

รายงาน

เรื่อง การใช้ข้อมูลฝุ่นละอองขนาดเล็ก PM 2.5 ย้อนหลัง 10 ปี
เพื่อหาจำนวนวันและช่วงเวลาที่มีค่าการสะสมเกินค่ามาตรฐาน
เพื่อใช้ประกอบการตัดสินใจออกนโยบายและแก้ไขปัญห

จัดทำโดย

นางสาวณภัทร	เสริมสุวรรณสุข	รหัสนักศึกษา 63606019
นายสุทิน	จันเพชร	รหัสนักศึกษา 63606020

เสนอ

รศ.ดร.วรพจน์ กรีสระเดช

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประจำปีการศึกษา 2563

ที่มาและความสำคัญ

ปัญหาฝุ่นละอองขนาดเล็ก (PM2.5) กำลังเป็นปัญหาใหญ่ที่เกิดขึ้นในประเทศไทยและภูมิภาคเอเชียตะวันออกเฉียงใต้ โดยมีแนวโน้มจะมีความรุนแรงขึ้นในทุกๆ ปี ด้วยประชากรหนาแน่น ความคับคั่งของการจราจร รวมถึงเขม่าควัน และฝุ่นผงจากการก่อสร้าง แต่ปัญหาวิกฤติที่ชาวกรุงเทพมหานคร กำลังวิตกคือ ฝุ่นละอองขนาดเล็กกว่า 2.5 ไมครอน (Particulate Matter 2.5 – PM2.5) ซึ่งไม่สามารถมองเห็นได้ด้วยตาเปล่า ทำให้ขาดความตระหนักถึงอันตรายต่อสุขภาพอย่างใหญ่หลวง

ฝุ่นละอองจิ๋ว PM 2.5 ส่วนใหญ่เกิดจากการเผาไหม้เครื่องจักร โดยเฉพาะเครื่องยนต์ของทั้งรถยนต์ใหม่และเก่า มักมีปริมาณสูงสุดช่วงรถติดมากๆ ในช่วงเช้าและเย็นของวันทำงาน โดยมากจะเกิดในช่วงฤดูหนาวที่อากาศนิ่งและแห้ง ส่งผลให้ฝุ่นไม่ลอยขึ้นที่สูง อีกทั้งปัจจุบันกรุงเทพฯ กำลังประสบปัญหาหมอกควันได้ยาก อากาศหยุดนิ่ง เนื่องจากมีตึกสูงปิดกั้นทางลมรวมถึงฝุ่นจากการก่อสร้างที่มีอยู่แทบทุกพื้นที่ มาเป็นปัจจัยเกื้อหนุนทำให้คุณภาพอากาศเลวร้ายลง คณะผู้จัดทำจึงได้นำข้อมูลฝุ่นละอองขนาดเล็ก (PM 2.5) ย้อนหลัง 10 ปี เพื่อหาหาจำนวนวันและช่วงเวลา เพื่อใช้ประกอบการตัดสินใจออกนโยบายแก้ไขปัญหาดังต่อไป

ข้อมูลที่ใช้ในการทำงาน

ใช้ข้อมูลจากจุดตรวจวัดคุณภาพอากาศบริเวณริมถนนดินแดง เขตดินแดง จังหวัดกรุงเทพมหานคร โดยเป็นข้อมูลการสะสมของปริมาณฝุ่นละอองขนาดเล็ก (PM 2.5) รายชั่วโมง ย้อนหลังไป 10 ปี ตั้งแต่ปี พ.ศ. 2554-2563

วิธีการดำเนินงาน

การนำข้อมูลฝุ่นละอองขนาดเล็ก (PM 2.5) มาดำเนินการวิเคราะห์ จะใช้แนวคิดทาง Data science โดยใช้หลักการ CRISP-DM ดังนี้

1. Business Understanding

ขั้นตอนแรกมุ่งไปที่การทำความเข้าใจธุรกิจ ปัญหาและวัตถุประสงค์ของโครงการจากมุมมองทางธุรกิจ จากนั้นแปลงปัญหาให้อยู่ในรูปของโจทย์สำหรับการวิเคราะห์ข้อมูล และวางแผนการดำเนินงานเบื้องต้น โดยต้องเข้าใจความสัมพันธ์ของช่วงเวลาที่ปริมาณฝุ่นละอองขนาดเล็ก (PM 2.5) เกินค่ามาตรฐาน (50 ไมโครกรัม/ลูกบาศก์เมตร) และสามารถใช้ผลลัพธ์ของการทำ Data science ในการออกนโยบายเพื่อแก้ไขปัญหาได้

