3	4
summary	count	mean	stddev	min	max
Month	1096	None	None	Apr-21	Sep-23
Day	1096	15.732664233576642	8.805388067206271	1	31
T	1096	13.469597069597068	11.178302876535033	-	9.9
TM3	1096	19.27673992673993	11.410651961206215	-	9.7
Tm4	1096	6.925274725274725	11.448850999133079	-	9.9
SLP	1096	1026.5	4.666904755831149	-	1029.8
H	1096	52.55270394133822	21.46688765559119	-	98
PP	1096	1.987713997985901	10.518343879840208	-	9.91
VV	1096	6.34906474820144	2.751739531585815	-	9.8
V	1096	10.480036630036627	5.160226939791607	-	9.8
VM	1096	23.005402930402948	10.4054425106767	-	7.2
VG	1096	44.45928030303032	13.546862547100998	-	97

Describe result transformation

```
[94]: #Organise the input columns - Select the data
vector_assembler = VectorAssembler(inputCols = ['AQI','PM25','PM10','So2','No2','Co','O3'], outputCol = 'features')
vector_output = vector_assembler.transform(spark_df)
```

Set target and transformation

```
vector_output.printSchema()
vector_output.head(1)
```

```
[81]: [Row(Month='Jan-21', Day=1, T='-3.2', TM3='2.5', Tm4='-10', SLP='-', H='25', PP='0', VV='6.4', V='9.1', VM='18', VG='-', RA='NaN', SN='NaN',
TS='NaN', FG='NaN', Date='1/01/21', AQI Quality='Good', AQI=61.0, AQI Ranking=84.0, PM25=41.0, PM10=66.0, So2=23.0, No2=49.0, Co=1.09, O3=29.
0, features=DenseVector([41.0, 66.0, 23.0, 49.0, 1.09, 29.0]))]
```

The regression model:

```
#Regressions:
from pyspark.ml.regression import LinearRegression
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml import Pipeline

# 1. Feature Vector Assembly:

feature_cols = ['T', 'H', 'V', 'PM2.5', 'PM10', 'So2', 'No2', 'Co', 'O3']
assembler = VectorAssembler(inputCols=feature_cols, outputCol='features')
df = assembler.transform(df)

# 2. Split Data:
```

```
# Use randomSplit for a roughly 70/30 split
(trainingData, testData) = df.randomSplit([0.7, 0.3], seed=1)

# 3. Build the Linear Regression Model:

lr = LinearRegression(featuresCol='features', labelCol='AQI')
model = lr.fit(trainingData)

# 4. Make Predictions:

predictions = model.transform(testData)

# 5. Evaluate the Model:

evaluator = RegressionEvaluator(predictionCol="prediction", labelCol="AQI", metricName="rmse")
rmse = evaluator.evaluate(predictions)

print(f"Root Mean Squared Error (RMSE) on test data = {rmse:.2f}")

# Output the model coefficients and intercept
print("Coefficients: \n" + str(model.coefficients))
print("Intercept: \n" + str(model.intercept))
```

The classification models:

```
from pyspark.ml.classification import LogisticRegression, DecisionTreeClassifier,
RandomForestClassifier
from pyspark.ml.feature import VectorAssembler, StringIndexer
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml import Pipeline
from pyspark.sql.functions import col, when, mean, count, isin

# Define a function to build and evaluate a classification model
def cmodel(classifier, df, predictors, label_col):
    # Create label indexer
    labelIndexer = StringIndexer(inputCol=label_col, outputCol="indexedLabel").fit(df)

    # Assemble features into a vector
    assembler = VectorAssembler(inputCols=predictors, outputCol="features")

    # Chain indexers and model in a Pipeline
    pipeline = Pipeline(stages=[labelIndexer, assembler, classifier])

    # Split the data into training and test sets (70% held out for training)
    (trainingData, testData) = df.randomSplit([0.7, 0.3], seed=1)
```

```
    # Train model
    model = pipeline.fit(trainingData)

    # Make predictions
    predictions = model.transform(testData)

    # Select example rows to display.
    predictions.select("prediction", "indexedLabel", "features").show(5)

    # Evaluate model
    evaluator = MulticlassClassificationEvaluator(
        labelCol="indexedLabel", predictionCol="prediction", metricName="accuracy")
    accuracy = evaluator.evaluate(predictions)
    print(f"Accuracy of {classifier} = {accuracy}")

# Assuming you have your filtered_df from previous data processing steps
high_aqi_df = filtered_df.filter(col("AQI") > 300)

# Get descriptive statistics for predictor columns in high_aqi_df
predictor_stats = high_aqi_df.select(predictors).describe()

# Show the descriptive statistics
print("\nDescriptive Statistics for High AQI Predictors:")
predictor_stats.show()

# Define predictor columns for classification models
predictors = ['RA','SN','TS','FG']

# Build and evaluate the classification models
cmodel(LogisticRegression(), df, predictors, 'AQI')
cmodel(DecisionTreeClassifier(), df, predictors, 'AQI')
cmodel(RandomForestClassifier(), df, predictors, 'AQI')
```

