Data science Final Project: PM2.5

รายชื่อกลุ่ม

1. นาย วีรวงศ์ ศรีรัตนสมบุญ รหัสนิสิต 6130639821
2. นาย เสรี เยี่ยงสกุลไพศาล รหัสนิสิต 6130640321
3. นาย ชนภัทร กีรติเชาวนากุล รหัสนิสิต 6131006721

Dataset

ปี/เดือน/วัน	ชั่วโมง	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	Station	index
191201	100	16	4	0.4	89	22.7	96	78t	1.354545
191201	200	13	4	0.4	116	22.6	96	78t	1.35
191201	300	16	3	0.3	69	22.5	96	78t	1.366667
191201	400	13	3	0.5	103	22.5	96	78t	1.225
191201	500	11	2	0.3	129	22.5	96	78t	1.883333

Train dataset

- ข้อมูล PM10, PM2.5, Wind speed, Wind direction, Temperature, Humidity ได้มาจาก dataset PCD Data before-2020-09
- ข้อมูล index(ดัชนีรถติด) ได้มาจาก https://traffic.longdo.com/download

Test dataset

- ข้อมูล PM10, PM2.5, Wind speed, Wind direction, Temperature, Humidity ได้มาจากการ scrape ข้อมูลบน http://air4thai.pcd.go.th/ โดยมีข้อมูลตั้งแต่วันที่ 01/03/2022 ถึงปัจจุบัน
- ข้อมูล index ได้มาจาก https://traffic.longdo.com/download

Data Scraping

ใช้ในการ scrape ข้อมูลใส่ test dataset และเป็นข้อมูล input ตอนรัน airflow pipeline

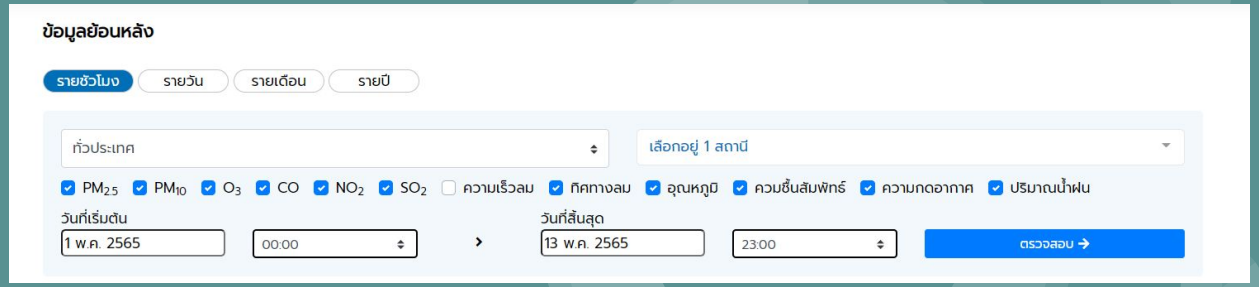
Source:

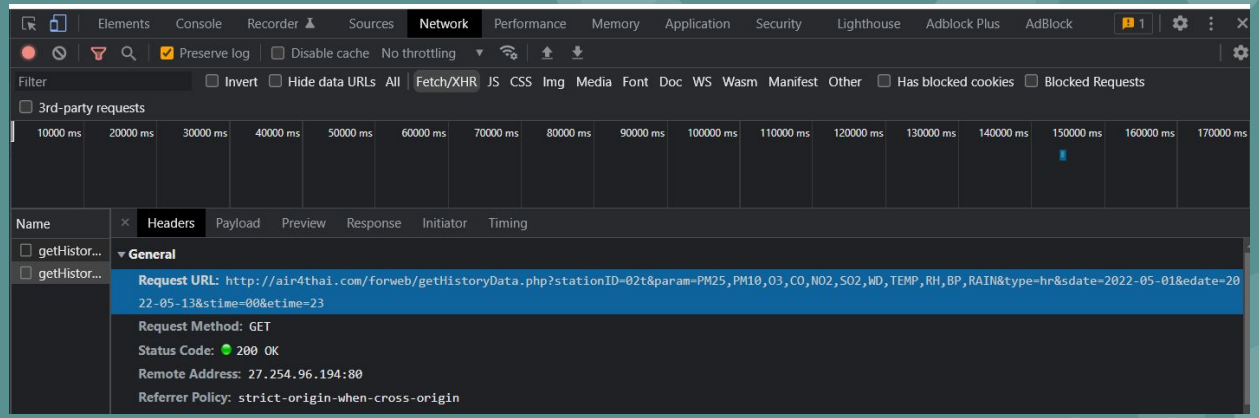
1. http://air4thai.pcd.go.th/
2. https://traffic.longdo.com/

Air4Thai (1)

Scraped data

1. Datetime
2. Latitude, Longitude
3. Temperature
4. PM10
5. Pm2.5
6. Wind speed
7. Wind dir
8. Humidity





ตัวอย่าง request

http://air4thai.com/forweb/getHistoryData.php?stationID=02t&param=PM25,PM10,WD,TEMP,RH,BP&type=hr&sdate=2022-05-01&edate=2022-05-13&stime=00&etime=00

Air4Thai (2)

```
data_info = requests.get(api_url)
info = json.loads(data_info.text)
```

ตัวอย่าง response

{"result":"OK","error":"","stations":[{"stationID":"02t","params":["WS","WD","TEMP","RH","PM10","PM25"],

"data":[{"DATETIMEDATA":"2022-05-19
20:00:00","WS":0.3,"WD":44,"TEMP":30.8,"RH":63,"PM10":21,"PM25":13}],

"summary":{"WS":{"max":0.3,"min":0.3,"average":0.3,"totalcount":1,"count":1,"countPercentage":100},"WD":{"max":44,"min":44,"average":44,"totalcount":1,"count":1,"countPercentage":100},"TEMP":{"max":30.8,"min":30.8,"average":30.8,"totalcount":1,"count":1,"countPercentage":100},"RH":{"max":63,"min":63,"average":63,"totalcount":1,"count":1,"countPercentage":100},"PM10":{"max":21,"min":21,"average":21,"totalcount":1,"count":1,"countPercentage":100},"PM25":{"max":13,"min":13,"average":13,"totalcount":1,"count":1,"countPercentage":100}}}]}

Longdo Traffic

Scraped data: **ดัชนีรถติดเฉลี่ยทั้งประเทศ (index)**

Request: https://traffic.longdo.com/api/json/traffic/index?callback=callback_function

ตัวอย่าง response

```
callback_function({"index":2.5,"time":1652986200})
```

Dataset

ปี/เดือน/วัน	ชั่วโมง	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	Station	index
191201	100	16	4	0.4	89	22.7	96	78t	1.354545
191201	200	13	4	0.4	116	22.6	96	78t	1.35
191201	300	16	3	0.3	69	22.5	96	78t	1.366667
191201	400	13	3	0.5	103	22.5	96	78t	1.225
191201	500	11	2	0.3	129	22.5	96	78t	1.883333
191201	600	9	3	0.5	65	22.4	97	78t	2.075
191201	700	8	2	0.6	263	22.4	97	78t	2.5
191201	800	9	3	0.2	73	22.4	97	78t	3.016667
191201	900	10	4	0.4	50	22.5	97	78t	3.808333
191201	1000	8	4	0.6	63	22.7	97	78t	4.366667
191201	1100	9	3	0.6	1	22.9	96	78t	4.85
191201	1200	9	3	0.6	68	23.4	96	78t	4.991667
191201	1300	8	2	0.2	87	23.4	96	78t	5.033333
191201	1400	7	2	0.4	59	23.6	95	78t	4.966667
191201	1500	8	3	0.9	47	23.5	95	78t	4.758333
191201	1600	8	3	0.5	58	23.4	95	78t	5.183333
191201	1700	9	2	0.5	104	23.3	95	78t	5.608333
191201	1800	8	2	0.5	48	23	95	78t	5.983333
191201	1900	10	3	0.2	165	22.6	96	78t	5.125
191201	2000	10	4	0.4	51	22.4	96	78t	4.533333
191201	2100	9	2	0.4	56	22.2	96	78t	3.991667
191201	2200	10	3	0.4	53	22.1	96	78t	2.316667