2. Data Understanding

ขั้นตอนนี้เริ่มต้นด้วยการรวบรวมข้อมูล จากนั้นทำความเข้าใจ ตรวจสอบคุณภาพ และเลือกข้อมูลที่เกี่ยวข้องรวบรวมมาว่าจะใช้ข้อมูลใดบ้างในการวิเคราะห์ โดยใช้ข้อมูลปริมาณการสะสมของฝุ่นละอองขนาดเล็ก (PM 2.5) ย้อนหลัง 10 ปี โดยลักษณะข้อมูลที่ได้รับจากหน่วยงานจะเป็นข้อมูลการสะสมของปริมาณฝุ่นละอองขนาดเล็ก (PM 2.5) รายชั่วโมง โดยข้อมูลที่ได้รับ มี Format ที่ไม่เป็นมาตรฐาน ต้องทำการปรับปรุงข้อมูลให้อยู่ในรูปแบบที่สามารถใช้งานได้ การวิเคราะห์ สามารถใช้การทำงานในโมเดล Time series ได้

3. Data Preparation

ขั้นตอนการเตรียมข้อมูล หมายถึง ขั้นตอนทั้งหมดที่จะทำเพื่อให้ข้อมูลดิบที่เรารวบรวมมา กลายเป็นข้อมูลสมบูรณ์ที่พร้อมจะเข้าสู่โมเดลในขั้นตอนที่ 4 เช่น การสร้างตาราง การลบข้อมูลที่ไม่ต้องการออก การแปลงข้อมูลให้อยู่ในรูปแบบที่ต้องการ เป็นต้น โดยเลือกใช้ข้อมูลจากจุดตรวจวัด ริมถนนดินแดง เขตดินแดง กรุงเทพมหานคร เพื่อใช้เป็นตัวแทนของข้อมูลการสะสมของปริมาณฝุ่นละอองขนาดเล็ก (PM 2.5) ในจังหวัดกรุงเทพมหานคร และปรับข้อมูลให้อยู่ในรูปแบบที่ต้องการ ใช้เครื่องมือในภาษา Python เช่น pandas, numpy และรันในโปรแกรม Google colabory ในการปรับปรุงข้อมูล โดยมีรายละเอียดดังนี้

ขั้นตอนการ Clean data

- เรียกชุดข้อมูลด้วยคำสั่ง `pd.read_csv` โดยจะเรียกจากทั้ง 2 sheet แยกกัน

```
1 import pandas as pd
```

```
1 df1 = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Data Science/project/(54t)การเคหะดินแดง_แก่วันที่.xlsx', sheet_name='11-16')
2 df2 = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Data Science/project/(54t)การเคหะดินแดง_แก่วันที่.xlsx', sheet_name='17-21')
```

- จากนั้นนำข้อมูลทั้ง 2 ชุดรวมกันด้วยคำสั่ง `pd.concat`

```
1 df = pd.concat([df1,df2])
2 df
```

	วัน/เดือน/ปี	ชั่วโมง	PM2.5 (มคก./ลบ.ม.)
0	110501	100	NaN
1	110501	200	NaN
2	110501	300	NaN
3	110501	400	NaN
4	110501	500	NaN
...
36475	210228	2000	66
36476	210228	2100	61
36477	210228	2200	-
36478	210228	2300	57
36479	210228	2400	49

86208 rows × 3 columns

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 86208 entries, 0 to 36479
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   วัน/เดือน/ปี           86208 non-null int64
1   ชั่วโมง               86208 non-null int64
2   PM2.5 (มคก./ลบ.ม.)   60430 non-null object
dtypes: int64(2), object(1)
memory usage: 2.6+ MB
```

- ตรวจสอบข้อมูลพบว่ามีค่า null จึงกำจัดด้วยการลบข้อมูลนั้นออกด้วย `df.dropna()` จะได้ข้อมูลจำนวน 60,430 ข้อมูล

```
1 df = df.dropna()
2 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60430 entries, 369 to 36479
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   วัน/เดือน/ปี           60430 non-null int64
1   ชั่วโมง               60430 non-null int64
2   PM2.5 (มคก./ลบ.ม.)   60430 non-null object
dtypes: int64(2), object(1)
memory usage: 1.8+ MB
```

- ทำการบันทึกข้อมูลเป็นไฟล์ csv แล้วเรียกใช้อีกครั้ง เพื่อทำการกำหนดคอลัมน์ “วัน/เดือน/ปี” ให้เป็นชนิด datetime โดยจะต้องทำการกำหนดรูปแบบเพื่อให้สามารถแปลงวันที่ได้ถูกต้อง ด้วยการใช้ datetime.strptime()
- strftime ใช้สำหรับแปลงข้อมูลชนิด str เป็น datetime ซึ่งจะต้องกำหนด format โดย %y คือ เลขลงท้ายของปีค.ศ. เช่น 00, 01,...99 %m คือ เดือนที่เป็นเลขฐานสิบ เช่น 01, 02....12 และ %d คือ วันที่ ที่เป็นเลขฐานสิบ เช่น 01, 02, 03,.....31