Method	**Main Advantage**	**Chosen Parameters**	**Purpose of Parameters**
LinearRegression	Fast results with optimizations for large datasets; combines predictions to control overfitting.	Learning rate, Number of trees, Maximum number of leaves	Control model complexity and overfitting; determine model correction rate in each iteration.

7.Data Mining

7.1 Create and justify test designs

The primary objective is to carry out the data mining as scheduled in steps 5 to 6 and verify their accuracy through the analysis node. The dataset is divided into two separate datasets, with the training data being utilized to build the model and the testing data being employed to test and confirm the model's competence. For this step, a 70/30 training/testing split is adopted, considering that the total amount of the data is only 1000. A small testing set might result in significant deviation. Therefore, to ensure the accuracy of the testing outcome, the size of the testing partition is increased to 30%.

We chose a 70/30 training/testing split for the following reasons:
• This split ensures that there is enough data for training the model to capture
the main features and patterns in the data.
• At the same time, it reserves enough data for testing, allowing for an effective
assessment of the model's generalization ability.

In [19]:
```
#Split method 2
splits = vector_output.randomSplit([0.7, 0.3])
train_df = splits[0]
test_df = splits[1]
```

```
Training set: 70.00%
Testing set: 30.00%
```

7.2 Conduct data mining

Linear Regression:

```
from sklearn.cross_validation import train_test_split
#Regressions:

from pyspark.ml.regression import LinearRegression
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml import Pipeline
from pyspark.sql.functions import col, when, regexp_replace, mean
from pyspark.sql.types import FloatType

import numpy as np

# Input validation and data preparation (same as before)
# (Assuming you've already read the CSV into 'df' and cleaned/converted data as needed)
from pyspark.sql.functions import col, when, regexp_replace

# Filter out rows where AQI is not numeric or contains '-'
filtered_df = df.filter((col("AQI") != "-") & (col("AQI").rlike("^[0-9]*\.?[0-9]*$")))

# Convert columns to float
for column_name in ['T', 'H', 'V', 'PM25', 'PM10', 'So2', 'No2', 'Co', 'O3']:
    filtered_df = filtered_df.withColumn(column_name, col(column_name).cast("float"))

# Convert AQI to numeric (float) and fill missing values with the mean


filtered_df = df.filter(col("AQI").rlike("^[0-9]*\.?[0-9]*$"))
filtered_df = filtered_df.withColumn("AQI", col("AQI").cast("float"))
mean_aqi = filtered_df.select(mean(col("AQI"))).collect()[0][0]
filtered_df = filtered_df.fillna(mean_aqi, subset=["AQI"])



# Drop rows with NaN values in AQI after conversion
df = filtered_df.na.drop(subset=["AQI"])

# Feature vector assembly
featureCols = ['PM25','PM10','So2','No2','Co','O3']
assembler = VectorAssembler(inputCols=featureCols, outputCol='features')
```

```
# Create a LinearRegression object
linreg = LinearRegression(featuresCol='features', labelCol='AQI')

# Chain indexer and tree in a Pipeline
pipeline = Pipeline(stages=[assembler, linreg])

# Split the data into training and test sets (70% held out for training)
(trainingData, testData) = df.randomSplit([0.7, 0.3])

# Train model.    This also runs the indexer.
model = pipeline.fit(trainingData)

# Make predictions.
predictions = model.transform(testData)

# Select example rows to display.
predictions.select("prediction", "AQI", "features").show(5)

# Get evaluation metrics
evaluator_rmse = RegressionEvaluator(predictionCol="prediction", labelCol="AQI",
metricName="rmse")
rmse = evaluator_rmse.evaluate(predictions)

evaluator_mse = RegressionEvaluator(predictionCol="prediction", labelCol="AQI",
metricName="mse")
mse = evaluator_mse.evaluate(predictions)

evaluator_r2 = RegressionEvaluator(predictionCol="prediction", labelCol="AQI", metricName="r2")
r2 = evaluator_r2.evaluate(predictions)

# Print evaluation metrics
print("Root Mean Squared Error (RMSE) on test data = %g" % rmse)
print("Mean Squared Error (MSE) on test data = %g" % mse)
print("R-squared on test data = %g" % r2)

# Summary statistics of AQI
df.describe(['AQI']).show()
```