Train:
ข้อมูลก่อนปี 2020/09

Test:
ข้อมูลปี 2022/03 ถึงปัจจุบัน

PM10, Wind speed, Wind dir, Temp, Rel hum, index ที่มีค่าเป็น NaN จะถูกแทนที่ด้วยค่า mean

Machine learning model

Sarimax Model

```
[ ] import pandas as pd
    import numpy as np
```

```
[ ] df = pd.read_csv('train_HJ.xlsx - Sheet1.csv')
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882: DtypeWarning: Columns (4) have mixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
[ ] df.head()
```

	Unnamed: 0	ปี/เดือน/วัน	ชั่วโมง	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	Station	index
0	0	191201	100	16.0	4	0.4	89.0	22.7	96.0	78t	1.354545
1	1	191201	200	13.0	4	0.4	116.0	22.6	96.0	78t	1.350000
2	2	191201	300	16.0	3	0.3	69.0	22.5	96.0	78t	1.366667
3	3	191201	400	13.0	3	0.5	103.0	22.5	96.0	78t	1.225000
4	4	191201	500	11.0	2	0.3	129.0	22.5	96.0	78t	1.883333

```
[ ] df = df.rename(columns={'ปี/เดือน/วัน': 'Date', 'ชั่วโมง': 'Hour'})
```

```
[ ] df.head()
```

	Unnamed: 0	Date	Hour	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	Station	index
0	0	191201	100	16.0	4	0.4	89.0	22.7	96.0	78t	1.354545
1	1	191201	200	13.0	4	0.4	116.0	22.6	96.0	78t	1.350000
2	2	191201	300	16.0	3	0.3	69.0	22.5	96.0	78t	1.366667
3	3	191201	400	13.0	3	0.5	103.0	22.5	96.0	78t	1.225000

อ่าน CSV และ เปลี่ยนชื่อ column

Sarimax Model

```
[ ] df['Date'] = df['Date'].astype(str)
    df['Date'] = df['Date'].str.pad(7, 'left', '0')
    df['Date'] = df['Date'].str.pad(8, 'left', '2')
```

```
[ ] df.head()
```

	Unnamed: 0	Date	Hour	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	Station	index
0	0	20191201	100	16.0	4	0.4	89.0	22.7	96.0	78t	1.354545
1	1	20191201	200	13.0	4	0.4	116.0	22.6	96.0	78t	1.350000
2	2	20191201	300	16.0	3	0.3	69.0	22.5	96.0	78t	1.366667
3	3	20191201	400	13.0	3	0.5	103.0	22.5	96.0	78t	1.225000
4	4	20191201	500	11.0	2	0.3	129.0	22.5	96.0	78t	1.883333

```
[ ] #df['Date'].to_datetime()
    #pd.to_datetime(df['Date'])
    df['Date'] = pd.to_datetime(df['Date'], format='%Y%m%d', errors='coerce')
```

เปลี่ยน Date Time format

Sarimax Model

```
[ ]  df = df.interpolate('ffill')
```

```
[ ]  df = df.drop('Unnamed: 0', axis = 1)
```

```
[ ]  #df['PM2.5'] = 'InVld'
     df.loc[df['PM2.5'] == 'InVld']
```

	Date	Hour	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	Station	index
24708	2018-10-24	NaT	12.0	InVld	1.6	300.0	27.9	73.0	72t	4.925000
24721	2018-10-25	NaT	24.0	InVld	0.5	337.0	25.0	88.0	72t	1.141667

```
[ ]  #df.drop(df.loc[df['PM2.5'] == 'InVld'].index, inplace=True)
     #df.drop(df.loc[df['PM2.5'] == 'NoData'].index, inplace=True)
```

```
[ ]  df['PM2.5'].replace({'InVld': None},inplace =True)
     df['PM2.5'].replace({'NoData': None},inplace =True)
```

```
[ ]  df['PM2.5'] = df['PM2.5'].astype(float)
```

```
[ ]  df['Date'] = pd.DatetimeIndex(df['Date'])
```

Replace missing

Sarimax Model

```
[ ] df['PM2.5'] = df['PM2.5'].astype(float)
```

```
[ ] df['Date'] = pd.DatetimeIndex(df['Date'])
```

```
[ ] dfdate = df.groupby('Date').mean()
```

```
[ ] dfdate.head()
```

Date	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	index
2013-10-08	21.090909	16.909091	1.413636	142.954545	26.263636	62.0	2.979387
2013-10-09	56.208333	44.250000	0.729167	228.500000	27.833333	62.0	2.948057
2013-10-10	37.208333	28.625000	0.891667	235.208333	29.295833	62.0	2.813984
2013-10-11	50.500000	37.541667	0.737500	243.125000	28.900000	62.0	3.219949
2013-10-12	89.458333	62.833333	0.850000	194.250000	30.141667	62.0	1.746875

```
[ ] dfdate.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2539 entries, 2013-10-08 to 2020-09-30
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   PM10        2539 non-null   float64
 1   PM2.5       2539 non-null   float64
 2   Wind speed  2539 non-null   float64
 3   Wind dir    2539 non-null   float64
 4   Temp        2539 non-null   float64
```

```
[ ] dfdate = pd.read_csv('dfdate1.csv')
```

```
[ ] dfdate.head()
```

	Date	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	index	Unnamed: 8	Unnamed: 9
0	2013-10-08	21.090909	16.909091	1.413636	142.954545	26.263636	62.0	2.979387	08-10-13	True
1	2013-10-09	56.208333	44.250000	0.729167	228.500000	27.833333	62.0	2.948057	09-10-13	True
2	2013-10-10	37.208333	28.625000	0.891667	235.208333	29.295833	62.0	2.813984	10-10-13	True
3	2013-10-11	50.500000	37.541667	0.737500	243.125000	28.900000	62.0	3.219949	11-10-13	True
4	2013-10-12	89.458333	62.833333	0.850000	194.250000	30.141667	62.0	1.746875	12-10-13	True

```
[ ] #dfdate['Date'] = pd.to_datetime(dfdate['Date'], format='%Y%m%d', errors='coerce')
```

```
[ ] dfdate['Date'] = pd.to_datetime(dfdate['Date'])
    dfdate = dfdate.set_index('Date')
```

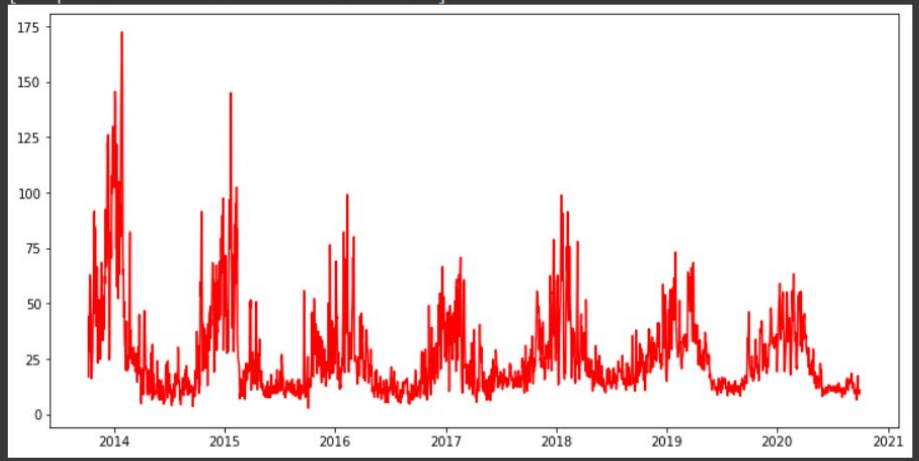
```
[ ] import matplotlib.pyplot as plt
```