```
1 from datetime import datetime
2
3 custom_date_parser = lambda x: datetime.strptime(x, "%y%m%d")
4
5 df3 = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Data Science/all_data.csv',
6                 parse_dates=['วัน/เดือน/ปี'],
7                 date_parser=custom_date_parser)
8 df3
```

	วัน/เดือน/ปี	ชั่วโมง	PM2.5 (มคก./ลบ.ม.)
0	2011-05-16	1000	29.0
1	2011-05-16	1100	2.0
2	2011-05-16	1200	100.0
3	2011-05-16	1300	45.0
4	2011-05-16	1400	50.0

```
1 df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60430 entries, 0 to 60429
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   วัน/เดือน/ปี            60430 non-null  datetime64[ns]
1   ชั่วโมง                60430 non-null  int64
2   PM2.5 (มคก./ลบ.ม.)    60430 non-null  object
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 1.4+ MB
```

- ทำการแปลงเวลาในคอลัมน์ชั่วโมงให้อยู่ในรูปแบบ %H%M เช่น 10:00, 01:00 เป็นต้น โดยจะต้องเปลี่ยนชั่วโมง 2400, 100 และ 2400 เป็น 0000, 0100 และ 0200 ด้วยคำสั่ง .replace() ก่อน เนื่องจาก pd.to_datetime จะมีรูปแบบเวลาเป็น 00:00 -23:00 นาฬิกา

```
1 df3['ชั่วโมง'] = df3['ชั่วโมง'].astype(str)
```

```
1 df3['ชั่วโมง'] = df3['ชั่วโมง'].replace(['2400', '100', '200'], ['0000', '0100', '0200'])
```

```
1 df3[0:20]
```

```
1 tt=[]
2
3 for i in range(len(df3)):
4     t = [df3['ชั่วโมง'][i]]
5     x = pd.to_datetime(t, format="%H%M")
6     y = x.strftime("%H:%M").tolist()
7     tt.append(y)
8
```

```
1 df3['time'] = pd.DataFrame(tt)
```

```
1 df3
```

	วัน/เดือน/ปี	ชั่วโมง	PM2.5 (มคก./ลบ.ม.)	time
0	2011-05-16	1000	29.0	10:00
1	2011-05-16	1100	2.0	11:00
2	2011-05-16	1200	100.0	12:00
3	2011-05-16	1300	45.0	13:00
4	2011-05-16	1400	50.0	14:00
...
60425	2021-02-28	2000	66	20:00
60426	2021-02-28	2100	61	21:00
60427	2021-02-28	2200	-	22:00
60428	2021-02-28	2300	57	23:00
60429	2021-02-28	0000	49	00:00

```
60430 rows x 4 columns
```

- สร้างคอลัมน์ Timestamp และกำหนดให้เป็นชนิด datetime

```
1 df3['Timestamp'] = df3['วัน/เดือน/ปี'].astype(str) + ' ' + df3['time']
2 df3.head()
```

	วัน/เดือน/ปี	ชั่วโมง	PM2.5 (มคก./ลบ.ม.)	time	Timestamp
0	2011-05-16	1000	29.0	10:00	2011-05-16 10:00
1	2011-05-16	1100	2.0	11:00	2011-05-16 11:00
2	2011-05-16	1200	100.0	12:00	2011-05-16 12:00
3	2011-05-16	1300	45.0	13:00	2011-05-16 13:00
4	2011-05-16	1400	50.0	14:00	2011-05-16 14:00

```
1 df3['Timestamp'] = pd.to_datetime(df3['Timestamp'])
```

```
1 df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60430 entries, 0 to 60429
Data columns (total 5 columns):
#   Column              Non-Null Count  Dtype
---  -
0   วัน/เดือน/ปี         60430 non-null  datetime64[ns]
1   ชั่วโมง             60430 non-null  object
2   PM2.5 (มคก./ลบ.ม.) 60430 non-null  object
3   time                60430 non-null  object
4   Timestamp           60430 non-null  datetime64[ns]
dtypes: datetime64[ns](2), object(3)
memory usage: 2.3+ MB
```

- ทำการจัดรูปข้อมูลให้เรียกใช้งานได้ง่ายขึ้นต่อการวิเคราะห์ข้อมูล

```
3 df3['Year'] = [year.strftime("%Y") for year in df3['Timestamp']]
4 df3['Month'] = [month.strftime("%Y-%m") for month in df3['Timestamp']]
5 df3['Date'] = [date.strftime("%Y-%m-%d") for date in df3['Timestamp']]
6 df3['Time'] = [time.strftime("%H:%M") for time in df3['Timestamp']]
7
8 df3.drop(['วัน/เดือน/ปี', 'ชั่วโมง', 'time'], axis=1, inplace=True)
9
10 df3[0:10]
```

	PM2.5 (มคก./ลบ.ม.)	Timestamp	Year	Month	Date	Time
0	29.0	2011-05-16 10:00:00	2011	2011-05	2011-05-16	10:00
1	2.0	2011-05-16 11:00:00	2011	2011-05	2011-05-16	11:00
2	100.0	2011-05-16 12:00:00	2011	2011-05	2011-05-16	12:00
3	45.0	2011-05-16 13:00:00	2011	2011-05	2011-05-16	13:00
4	50.0	2011-05-16 14:00:00	2011	2011-05	2011-05-16	14:00
5	23.0	2011-05-16 15:00:00	2011	2011-05	2011-05-16	15:00
6	21.0	2011-05-16 16:00:00	2011	2011-05	2011-05-16	16:00
7	39.0	2011-05-16 17:00:00	2011	2011-05	2011-05-16	17:00
8	36.0	2011-05-16 18:00:00	2011	2011-05	2011-05-16	18:00
9	15.0	2011-05-16 19:00:00	2011	2011-05	2011-05-16	19:00

- กำจัดข้อมูลปี 2021 และ ข้อมูลที่ใส่เครื่องหมาย“-” ออก รวมถึงเปลี่ยนข้อมูล PM2.5 ให้เป็นชนิด float