Classifications:

```
from pyspark.ml.classification import LogisticRegression, DecisionTreeClassifier,
RandomForestClassifier
from pyspark.ml.feature import VectorAssembler, StringIndexer
```

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator, BinaryClassificationEvaluator
from pyspark.ml import Pipeline
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.sql.functions import col, when, count

# Define a function to build and evaluate a classification model
def cmodel(classifier, df, predictors, label_col):
    # Convert label column to numeric using StringIndexer (if it's not already)
    labelIndexer = StringIndexer(inputCol=label_col, outputCol="indexedLabel").fit(df)

    # Assemble features into a vector
    assembler = VectorAssembler(inputCols=predictors, outputCol="features")

    # Chain indexer, assembler, and model in a Pipeline
    pipeline = Pipeline(stages=[labelIndexer, assembler, classifier])

    # Split the data into training and test sets (70% for training, 30% for testing)
    (trainingData, testData) = df.randomSplit([0.7, 0.3], seed=1)

    # Define parameter grid for cross-validation (adjust as needed)
    paramGrid = ParamGridBuilder() \
        .addGrid(classifier.maxDepth, [5, 10, 20]) \
        .addGrid(classifier.maxBins, [20, 32, 64]) \
        .build()

    # Create CrossValidator
    crossval = CrossValidator(estimator=pipeline,
                              estimatorParamMaps=paramGrid,
                              evaluator=BinaryClassificationEvaluator(labelCol="indexedLabel"),
                              numFolds=5)    # 5-fold cross-validation

    # Train model with cross-validation
    cvModel = crossval.fit(trainingData)

    # Make predictions on the test data
    predictions = cvModel.transform(testData)

    # Select and display example rows
    predictions.select("prediction", "indexedLabel", "features").show(5)

    # Evaluate the best model from cross-validation
    evaluator = BinaryClassificationEvaluator(labelCol="indexedLabel")
    accuracy = evaluator.evaluate(predictions)
    print(f"Test Accuracy of {classifier.__class__.__name__}: {accuracy:.3f}\n")
```

```
    # Show the best model parameters
    bestModel = cvModel.bestModel
    bestParams = bestModel.stages[-1].extractParamMap()
    print("Best Parameters:")
    for param, value in bestParams.items():
        print(f"\t{param}: {value}")
    print("\n")



# Create a binary label for high AQI
df = df.withColumn("HighAQI", when(col("AQI") > 300, 1.0).otherwise(0.0))
df = df.na.drop(subset=["HighAQI"])
df = df.withColumn("HighAQI", col("HighAQI").cast("int"))
# Filter out rows where AQI is not numeric or contains '-'
filtered_df = df.filter((col("AQI") != "-") & (col("AQI").rlike("^[0-9]*\.?[0-9]*$")))

# Convert columns to float
for column_name in ['T', 'H', 'V', 'PM2.5', 'PM10', 'So2', 'No2', 'Co', 'O3']:
    filtered_df = filtered_df.withColumn(column_name, col(column_name).cast("float"))

# Convert AQI to numeric (float) and fill missing values with the mean
filtered_df = filtered_df.withColumn("AQI", col("AQI").cast("float"))
mean_aqi = filtered_df.select(mean(col("AQI"))).collect()[0][0]
filtered_df = filtered_df.fillna(mean_aqi, subset=["AQI"])
df = filtered_df

# Get descriptive statistics for predictor columns in high_aqi_df
predictor_stats = high_aqi_df.select(predictor).describe()