เฉลี่ยข้อมูลแต่ละวัน และ save เพื่อ เพิ่มข้อมูลที่หาย ด้วย Excel

Sarimax Model

```
[ ]  plt.figure(figsize=(12,6))
     plt.plot(dfdate['PM2.5'],'r')
```

```
[<matplotlib.lines.Line2D at 0x7f2302ad8f10>]
```




```
]  train_size, valid_size, test_size = map(lambda r: int(r*dfdate.shape[0]), (0.6, 0.2, 0.2))

   train, valid, test = dfdate.iloc[:train_size], dfdate.iloc[train_size:train_size+valid_size ], dfdate.iloc[train_size+valid_size: ]
```

Plot Graph และแบ่ง Train/Test/Validate

Sarimax Model

```
[ ] train_size, valid_size, test_size = map(lambda r: int(r*dfdate.shape[0]), (0.6, 0.2, 0.2))
    train, valid, test = dfdate.iloc[:train_size], dfdate.iloc[train_size:train_size+valid_size ], dfdate.iloc[train_size+valid_size: ]
    #exog_columns = ['PM10',  'Wind speed', 'Wind dir', 'Temp', 'Rel hum',  'index']
    exog_columns = ['PM10', 'Wind speed', 'Wind dir', 'Temp', 'Rel hum', 'index']
```

```
[ ] from sklearn.metrics import mean_squared_error
    from statsmodels.tsa.statespace.sarimax import SARIMAX
    best_order = (0, 1, 1)
    best_seasonal_order = (1, 1, 0, 12)
```

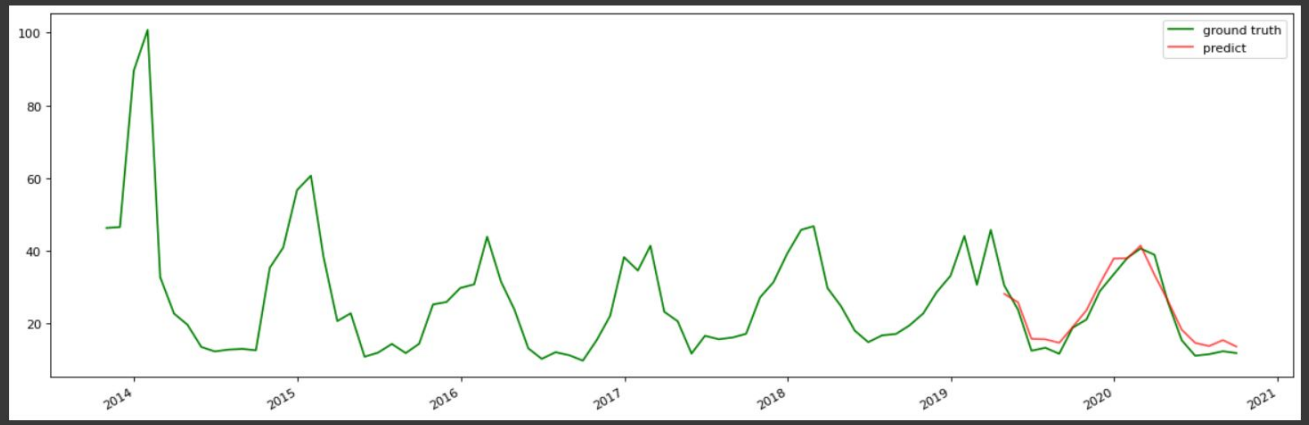
```
[ ] mod = SARIMAX(train['PM2.5'],
                  exog=train[exog_columns],
                  order=best_order,
                  seasonal_order=best_seasonal_order,
                  enforce_stationarity=False,
                  enforce_invertibility=False,
                  frequency = "D")

    results = mod.fit()

    print(results.summary().tables[1])
```

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
PM10           0.5447      0.067      8.140      0.000       0.414       0.676
Wind speed    -0.3207      0.612     -0.524      0.600      -1.521       0.879
Wind dir      -0.0137      0.026     -0.524      0.600      -0.065       0.038
Temp          -0.3605      1.521     -0.237      0.813      -3.341       2.620
Rel hum        0.4964      0.924      0.537      0.591      -1.314       2.307
index         -0.3896      1.465     -0.266      0.790      -3.262       2.482
ma.L1         -1.0032     10.879     -0.092      0.927     -22.325      20.318
ar.S.L12       0.0129      0.148      0.087      0.931      -0.278       0.304
```

Fit Model

Sarimax Model

```
test_exog = pd.concat((valid[exog_columns], test[exog_columns]), axis=0)
pred = results.get_prediction(start=test.index[0], end=test.index[-1], exog=test_exog, dynamic=False)
pred_ci = pred.conf_int()

from matplotlib.pyplot import figure
figure(num=None, figsize=(18, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(dfdate['PM2.5'], color='g',label='ground truth')
pred.predicted_mean.plot(alpha=.7, color='r',label='predict')
plt.legend(loc="upper right")

plt.show()
```



Prediction

Linear Regression Model

```
[ ] df['Temp40'] = df['Temp'] // 40
```

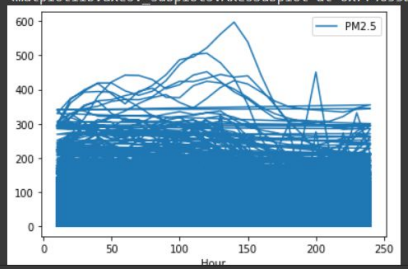
```
[ ] df['Hour'] = df['Hour'] / 10
```

```
[ ] df.head()
```

	Date	Year	Month	Day	Hour	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	index	Temp40
0	2019-12-01 01:00	2019	12	1	10.0	16.0	4.0	0.4	89.0	22.7	96.0	1.354545	0.0
1	2019-12-01 02:00	2019	12	1	20.0	13.0	4.0	0.4	116.0	22.6	96.0	1.350000	0.0
2	2019-12-01 03:00	2019	12	1	30.0	16.0	3.0	0.3	69.0	22.5	96.0	1.366667	0.0
3	2019-12-01 04:00	2019	12	1	40.0	13.0	3.0	0.5	103.0	22.5	96.0	1.225000	0.0
4	2019-12-01 05:00	2019	12	1	50.0	11.0	2.0	0.3	129.0	22.5	96.0	1.883333	0.0

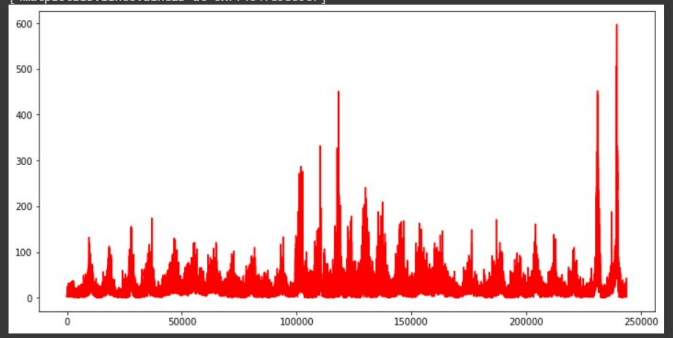
```
[ ] df.plot(x='Hour', y='PM2.5')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4053bbc990>
```