```
1 df3.drop(df3[df3['Year'] == '2021'].index, inplace=True)
2 df3.tail()
```

	PM2.5 (มคก./ลบ.ม.)	Timestamp	Year	Month	Date	Time
59009	20	2020-12-31 20:00:00	2020	2020-12	2020-12-31	20:00
59010	26	2020-12-31 21:00:00	2020	2020-12	2020-12-31	21:00
59011	23	2020-12-31 22:00:00	2020	2020-12	2020-12-31	22:00
59012	21	2020-12-31 23:00:00	2020	2020-12	2020-12-31	23:00
59013	16	2020-12-31 00:00:00	2020	2020-12	2020-12-31	00:00

```
1 df3[df3['PM2.5 (มคก./ลบ.ม.)'] == '-']
```

	PM2.5 (มคก./ลบ.ม.)	Timestamp	Year	Month	Date	Time
38875	-	2018-09-14 04:00:00	2018	2018-09	2018-09-14	04:00
38876	-	2018-09-14 05:00:00	2018	2018-09	2018-09-14	05:00
38877	-	2018-09-14 06:00:00	2018	2018-09	2018-09-14	06:00
38878	-	2018-09-14 07:00:00	2018	2018-09	2018-09-14	07:00
38879	-	2018-09-14 08:00:00	2018	2018-09	2018-09-14	08:00

```
1 df3.drop(df3[df3['PM2.5 (มคก./ลบ.ม.)'] == '-'].index, inplace=True)
2 df3[df3['PM2.5 (มคก./ลบ.ม.)'] == '-']
```

	PM2.5 (มคก./ลบ.ม.)	Timestamp	Year	Month	Date	Time
--	--------------------	-----------	------	-------	------	------

```
1 df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 58807 entries, 0 to 59013
Data columns (total 6 columns):
#   Column              Non-Null Count  Dtype
---  -
0   PM2.5 (มคก./ลบ.ม.)  58807 non-null object
1   Timestamp            58807 non-null datetime64[ns]
2   Year                 58807 non-null object
3   Month                58807 non-null object
4   Date                 58807 non-null object
5   Time                 58807 non-null object
dtypes: datetime64[ns](1), object(5)
memory usage: 3.1+ MB
```

```
1 df3['PM2.5 (มคก./ลบ.ม.)'] = df3['PM2.5 (มคก./ลบ.ม.)'].astype(float)
```

- หา PM2.5 ต่อวัน โดยการใช้คำสั่ง df3.groupby(by= 'Date')

```
1 PM_per_Day = df3.groupby(by='Date').agg(['min','max','mean','sum','count'])
2 PM_per_Day = PM_per_Day.drop(['Year'], axis=1)
3 PM_per_Day
```

	PM2.5 (มคก./ลบ.ม.)				
	min	max	mean	sum	count
Date					
2011-05-16	2.0	100.0	44.071429	617.0	14
2011-05-17	1.0	102.0	44.800000	896.0	20
2011-05-18	4.0	82.0	45.380952	953.0	21
2011-05-19	5.0	91.0	49.428571	1038.0	21
2011-05-20	4.0	92.0	43.476190	913.0	21
...
2020-12-27	31.0	84.0	51.666667	1240.0	24
2020-12-28	26.0	56.0	42.708333	1025.0	24
2020-12-29	23.0	63.0	45.041667	1081.0	24
2020-12-30	11.0	73.0	41.043478	944.0	23

- หาวันที่ค่าเฉลี่ย PM2.5 เกินค่ามาตรฐานจากชุดข้อมูล PM_per_Day[PM_per_Day[('PM2.5 (มคก./ลบ.ม.)', 'mean')]]> 50]