# Show the descriptive statistics
print("\nDescriptive Statistics for High AQI Predictors:")
predictor_stats.show()

# Define predictor columns for classification models
predictors = ['RA','SN','TS','FG']

# Build and evaluate the classification models
cmodel(LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8), df, predictors,
'HighAQI')
cmodel(DecisionTreeClassifier(maxDepth=5, maxBins=32), df, predictors, 'HighAQI')
cmodel(RandomForestClassifier(numTrees=100, maxDepth=4, maxBins=32), df, predictors,
'HighAQI')
```

In the data mining, the prediction method is implemented with the Linear Regression algorithm. Also, the classification method is applied, using the decision tree, logistic regression, and random forest algorithms to analyze how meteorology affects air quality.

7.3 Search for patterns

Observing the models' performances reveals crucial patterns:

1. In the linear model, Co (-2.019), PM2.5 (0.961), and wind speed (0.562) are weighted the most. It seems that there is no relevant relationship between AQI and certain meteorological factors, like temperature (0.050) and humidity (0.040).

```
[0.44980179]
[[ 0.0500712   0.03979566  0.56181032  0.96055723  0.12464265
0.05712086
   0.17555371 -2.01920139  0.16341463]]
```

2. During the generation of a Decision tree model, we can effortlessly find the total amounts of each meteorological binary factor for each AQI quality. When the air is in a severely polluted state, it seems that there is no rain or storm meteorological situation.

	RA	SN	TS	FG
count	27	27	27	27
unique	1	2	1	2
top	False	False	False	True
freq	27	22	27	19

8.Interpretation

8.1 Study and discuss the mined patterns

In the realm of data science, simply constructing models and attaining results is not the culmination of the process. It is of paramount importance to comprehend and analyze these results, unraveling the patterns that arise from both the data and the models employed.

8.1.1 Understanding the Patterns

Decision Trees:

Transparent and Intuitive: Decision trees excel in offering a clear visual representation of how different factors influence stroke risk. You can easily see how each variable splits the data, making it a great choice for understanding the decision-making process.

Complexity vs. Accuracy: Increasing the depth of the tree allows it to capture more complex relationships, potentially leading to better predictions. However, this comes at the cost of increased complexity and computational demands. A very deep tree can overfit the data, meaning it performs well on the training data but may not generalize well to unseen data.

Random Forests:

Ensemble Power: Random forests leverage the power of ensembles. They build multiple decision trees and average their predictions, which helps to reduce overfitting compared to single decision trees.

Accuracy and Efficiency Trade-off: Similar to decision trees, random forests can benefit from increased tree depth, allowing them to capture finer patterns and potentially improve accuracy. However, this also increases computational requirements.

Linear models:

Simplicity: The structure of the model is relatively simple and easy to understand and interpret.

Additivity: The effects of variables can be linearly added.

Predictability: It can make certain predictions for the data.

8.1.2 Discussion of Mined Patterns

Feature Importance: Understanding which features or variables have the most significant impact on the model's predictions is critical.

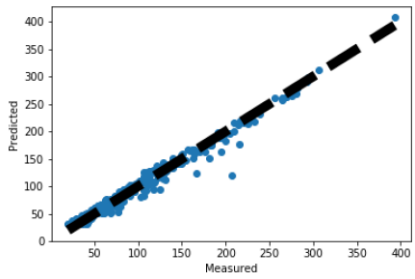
Model Complexity and Interpretability: In the context of these models, there exists a direct trade-off between model complexity (such as the depth of the tree) and its interpretability. Although deeper trees have the potential to capture more complex relationships and achieve higher accuracies, they become increasingly difficult to interpret and demand more computational resources. This trade-off requires thorough consideration, particularly when implementing models in real-world scenarios where interpretability can be as important as accuracy.

In conclusion, the mined patterns provide profound insights into air pollution prediction.

Focus on our project diagram:

```
[0.44980179]
[[ 0.0500712   0.03979566  0.56181032  0.96055723  0.12464265
0.05712086
   0.17555371 -2.01920139  0.16341463]]
```

```
Root Mean Squared Error (RMSE) on test data = 10.402
Mean Squared Error (MSE) on test data = 108.203
R-squared on test data = 0.981363
```



From the data mining result presented in the prediction part above, it can be observed that the R2 value (which ranges from 0 to 1 and in this case is 0.981), representing the predictive capability of the Linear Regression model, is extremely close to 1. This indicates the existence of a highly linear relationship. In addition, based on the Root Mean Squared Error (RMSE), which gauges how closely the predictions approach the actual results, with a lower value being better, we can determine that the prediction outcome (9.797) is also

very close to the actual value. When comparing it with the actual "AQI" value, including its mean, minimum, and maximum, this becomes evident.