```
[ ] import matplotlib.pyplot as plt
    plt.figure(figsize=(12,6))
    plt.plot(df['PM2.5'],'r')
```

```
[<matplotlib.lines.Line2D at 0x7f404fe98090>]
```




```
[ ] from sklearn import linear_model

    x = df[['Year']]
    y = df['PM2.5']

    regr = linear_model.LinearRegression()
    regr.fit(x, y)
    a0 = regr.intercept_
```

สร้าง column ใหม่ แบ่ง temp น้อยและมาก และ plot graph

Linear Regression Model

```
[ ]  from sklearn import linear_model

     x = df[['Year']]
     y = df['PM2.5']

     regr = linear_model.LinearRegression()
     regr.fit(x, y)
     a0 = regr.intercept_
     a1 = regr.coef_
     print(a0, a1)
```

```
1899.0644713031472 [-0.92844995]
```

```
[ ]  df['YrForecast'] = a0 + a1*df['Year']
     df['PM2.5/Yr'] = df['PM2.5'] / df['YrForecast']
     df.head()
```

	Date	Year	Month	Day	Hour	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	index	Temp40	YrForecast	PM2.5/Yr
0	2019-12-01 01:00	2019	12	1	10.0	16.0	4.0	0.4	89.0	22.7	96.0	1.354545	0.0	24.524023	0.163105
1	2019-12-01 02:00	2019	12	1	20.0	13.0	4.0	0.4	116.0	22.6	96.0	1.350000	0.0	24.524023	0.163105
2	2019-12-01 03:00	2019	12	1	30.0	16.0	3.0	0.3	69.0	22.5	96.0	1.366667	0.0	24.524023	0.122329
3	2019-12-01 04:00	2019	12	1	40.0	13.0	3.0	0.5	103.0	22.5	96.0	1.225000	0.0	24.524023	0.122329
4	2019-12-01 05:00	2019	12	1	50.0	11.0	2.0	0.3	129.0	22.5	96.0	1.883333	0.0	24.524023	0.081553

```
[ ]  from sklearn import linear_model

     x1 = df[['Month']]
     y1 = df['PM2.5/Yr']

     regr = linear_model.LinearRegression()
```

ใช้ Linear Regression ระหว่าง ปี และ PM2.5 เพื่อดู trend ระยะยาว

Linear Regression Model

```
[ ] from sklearn import linear_model

    x1 = df[['Month']]
    y1 = df['PM2.5/Yr']

    regr = linear_model.LinearRegression()
    regr.fit(x1, y1)
    b0 = regr.intercept_
    b1 = regr.coef_
    print(b0, b1)
```

```
1.3264339028048946 [-0.05080388]
```

```
[ ] df['MForecast'] = b0 + b1*df['Month']
    df['PM2.5/M'] = df['PM2.5'] / df['MForecast']
    df.head()
```

	Date	Year	Month	Day	Hour	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	index	Temp40	YrForecast	PM2.5/Yr	MForecast	PM2.5/M
0	2019-12-01 01:00	2019	12	1	10.0	16.0	4.0	0.4	89.0	22.7	96.0	1.354545	0.0	24.524023	0.163105	0.716787	5.580456
1	2019-12-01 02:00	2019	12	1	20.0	13.0	4.0	0.4	116.0	22.6	96.0	1.350000	0.0	24.524023	0.163105	0.716787	5.580456
2	2019-12-01 03:00	2019	12	1	30.0	16.0	3.0	0.3	69.0	22.5	96.0	1.366667	0.0	24.524023	0.122329	0.716787	4.185342
3	2019-12-01 04:00	2019	12	1	40.0	13.0	3.0	0.5	103.0	22.5	96.0	1.225000	0.0	24.524023	0.122329	0.716787	4.185342
4	2019-12-01 05:00	2019	12	1	50.0	11.0	2.0	0.3	129.0	22.5	96.0	1.883333	0.0	24.524023	0.081553	0.716787	2.790228

```
[ ] from sklearn import linear_model

    X = df[['Hour', 'PM10', 'Wind speed', 'Wind dir', 'Temp', 'Rel hum', 'index', 'Temp40']]
    Y = df['PM2.5/M']
```

นำข้อมูลที่ได้ จากสมการแรก มาเปรียบเทียบกับเดือน ด้วย Linear Regressionเพื่อหา Seasonal

Linear Regression Model

```
[ ] from sklearn import linear_model

    X = df[['Hour', 'PM10', 'Wind speed', 'Wind dir', 'Temp', 'Rel hum', 'index', 'Temp40']]
    Y = df['PM2.5/M']

    regr = linear_model.LinearRegression()
    regr.fit(X, Y)
    c0 = regr.intercept_
    c1 = regr.coef_[0]
    c2 = regr.coef_[1]
    c3 = regr.coef_[2]
    c4 = regr.coef_[3]
    c5 = regr.coef_[4]
    c6 = regr.coef_[5]
    c7 = regr.coef_[6]
    c8 = regr.coef_[7]
    print(c0, c1, c2, c3, c4, c5, c6, c7, c8)
```

```
20.437409965348824 -0.007932379515054476 0.6000160281647597 -0.5166667678529522 -0.004644518158401889 -0.494973892243922 -0.07766773410748254 -0.44488973350359173 0.587705582596042
```

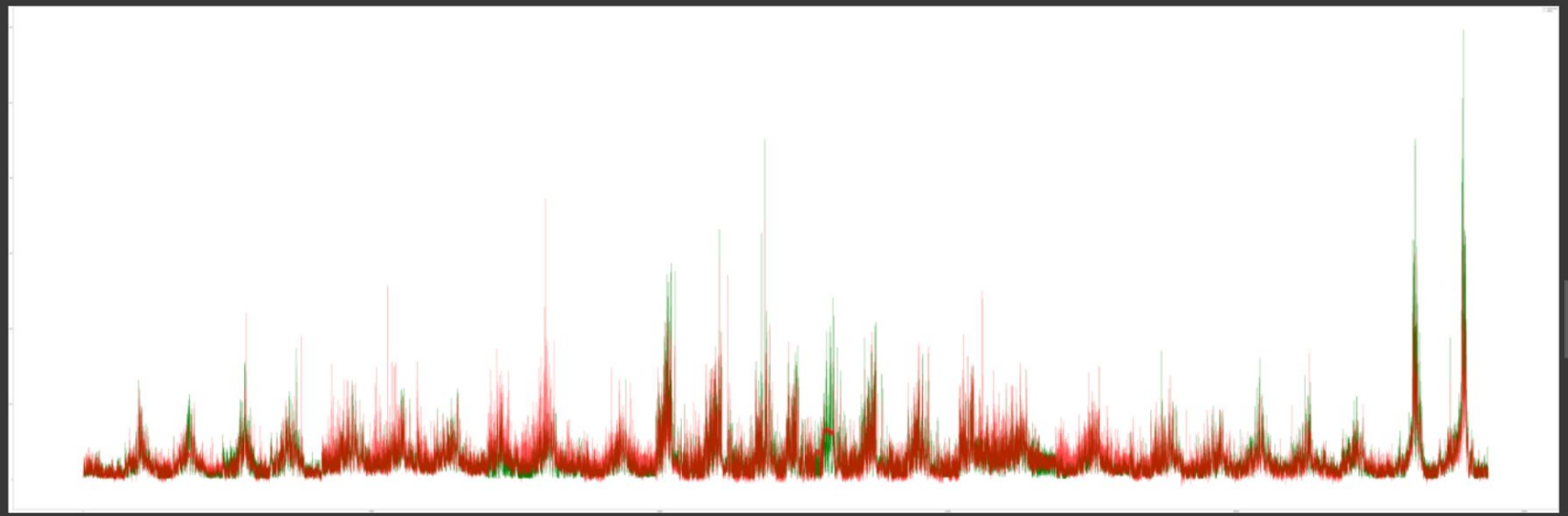
```
[ ] df['PM2.5f'] = c0 + c1*df['Hour'] + c2*df['PM10'] + c3*df['Wind speed'] + c4*df['Wind dir'] + c5*df['Temp'] + c6*df['Rel hum'] + c7*df['index']  + c8*df['Temp40']