```
1 Over_Std_day = PM_per_Day[PM_per_Day[('PM2.5 (มคก./ลบ.ม.)', 'mean')]]> 50]
2 Over_Std_day
```

Date	PM2.5 (มคก./ลบ.ม.)				
	min	max	mean	sum	count
2011-10-19	45.0	83.0	57.041667	1369.0	24
2011-10-20	48.0	86.0	64.125000	1539.0	24
2011-10-21	44.0	80.0	56.083333	1346.0	24
2011-10-24	46.0	71.0	57.541667	1381.0	24
2011-11-03	29.0	68.0	50.125000	1203.0	24
...
2020-12-16	24.0	104.0	62.208333	1493.0	24
2020-12-17	27.0	152.0	57.791667	1387.0	24
2020-12-23	44.0	62.0	53.166667	1276.0	24
2020-12-24	39.0	67.0	56.000000	1344.0	24
2020-12-27	31.0	84.0	51.666667	1240.0	24

- หาช่วงเวลาที่มีค่า PM2.5 เกินค่ามาตรฐาน โดยใช้ df3[df3['PM2.5 (มคก./ลบ.ม.)']> 50]

```
1 #ช่วงเวลาที่มีค่าPM2.5 เกินค่ามาตรฐาน
2 Over_Std = df3[df3['PM2.5 (มคก./ลบ.ม.)']> 50]
3 Over_Std
```

	PM2.5 (มคก./ลบ.ม.)	Timestamp	Year	Month	Date	Time
2	100.0	2011-05-16 12:00:00	2011	2011-05	2011-05-16	12:00
11	77.0	2011-05-16 21:00:00	2011	2011-05	2011-05-16	21:00
12	78.0	2011-05-16 22:00:00	2011	2011-05	2011-05-16	22:00
13	52.0	2011-05-16 00:00:00	2011	2011-05	2011-05-16	00:00
14	78.0	2011-05-17 01:00:00	2011	2011-05	2011-05-17	01:00
...
58765	63.0	2020-12-30 06:00:00	2020	2020-12	2020-12-30	06:00
58766	68.0	2020-12-30 07:00:00	2020	2020-12	2020-12-30	07:00
58767	68.0	2020-12-30 08:00:00	2020	2020-12	2020-12-30	08:00
58768	73.0	2020-12-30 09:00:00	2020	2020-12	2020-12-30	09:00
58769	55.0	2020-12-30 10:00:00	2020	2020-12	2020-12-30	10:00

10745 rows × 6 columns

-หาจำนวนครั้งที่มีความเกินมาตรฐานในแต่ละชั่วโมง โดยใช้ Over_std.groupby(['Time'])

```
1 # จำนวนครั้งที่มีความเกินมาตรฐานในแต่ละชั่วโมง
2 data_Over_Std_Time = Over_Std.groupby(['Time'])['PM2.5 (มคก./ลบ.ม.)'].agg(['min','max','mean','sum','count'])
3 data_Over_Std_Time
```

	min	max	mean	sum	count
Time					
00:00	50.18	151.65	69.067374	31563.79	457
01:00	50.18	163.29	67.805356	27596.78	407
02:00	50.43	152.00	67.890197	24168.91	356
03:00	50.18	151.00	67.501360	22342.95	331
04:00	50.18	133.00	68.584858	21741.40	317
05:00	50.18	148.54	68.175586	22088.89	324
06:00	50.43	154.00	68.948438	25166.18	365
07:00	50.18	156.00	69.702120	30250.72	434
08:00	50.18	196.69	70.105741	37366.36	533
09:00	50.18	200.05	69.357143	47579.00	686
10:00	50.18	202.64	69.101713	49615.03	718

- หาจำนวนครั้งที่มีความเกินมาตรฐานต่อวัน โดยใช้ Over_std.groupby(['Date'])

```
1 # จำนวนครั้งที่มีความเกินมาตรฐานต่อวัน
2 data_Over_Std = Over_Std.groupby(['Date'])['PM2.5 (มคก./ลบ.ม.)'].agg(['min','max','mean','sum','count'])
3 data_Over_Std
```

	min	max	mean	sum	count
Date					
2011-05-16	52.0	100.0	76.750000	307.0	4
2011-05-17	52.0	102.0	76.000000	608.0	8
2011-05-18	53.0	82.0	69.125000	553.0	8
2011-05-19	55.0	91.0	72.777778	655.0	9
2011-05-20	53.0	92.0	67.444444	607.0	9
...
2020-12-26	54.0	72.0	64.000000	320.0	5
2020-12-27	51.0	84.0	60.538462	787.0	13
2020-12-28	52.0	56.0	54.285714	380.0	7
2020-12-29	51.0	63.0	56.875000	455.0	8
2020-12-30	55.0	73.0	62.400000	624.0	10

1394 rows × 5 columns

- หาจำนวนครั้งที่มียค่าเกินมาตรฐานต่อเดือน โดยใช้ Over_std.groupby(['Month'])