```
+-------+------------------+
|summary|               AQI|
+-------+------------------+
|  count|              1095|
|   mean|102.25022831050228|
| stddev| 70.79867843303101|
|    min|              18.0|
|    max|             478.0|
+-------+------------------+
```

Conversely, the results of the classification output obtained from binary values are not good (the accuracy is only 0.306%).

```
In [158]: cmodel1(LogisticRegression(), df,predictor,'AQI')
Accuracy:0.306%

Cross-Validation Score : 0.459%
Accuracy:0.306%
Cross-Validation Score : 0.550%

In [159]: cmodel1(DecisionTreeClassifier(), df,predictor,'AQI')

In [160]: cmodel1(RandomForestClassifier(), df,predictor,'AQI')
Accuracy:0.306%
Cross-Validation Score : 0.917%
```

This reveals that meteorological factors like rain, storm, fog, and snow do not directly affect the quality of AQI.

8.2 Visualize the data, results, models, and patterns

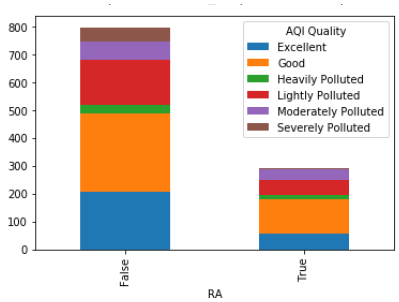
Visualization has a key role in understanding complex data sets and model outputs.

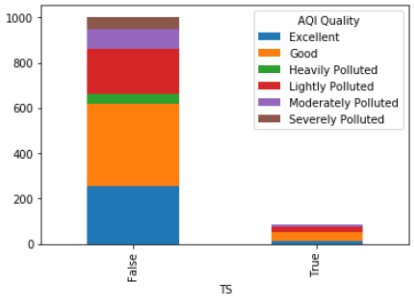
It presents an intuitive way to perceive the relationships, distributions, and patterns that are latent within the data.

Let's take a look linear models:

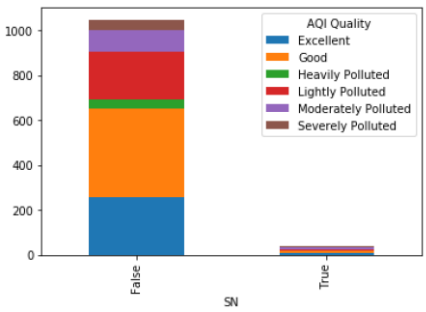
```
In [161]: print (linreg.intercept_)
     ...: print (linreg.coef_)
[0.44980179]
[[ 0.0500712   0.03979566  0.56181032  0.96055723  0.12464265
0.05712086
   0.17555371 -2.01920139  0.16341463]]