    df.head()
```

	Date	Year	Month	Day	Hour	PM10	PM2.5	Wind speed	Wind dir	Temp	Rel hum	index	Temp40	YrForecast	PM2.5/Yr	MForecast	PM2.5/M	PM2.5f
0	2019-12-01 01:00	2019	12	1	10.0	16.0	4.0	0.4	89.0	22.7	96.0	1.354545	0.0	24.524023	0.163105	0.716787	5.580456	10.043681
1	2019-12-01 02:00	2019	12	1	20.0	13.0	4.0	0.4	116.0	22.6	96.0	1.350000	0.0	24.524023	0.163105	0.716787	5.580456	8.090426
2	2019-12-01 03:00	2019	12	1	30.0	16.0	3.0	0.3	69.0	22.5	96.0	1.366667	0.0	24.524023	0.122329	0.716787	4.185342	10.123192

นำข้อมูลจากสมการ มาเข้า Linear Regression เปรียบเทียบกับตัวแปรอื่นๆ

Linear Regression Model

```
[ ]  from matplotlib.pyplot import figure
     figure(num=None, figsize=(180, 60), dpi=80, facecolor='w', edgecolor='k')
     plt.plot(df['PM2.5'], color='g',label='ground truth')
     plt.plot(df['PM2.5f'], alpha=.7, color='r',label='predict')
     plt.legend(loc="upper right")
     plt.show()
```



Plot graph โดยที่ข้อมูล มี RMSE อยู่ที่ 11.51

Data Pipeline

Using Airflow

Raw Python Model

Web scraping

Regression model

Google sheet Python

Pipeline Model

Apache Airflow

WebScraping → Predict_Model → Send_GoogleSheet

Data

Google sheet excel

visualization

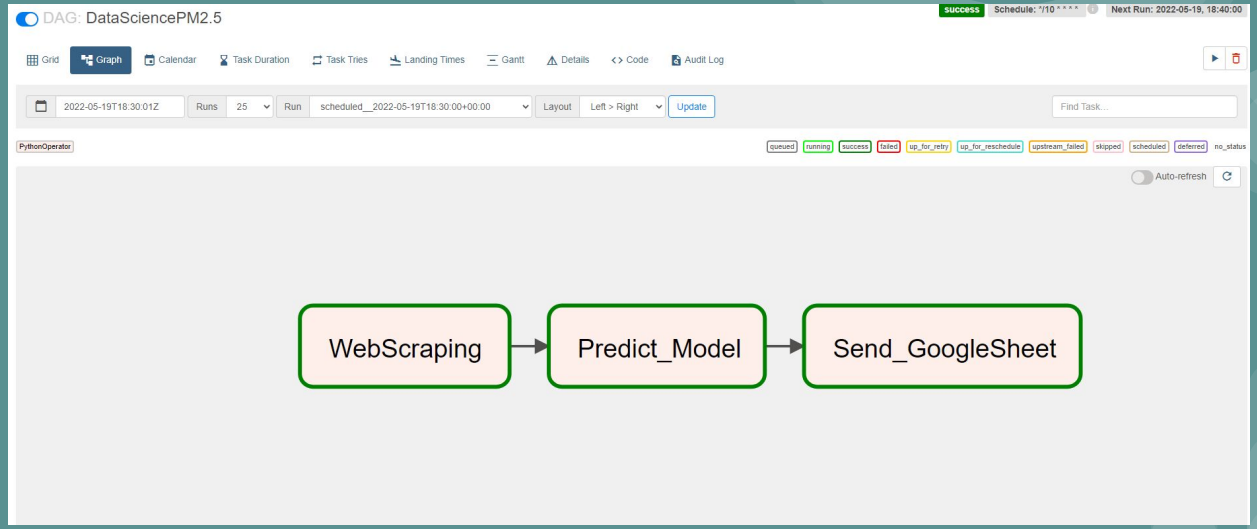
Power BI

ประกอบด้วย 3 Operator หลักๆ คือ

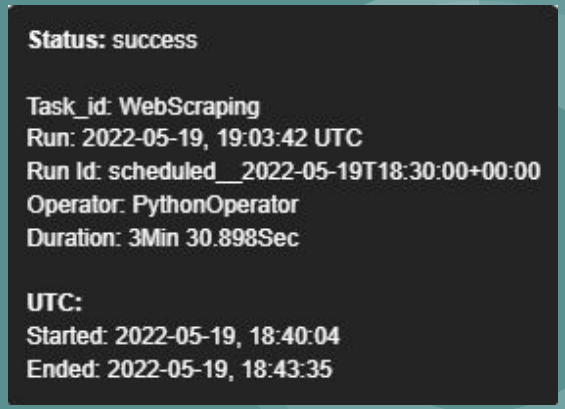
-Web Scraping Operator

-Predict Model Operator

-Send_GoogleSheet Operator



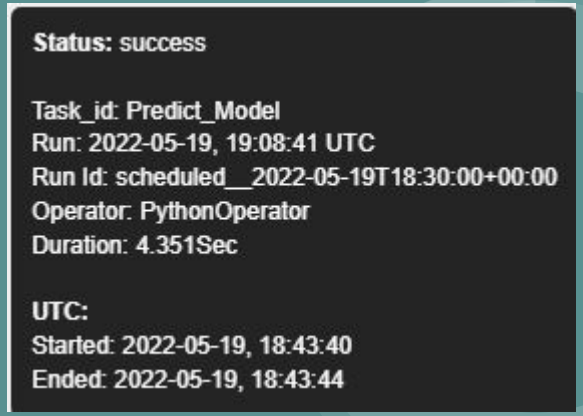
Web Scraping Operator



ดึง ค่าจาก air4thai และ traffic ออกมา ใส่ตัว list ของเราโดย จะส่งค่า real time ที่ได้จาก การ scrap ผ่านหน้าเว็บ เก็บเป็น list และส่งเข้าไปยัง predict model operator ต่อ
Type ที่เราให้คือ Pythonoperator โดยมีการเชื่อมต่อกับ xcom_push เพื่อส่ง Data ไปยัง Predict Model Operator ต่อ

```
ti.xcom_push(key ='PushData',value = li)
print("Model Here Sent : ",li)
```

Predict Model Operator

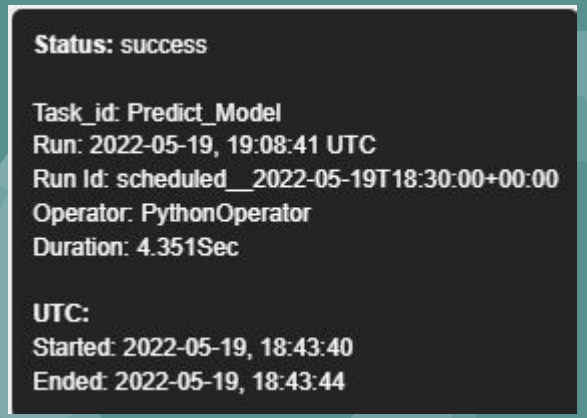


1.Predict ค่าที่ได้จาก Web Scraping Operator ของเรา
2.ใช้ Regression Model ที่ได้รับการ Optimize มาแล้ว ตาม machine learning Model
3.เมื่อทำการ Predict เสร็จ ผลลัพธ์ที่ได้ออกมาคือ PM2.5 Forcast มาบอกความแม่นยำของ Model เทียบกับ PM2.5 ของจริง
4.เก็บใน list และส่งให้กับ Send_googleSheet Operator ต่อเพื่อส่งไปยัง google sheet
5.ใช้ Ti.xcom_Push() และ Ti.xcom_Pull() ในการดึง value จากตัว operator ข้างนอก และส่ง ค่าเข้า sendgoogle_sheet