```
1 Over_Std_Month = Over_Std.groupby(['Month'])['PM2.5 (มคก./ลบ.ม.)'].agg(['min','max','mean','sum','count'])
2 Over_Std_Month
```

	min	max	mean	sum	count
Month					
2011-05	52.0	102.0	68.355932	4033.0	59
2011-06	52.0	93.0	62.529412	2126.0	34
2011-07	51.0	82.0	59.727273	1971.0	33
2011-08	51.0	88.0	60.211538	3131.0	52
2011-09	51.0	91.0	60.728571	4251.0	70
...
2020-06	51.0	53.0	52.000000	104.0	2
2020-09	53.0	59.0	55.800000	279.0	5
2020-10	51.0	66.0	55.416667	1330.0	24
2020-11	51.0	95.0	58.186275	5935.0	102
2020-12	51.0	160.0	72.617886	17864.0	246

86 rows × 5 columns

4. Modeling

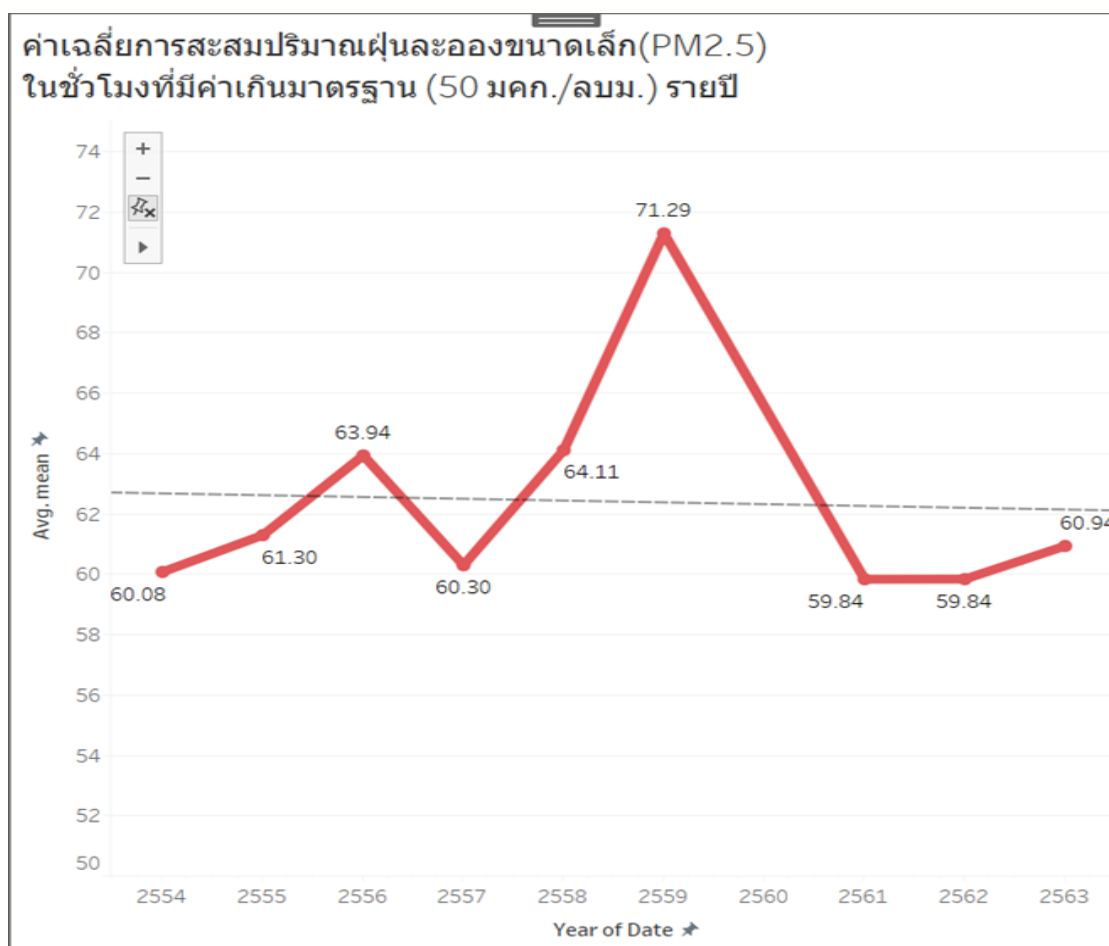
ในขั้นตอนนี้เราจะเลือกและทดสอบสร้างโมเดลหลายๆแบบที่น่าจะสามารถแก้ไขปัญหที่ต้องการได้ จากนั้นค่อยๆปรับค่าพารามิเตอร์ในแต่ละโมเดล เพื่อให้ได้โมเดลที่เหมาะสมที่สุดมาใช้ในการแก้ไขปัญา โดยทำการ Visualization ข้อมูล ด้วยโปรแกรม Tableau โดยข้อมูลจะออกมาในรูปแบบกราฟเปรียบเทียบทางสถิติ ดังนี้

- ค่าเฉลี่ยการสะสมปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) ในช่วงที่มีค่าเกินมาตรฐาน (50 ไมโครกรัม/ลูกบาศก์เมตร)
- จำนวนวันที่มีค่าเฉลี่ยการสะสมของปริมาณฝุ่นละอองขนาดเล็ก เฉลี่ย 24 ชั่วโมง เกินค่ามาตรฐาน (50 ไมโครกรัม/ลูกบาศก์เมตร)
- ช่วงเวลาที่มีค่าการสะสมของปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) เกินค่ามาตรฐาน (50 ไมโครกรัม/ลูกบาศก์เมตร) รายชั่วโมง ตั้งแต่ปี 2011-2020