In [162]: X.head()
Out[162]:
     T     H     V  PM2.5   PM10   So2    No2    Co    O3
```

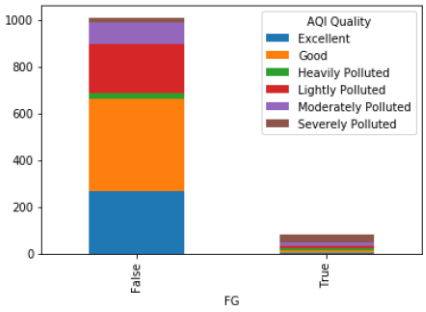
Based on this table, it can be known that CO (feature weight equals -2.019) and pm2.5 (feature weight is 0.961) are the most crucial predictors for AQI, while the importance of predictors for other pollutants is relatively low, namely, NO2 (0.176), O3 (0.163), PM10 (0.125), and SO2 (0.057). Compared with these values, the importance of predictors for meteorological factors is lower, namely, V (0.562), H (0.040), and T (0.050).





The graphs presented above indicate that there is no occurrence of severely polluted air quality when rain or storm arrives, suggesting that they might have a positive impact on the air. In contrast, more severe pollution phenomena occur when fog appears.

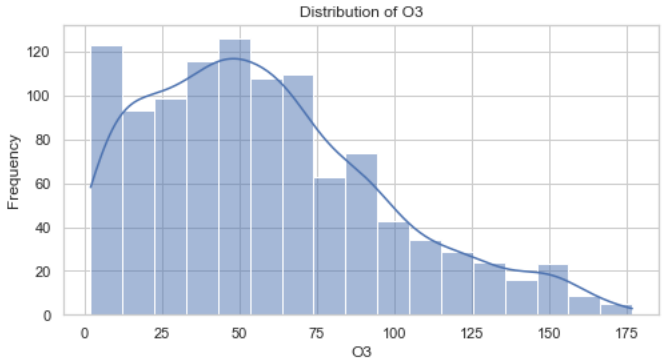


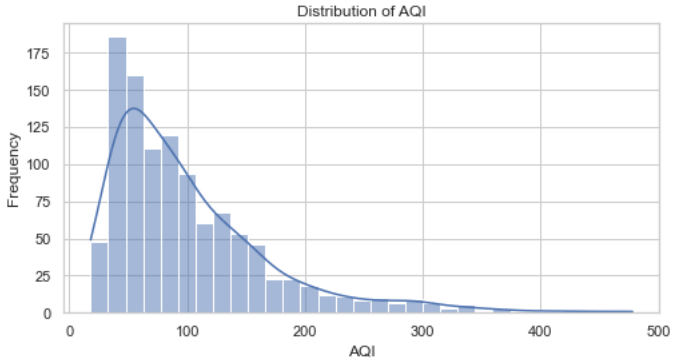


Upon comparing these two groups, it can be concluded that the proportions of excellent, good, mild, and moderate AQI in the cases of RA (rain) and TS (storm) are higher than those of SN (snow) and FG (fog). Overall, rain (RA) and TS (storm) have a favorable influence on the quality of AQI, the impact of snow (SN) remains uncertain, and fog (FG) appears to have a detrimental effect on the air quality.

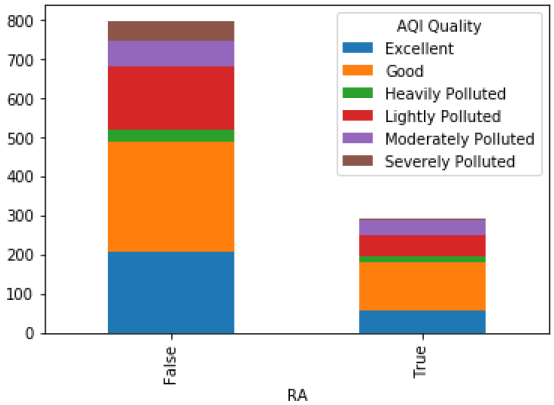
Data Visualization:

Histograms and box plots for continuous variables can depict their distribution and identify outliers.





Pie charts or bar graphs can represent categorical variables, giving insights into the data's composition



Correlation matrices and heat maps illustrate the relationships between variables, helping to understand multicollinearity.

Model Results Visualization:

Feature importance plots: Especially in tree-based models like Decision Trees, Random Forest, these plots highlight the most influential predictors.

8.3 Interpret the results, models, and patterns

8.3.1 Results

The Decision Tree serves as our initial point. Although its performance may not be at the highest level, it offers a fundamental comprehension of our data. The step-by-step decision-making process it presents visually demonstrates which factors are of utmost importance in making predictions.

Random Forest takes the essence of decision trees and amplifies it. Its ensemble nature makes it resilient to overfitting, allowing it to dive deeper into the data's intricacies.

8.3.2 Models

The architecture of the Decision Tree unveiled how variables hierarchically exert influence on the outcome. Random Forest, through averaging multiple trees, not only brings about a performance enhancement but also provides insights into the variability in the importance of features.

8.3.3 Patterns

As trees go deeper, they capture more granular relationships.

The interpretation of the patterns and results found during the data mining is as followed:

1. By using both prediction and classification methods, we find out that meteorological phenomenon are not decisive factors for the air quality.
2. Specific meteorological phenomenon may affect some circumstances (When storm comes, the air won't be severely polluted, and when fog come, the air quality won't be excellent and is very likely to be severely polluted). This is probably caused by the meteorological condition for the accumulation and spread of pollutant.
3. Comparing to other pollutants, PM2.5 and CO has the greatest impact on the AQI of Taipei

8.4 Assess and evaluate results, models, and patterns