```
st_list = ti.xcom_pull(key = 'PushData',task_ids='WebScraping')

Result = []
```

```
ti.xcom_push(key ='PushResult',value = Result)

print("Model Here Recieve : ",Result)
```

Send_googleSheet Operator




```
Result = ti.xcom_pull(key = 'PushResult',task_ids='Predict_Model')
```

1.Predict ค่าที่ได้จาก Web Scraping Operator ของเรา
2.ใช้ Regression Model ที่ได้รับการ Optimize มาแล้ว ตาม machine learning Model
3.เมื่อทำการ Predict เสร็จ ผลลัพธ์ที่ได้ออกมาคือ PM2.5 Forcast มาบอกความแม่นยำของ Model เทียบกับ PM2.5 ของจริง
4.เก็บใน list และส่งให้กับ Send_googleSheet Operator ต่อเพื่อส่งไปยัง google sheet

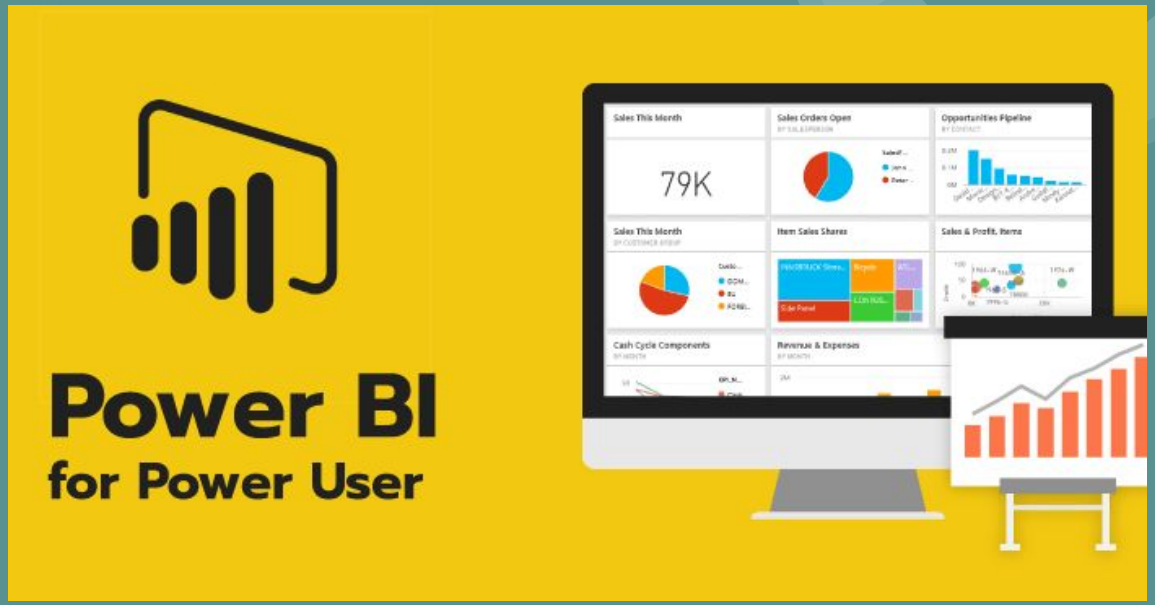
ตัวอย่าง Excel ที่ได้จากการ ส่งค่าผ่าน Google sheet operator

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Date	Year	Month	Day	Hour	Station	PM10	PM2.5	Wind speed	Wind direction	Temp	Real Humidity	Index	Latitude	Longtitude
2022-05-19 20:0	2022	5	19	200	02t	21.0	13.0	0.3	44.0	30.8	63.0	3.1	13.732846	100.487662
2022-05-19 21:0	2022	5	19	210	02t	15.0	10	0.3	34.0	30.7	64.0	1.8	13.732846	100.487662
2022-05-19 21:0	2022	5	19	210	03t	20.0	8	0.0	0.0	29.2	0.0	1.8	13.636514	100.414262
2022-05-19 21:0	2022	5	19	210	05t	14.0	10	2.1	181.0	29.4	81.0	1.8	13.666183	100.605742
2022-05-19 21:0	2022	5	19	210	10t	18.0	7	1.2	201.0	30.8	0.0	1.8	13.779873	100.646009
2022-05-19 21:0	2022	5	19	210	11t	20.0	3	0.4	162.0	30.3	60.0	1.8	13.77553	100.569195
2022-05-19 21:0	2022	5	19	210	12t	31.0	23	0.0	0.0	30.2	86.0	1.8	13.70806667	100.5473333