(ดูรายละเอียดเพิ่มเติมได้ที่ >> <https://public.tableau.com/profile/suthin.junphet#!/>)

สรุปผลการดำเนินงาน

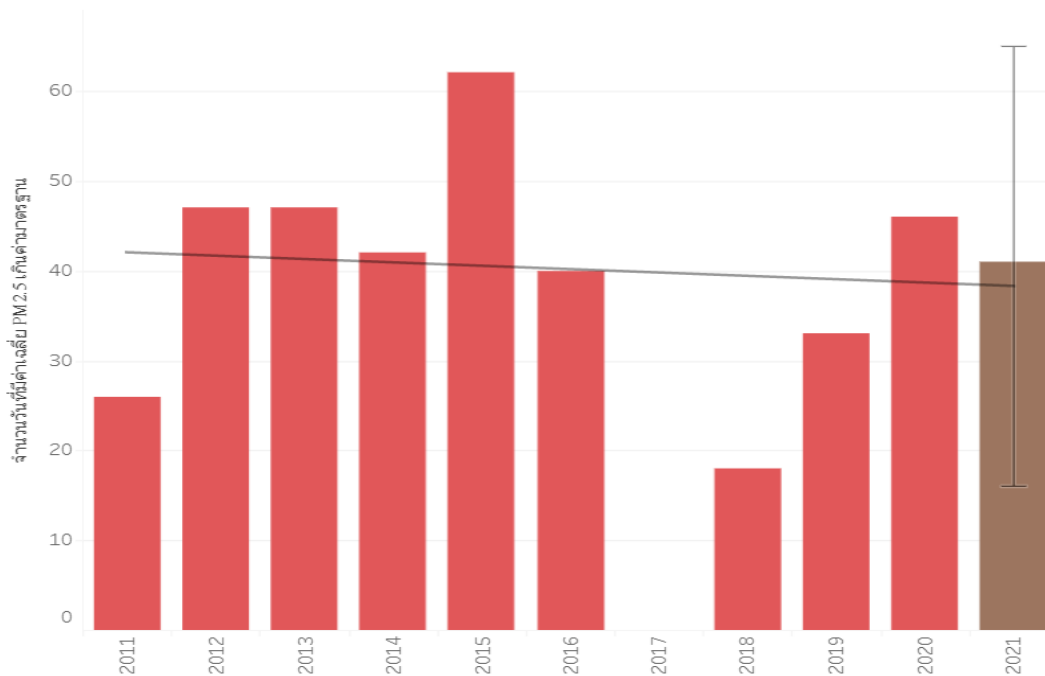
1. ค่าเฉลี่ยการสะสมปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) ในชั่วโมงที่มีค่าเกินมาตรฐาน (50 ไมโครกรัม/ลูกบาศก์เมตร)



- นำค่าการสะสมปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) ที่เกินมาตรฐานรายชั่วโมงมาทำการหาค่าเฉลี่ยจากระยะเวลา 10 ปี พบว่าปีที่มีค่าเฉลี่ยของการสะสมปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) เกินค่ามาตรฐานสูงสุด คือ ปี พ.ศ. 2559 โดยค่าเฉลี่ยอยู่ที่ 71 ไมโครกรัม/ลูกบาศก์เมตร และเมื่อดูจากเส้นแนวโน้มพบว่าค่าเฉลี่ยของรายชั่วโมงที่มีค่าเกินมาตรฐานมีแนวโน้มลดลงเล็กน้อย

2. จำนวนวันที่มีค่าเฉลี่ยการสะสมของปริมาณฝุ่นละอองขนาดเล็ก เฉลี่ย 24 ชั่วโมง เกินค่ามาตรฐาน (50 ไมโครกรัม/ลูกบาศก์เมตร)

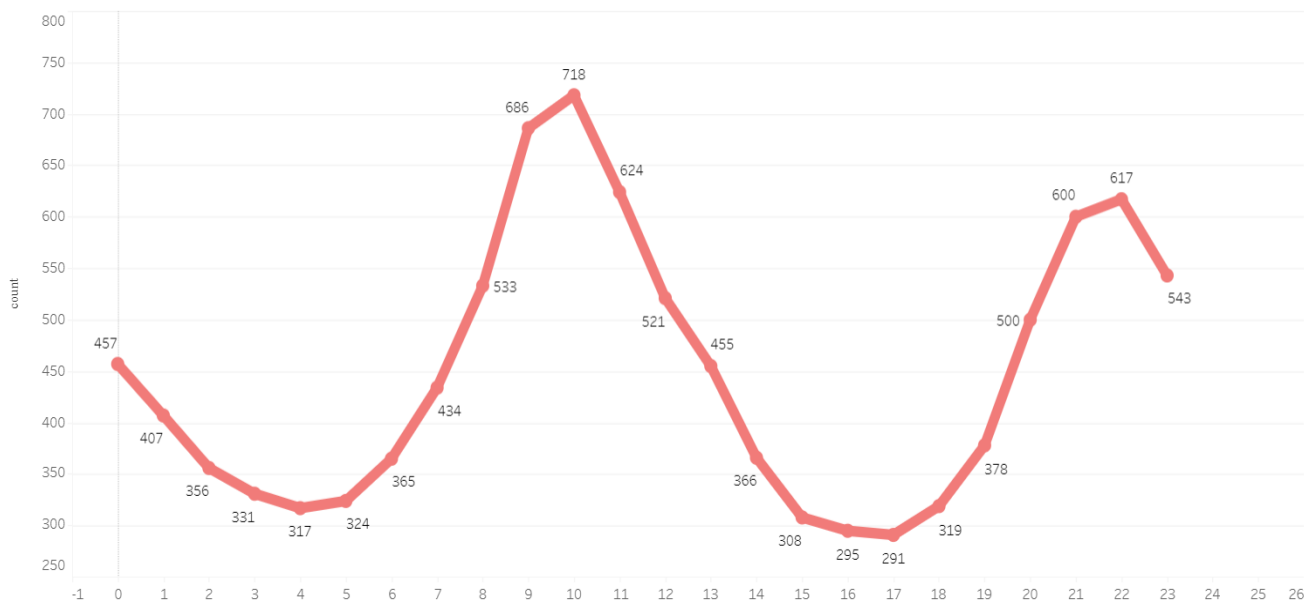
จำนวนวันที่มีค่าเฉลี่ยการสะสมของปริมาณฝุ่นละอองขนาดเล็ก เฉลี่ย 24 ชั่วโมง เกินค่ามาตรฐาน 50 (มคก./ลบม.)รายปี



- ปีที่มีจำนวนวันที่ค่าเฉลี่ยการสะสมของปริมาณฝุ่นละอองขนาดเล็ก เฉลี่ย 24 ชั่วโมง เกินค่ามาตรฐานมากที่สุด คือปี 2015 เป็นจำนวน 62 วัน
- เดือนที่มีจำนวนวันที่ค่าเฉลี่ยการสะสมของปริมาณฝุ่นละอองขนาดเล็ก เฉลี่ย 24 ชั่วโมง เกินค่ามาตรฐาน ได้แก่ เดือนมกราคม, กุมภาพันธ์, มีนาคม, ตุลาคม, พฤศจิกายน และธันวาคมของทุกปี
- จากกราฟพบว่าแนวโน้มของจำนวนวันที่ค่าเฉลี่ยการสะสมของปริมาณฝุ่นละอองขนาดเล็ก เฉลี่ย 24 ชั่วโมง เกินค่ามาตรฐาน มีแนวโน้มลดลงเล็กน้อย