The comprehensive evaluation of the results, models, and patterns is of utmost importance for extracting value from the entire data mining process. By conducting comparisons and evaluations, we are able to reach meaningful conclusions and develop implementable strategies.

Decision Tree: Suitable for scenarios with small data volumes and high feature interpretability requirements. For example, medical diagnosis, fraud detection, etc.

Random Forest: Suitable for large datasets with complex nonlinear relationships. For example, image recognition, natural language processing, etc.

LGB: Suitable for scenarios with large data volumes, many features, and high precision prediction requirements. For example, recommendation systems, advertising, etc.

Linear: Suitable for scenarios with large data volumes, low feature interpretability requirements, and linear relationships between data. For example, economic forecasting, market analysis, etc.

The impact of the meteorological phenomenon is lower than what I had anticipated. This can be attributed to two aspects: Firstly, it is not feasible to manage all the variables. The meteorological indicators vary daily, and the pollutants generated from different sources are challenging to be maintained at a specific level consistently over time. Secondly, meteorology is always a secondary factor influencing air quality. It neither creates nor eliminates pollutants in the air directly, but it does effectively facilitate the dispersion of polluted air when there is wind.

8.5 Iterate prior steps as required

To further verify the details, I sorted the data with AQI greater than 300 (severe polluted). As you noticed, there is no rain or storm

during these days (mostly in winter), which in this way is indeed a factor that is very important for the dispersion of pollutants.

T	TM	Tm	SLP	H	PP	VV	V	VM	VG	RA	SN	TS	FG	AQI	AQI Ranking	PM2.5	PM10	So2	No2	Co	O3
0.3	3.5	-2		80	0	1.1	10.4	35.9		0	0	0	1	478	366	476	461	35	136	8.11	5
-3.5	-2	-6		96	0	0.8	5.9	10.7		0	1	0	1	468	366	463	464	12	135	6.97	7
-4.4	0	-9		87		1.3	5.9	14.4		0	1	0	1	454	365	430	501	8	131	6.43	4
0.8	4	-4		73	0	1.4	6.5	32.4	50.4	0	0	0	1	429	363	393	429	10	152	7.97	4
21.3	25	15.9		20	0.25	2.9	16.7	61.1	90	0	0	0	0	420	354	336	772	2	21	0.29	79
-2.1	0	-5		94		1.1	5.6	10.7		0	1	0	1	409	342	365	422	8	133	7.67	4
-0.1	6	-9		80	0	2.1	6.9	14.4		0	0	0	1	393	360	344	452	18	145	7.28	8
7	11.8	2		50	0	1.9	10.9	28.7		0	0	0	0	376	364	326	277	46	113	4.38	11
-3.5	0	-7		81	0	1.3	4.3	25.2		0	0	0	1	372	366	328	375	34	111	4.73	5
-1.1	2.7	-4.2		94	0	1.1	4.3	7.2		0	0	0	1	364	358	339	337	23	121	5.71	7
-1.6	2	-5		54	0	4.7	9.4	46.9		0	0	0	0	364	359	337	419	81	69	2.62	16
-2.4	1	-7		90	0	1.3	5.6	10.7		0	0	0	1	351	347	308	304	29	104	5.41	7
16.3	22.2	12		70	0	2.6	6.7	14.4		0	0	0	1	342	366	291	179	13	114	2.46	49
9.9	18	3		60	0	3.1	6.5	10.7		0	0	0	0	338	362	287	291	30	99	2.25	49
-1.7	9	-9		67	0	2.4	4.8	14.4		0	0	0	1	338	354	290	380	17	137	5.61	9
2.7	8	-6		46	0	3.9	7	14.4		0	0	0	0	335	363	284	307	78	113	4.41	16
19.3	25	14.1		80	0	2.1	5.2	10.7		0	0	0	1	335	364	281	189	7	77	1.93	94
-3.1	1	-6		70	0	3.4	8.9	18	32.4	0	1	0	1	333	365	290	389	8	105	5.28	6
										0	0	0	0	329	353	280	329	21	127	4.4	4
7.2	15	0		64	0	2.4	5	10.7		0	0	0	1	321	366	271	281	44	114	3.38	23
19.9	25.4	14		47	0	2.6	15.4	32.4		0	0	0	0	315	364	265	96	7	74	1.81	64
-2.4	3.4	-7		76	0	2.3	4.6	14.4		0	0	0	1	310	362	263	303	32	114	4.16	6
7.9	11	3		88	0	1.4	5.6	14.4		0	0	0	1	309	364	258	172	16	79	3.56	5
1.7	10	-4		59	0	4	8.9	18	35.9	0	0	0	0	306	366	254	356	21	112	3.35	7

In order to check the linear relationship between meteorological factors and AQI, I make another iteration to check their relation using the same method and algorithm:

```
# check meteorological factors effect on AQI:

X1 = df[['T','H','V']]
X1.head()
Y1=df[['AQI']]
#split data
X1_train, X1_test, Y1_train, Y1_test = train_test_split(X1, Y1, test_size = 0.3, random_state=1)

linreg1 = LinearRegression()
linreg1.fit(X1_train, Y1_train)
# get intercept and coefficients
print (linreg1.intercept_)
print (linreg1.coef_)

Y1_pred = linreg1.predict(X1_test)

print ("MSE:",metrics.mean_squared_error(Y1_test, Y1_pred))
print ("RMSE:",np.sqrt(metrics.mean_squared_error(Y1_test, Y1_pred)))
print ("R2:",r2_score(Y1_test, Y1_pred))

df['AQI'].describe()
```

You can spot these codes right after the linear regression, and there is a comment indicating (Linear model with only meteorological

factors). This time, solely the meteorological factors are retained for the creation of the model, and the following is the output.

```
MSE: 3605.682708060186
RMSE: 60.04733722705934
R2: 0.1526196573372357
```

This time, the accuracy of the model is extremely low. The RMSE value is as high as 65.766 (a very large number when compared with the mean and standard deviation), and the R2 is merely 0.153. In comparison with the previous outcomes of the linear models, we can draw the conclusion that there is no linear connection between these chosen meteorological factors and the AQI.

This result proves our conclusion in the step 8.3.

After undergoing multiple iterations, we noticed that although there were certain differences in the specific prediction outcomes each time, the core conclusions and trends remained extremely consistent. This consistency offers us valuable insights, indicating that our model is stable and less prone to being affected by random noises or data perturbations. This stability builds a solid basis for the real-world applications of the model.

References:

1. Environmental Protection Administration (EPA) of Taiwan. (2023). Air Quality Monitoring Data.
2. World Health Organization (WHO). (2021). Air quality guidelines – global update 2021: particulate matter (PM2.5), ozone (O3), nitrogen dioxide (NO2), sulfur dioxide (SO2) and carbon monoxide (CO).
3.Brook, R. D., Rajagopalan, S., Pope III, C. A., Brook, J. R., Bhatnagar, A., Diez-Roux, A. V., ... & Kaufman, J. D. (2010). Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the American Heart Association. *Circulation*, *121*(21), 2331-2378.
3. Chen, H. H., & Schwartz, J. (2009). Air pollution and heart rate variability. *Environmental research*, *109*(5), 557-564.
4. Huang, C. L., & Wu, W. Y. (2011). Does haze hinder tourism? *Tourism Management*, *32*(2), 452-455.
5. World Bank. (2016). The cost of air pollution: strengthening the economic case for action.
6. Huang, Y. L., & Wang, C. H. (2016). Influence of meteorological factors on PM2.5 concentrations during heavy pollution episodes in a subtropical island city. *Environmental Science and Pollution Research*, *23*(21), 21655-21664.
7. Chen, S. J., Lai, C. H., Tsai, I. J., & Tsai, M. J. (2021). The impact of terrain on air quality: A case study of the Taipei Basin. *Atmospheric Environment*, *256*, 118461.
8. Lu, H. Y., & Chang, C. C. (2019). The impact of urbanization and industrialization on air pollution in Taiwan. *Sustainability*, *11*(16), 4422.
9. Hsieh, L. Y., & Chen, M. C. (2018). Challenges and opportunities for air quality monitoring in Taiwan. *Aerosol and Air Quality Research*, *18*(1), 1-14.
10. World Health Organization. (2021). Air quality guidelines – global update 2021: particulate matter (PM2.5), ozone (O3), nitrogen dioxide (NO2), sulfur dioxide (SO2) and carbon monoxide (CO).

Disclaimers:

"I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright. (See: https://www.auckland.ac.nz/en/students/forms-policies-and-guidelines/student-policies-and-guidelines/academic-integrity-copyright.htmlLinks to an external site.).

I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain, then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data."