2022-05-19 21:0	2022	5	19	210	50t	31.0	14	0.0	0.0	29.9	66.0	1.8	13.729852	100.536501
2022-05-19 21:0	2022	5	19	210	52t	15.0	10	0.7	258.0	0.0	0.0	1.8	13.727622	100.486568
2022-05-19 21:0	2022	5	19	210	53t	17.0	6	0.9	73.0	29.5	65.0	1.8	13.7954248	100.5930298
2022-05-19 21:0	2022	5	19	210	54t	45.0	25	0.1	198.0	29.3	74.0	1.8	13.76251667	100.5502
2022-05-19 22:0	2022	5	19	220	02t	20.0	12	0.3	18.0	30.6	64.0	2.0	13.732846	100.487662
2022-05-19 22:0	2022	5	19	220	03t	20.0	29	0.0	0.0	28.8	0.0	2.0	13.636514	100.414262
2022-05-19 22:0	2022	5	19	220	05t	15.0	9	2.1	182.0	29.2	83.0	2.0	13.666183	100.605742
2022-05-19 22:0	2022	5	19	220	10t	18.0	7	0.9	196.0	30.7	0.0	2.0	13.779873	100.646009
2022-05-19 22:0	2022	5	19	220	11t	20.0	3	0.2	158.0	30.2	60.0	2.0	13.77553	100.569195
2022-05-19 22:0	2022	5	19	220	02t	20.0	12	0.3	18.0	30.6	64.0	2.0	13.732846	100.487662

DEMO

Data Visualization

Using PowerBI



Key Point from data visualization

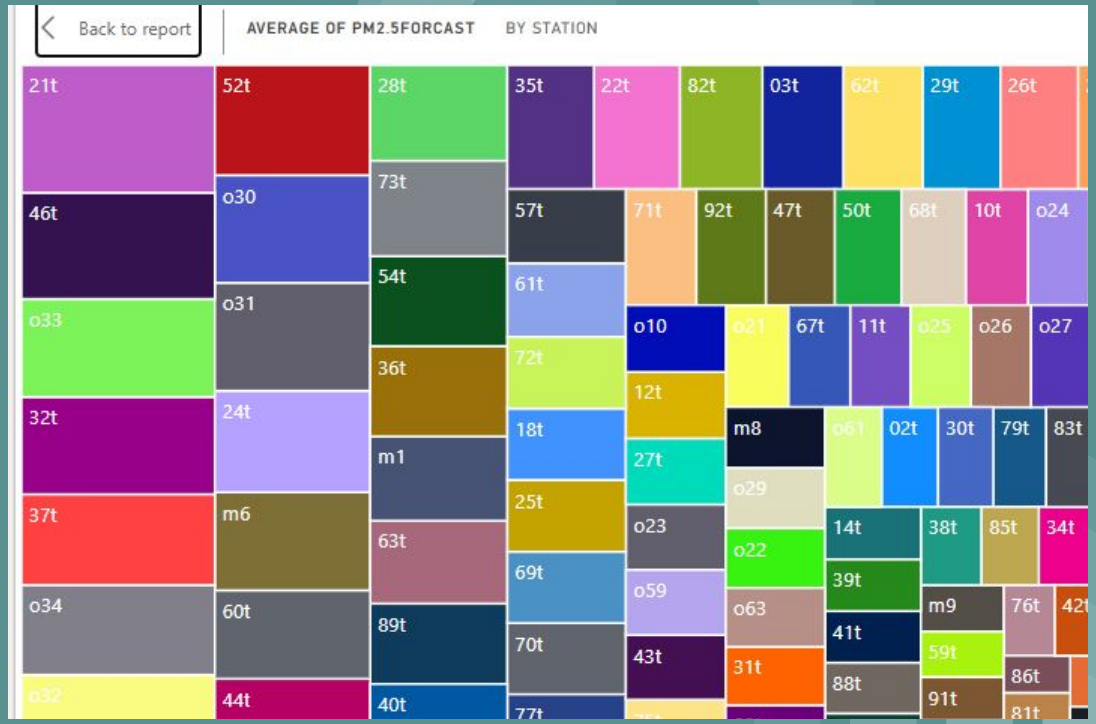
-เราจำแนก station ออกมาเพื่อดูว่า สถานีไหน มี PM2.5 ที่มีค่ามากที่สุด ตาม geography
-เราเปรียบเทียบ PM2.5 ตามช่วงเวลาในแต่ละชม ว่า trend ในแต่ละ สถานีต่างกันหรือไม่
-เรา visualize scatter plot เพื่อ เปรียบเทียบการแปรผันของ PM2.5 ว่าแต่ละตัวแปร (Humidity,Wind speed,Temp) ,มีผลกับค่าที่ forcast ได้อย่างไร
-เราหยิบ station ที่มีค่ามากสุดมา plot กราฟ ดูความคลาดเคลื่อนของ model ของเรา

data visualization Summarise