3. ช่วงเวลาที่มีค่าการสะสมของปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) เกินค่ามาตรฐาน (PM2.5) รายชั่วโมง ตั้งแต่ปี 2011-2020

ช่วงเวลาที่มีค่าการสะสมของปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) เกินค่ามาตรฐาน (PM2.5) รายชั่วโมง ตั้งแต่ปี 2011-2020



- ช่วงเวลา 10.00 น. ของแต่ละวัน จะมีค่าการสะสมของปริมาณฝุ่นละอองขนาดเล็ก (PM2.5) เกินค่ามาตรฐาน มากที่สุดอยู่ที่ 718 ครั้ง/10ปี
- ช่วงเวลาที่ควรให้ความสนใจเพื่อหาที่มาของแหล่งกำเนิด คือช่วงเวลา 08.00-11.00น. และช่วงเวลา 20.00-24.00น. ซึ่งเป็นช่วงเวลาที่เป็นชั่วโมงเร่งด่วน มีการจราจรหนาแน่น การออกนโยบายควรเกี่ยวกับการจำกัดปริมาณรถที่วิ่งเข้าสู่แยกดินแดงในชั่วโมงเร่งด่วน เป็นต้น

รายละเอียดเกี่ยวกับกราฟข้อมูลสามารถดูเพิ่มเติมได้ที่

https://public.tableau.com/profile/suthin.junphet#!/vizhome/PM2_550___/Dashboard1

<https://public.tableau.com/profile/suthin.junphet#!/vizhome/AVGCNTmonth/Dashboard1>

https://public.tableau.com/profile/suthin.junphet#!/vizhome/Freq_Hour/Freq_hour

บรรณานุกรม

Thapanee Boonchob. (2020). เข้าใจ CRISP-DM ฉบับเร่งรัด. ค้นเมื่อ 27 มีนาคม 2563, จาก

<https://kamboonchob.medium.com>

Samitivejchinateown. (2021). PM 2.5 ฝุ่นเล็กจิ๋ว แต่ส่งผลกระทบต่อสุขภาพมหาศาล. ค้นเมื่อ 27 มีนาคม 2563, จาก

<https://www.samitivejchinateown.com/th/health-article/PM-25-Effects-Your-Health>