-เราพบว่า ในกลุ่ม station 5 ลำดับแรกที่มี predit สูงที่สุด 3 ใน 5 ตั้งอยู่ใน กรุงเทพมหานคร (ถ้าเทียบกับ latitude longtitude) แสดงว่า trend กรุงเทพมหานคร มี PM 2.5 ที่ค่อนข้างสูง (21t,s2t,d32)

data visualization Summarise

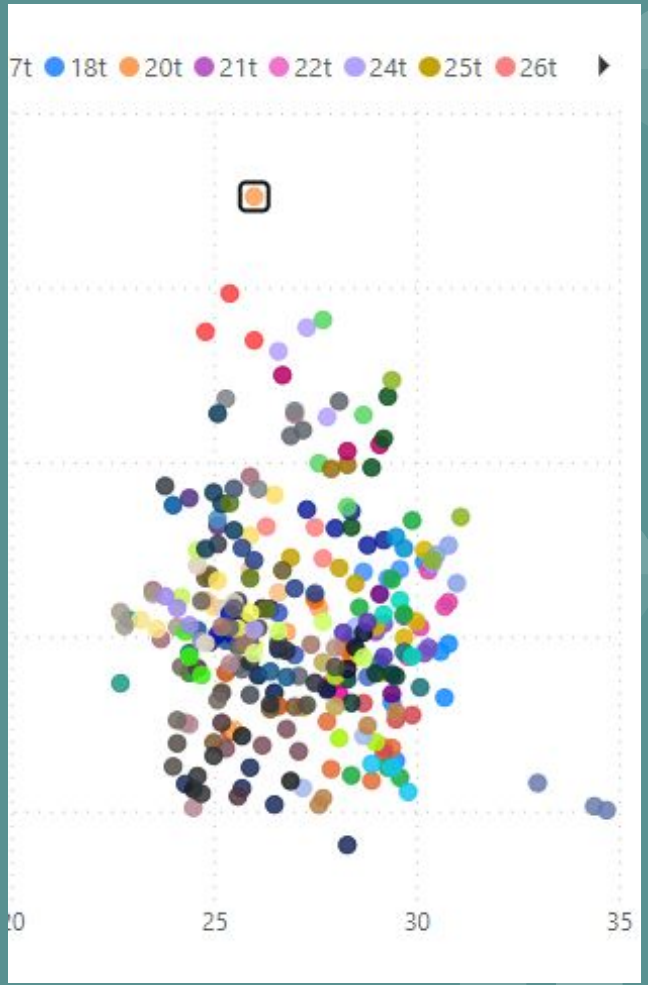
-แต่ถ้านอกเหนือจาก กรุงเทพแล้ว ก็จะกระจายตามจุดต่างๆ ทั่วประเทศ เราเลยจะวัดปัจจัยว่า อะไรมีผลต่อค่าของ PM2.5 บ้าง



Temperature

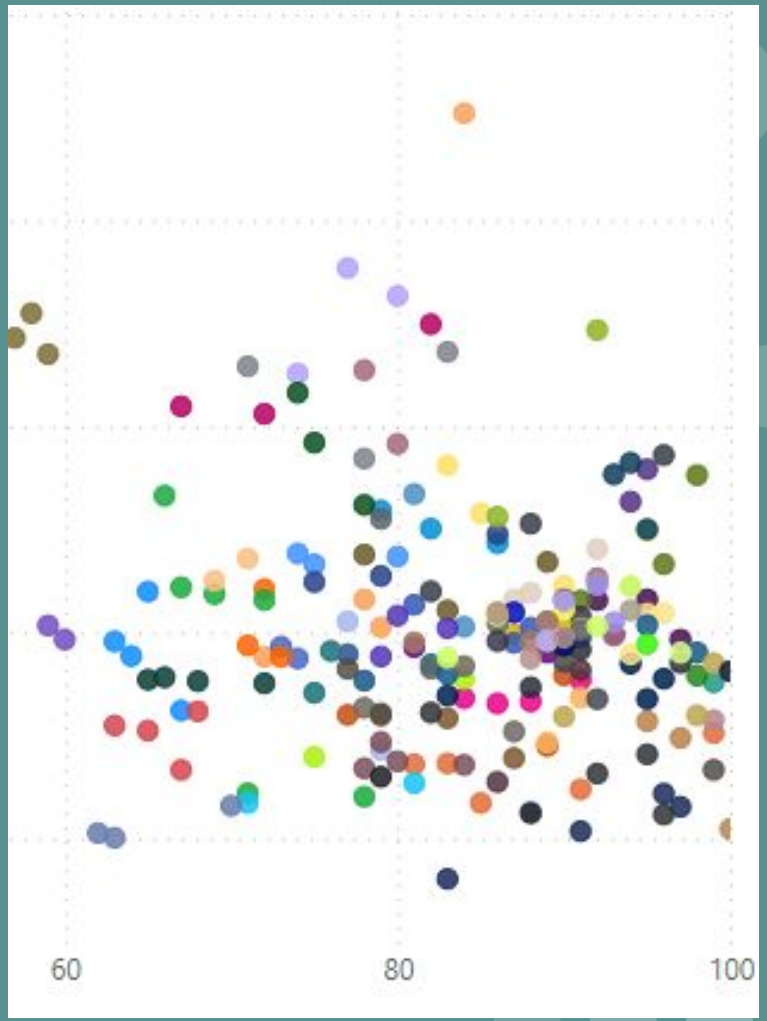
-จากการ plot กราฟ Scatter Plot พบว่า ช่วงอุณหภูมิ ที่ PM2.5 กระจุกกันแน่นหนานั้น จะอยู่ที่ ช่วง 25-27 องศา ซึ่งค่าสูงสุดก็จะได้ ณ ที่จุดนั้น ที่ station 20t (PM2.5 = 36)

-ทั้งนี้ ข้อมูลอาจมีการคลาดเคลื่อน เนื่องจาก เราเก็บ ข้อมูล ณ ปัจุบันได้แค่ 1 วัน ทำให้ เห็นการกระจายแค่ช่วงกลางคืน ไม่เห็นกลางวันมากนัก



Humidity

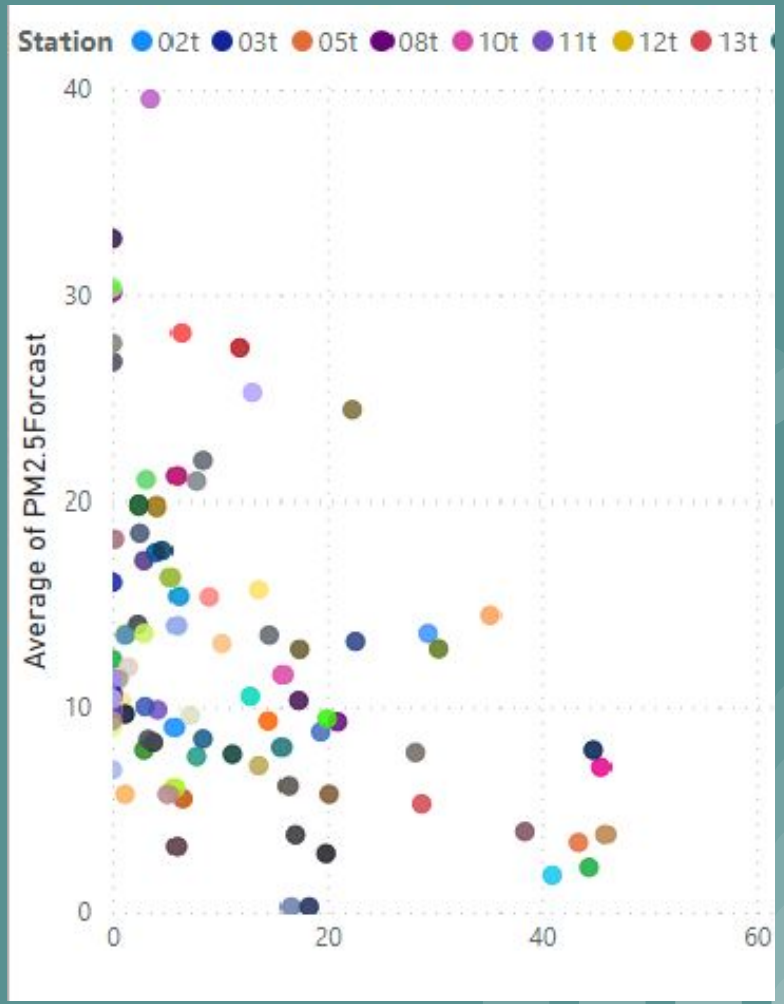
-จากการ plot กราฟ Scatter Plot พบว่า trend ของ การกระจายตัวของข้อมูลจะค่อยๆ ตีวงแคบลงเรื่อยๆ ในช่วง Humidity ระหว่าง 60-80 เมื่อเข้าใกล้ ช่วง 70-80 แล้ว ค่าที่เก็บได้ส่วนใหญ่ จะกระจายตัวอยู่ระหว่าง แค่ 10 PM เท่านั้น
-เราเลยตีความว่า ความชื้นสูงมาผลทำให้ ค่า PM25 ในอากาศลดลง



Wind Speed

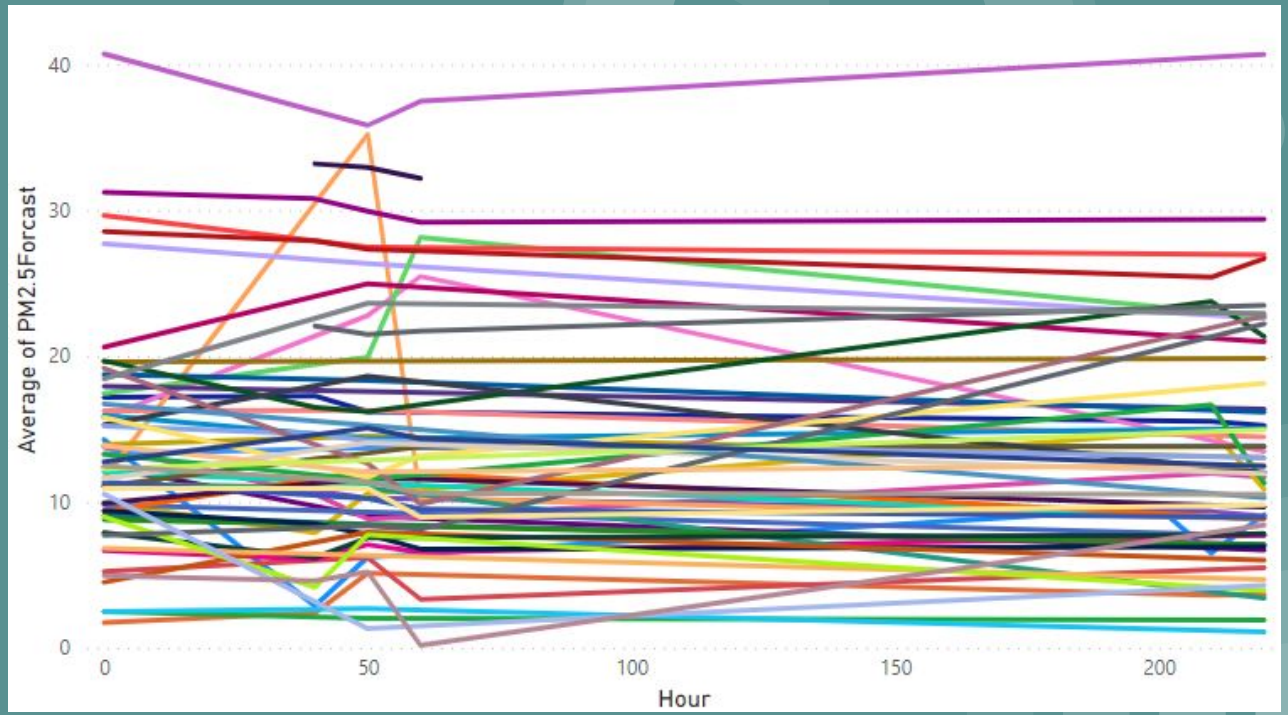
-จากการ plot กราฟ Scatter Plot พบว่า ค่า PM25 ที่มี ค่าสูง ส่วนใหญ่แล้ว จะพบในจุดที่ ไม่มีลม - มีลมน้อย แต่ค่าการกระจายตัวจะน้อยลงเมื่อมีลมมากขึ้น แล้ว trend ของ PM25 ก็น้อยตามด้วย สุดท้ายเมื่อถึง wind speed 40 จะพบว่า PM25 น้อยกว่า 10 เกือบทั้งหมด

-เราเลยตีความว่า ความเร็วลมส่งผลต่อ PM2.5 โดยที่ station ส่วนใหญ่มีความเร็วลมเป็น 0 แต่เมื่อมีลมเข้ามาเกี่ยว ค่า PM2.5 จะลดลงเรื่อยๆ และต่ำสุดเมื่อแตะ 40



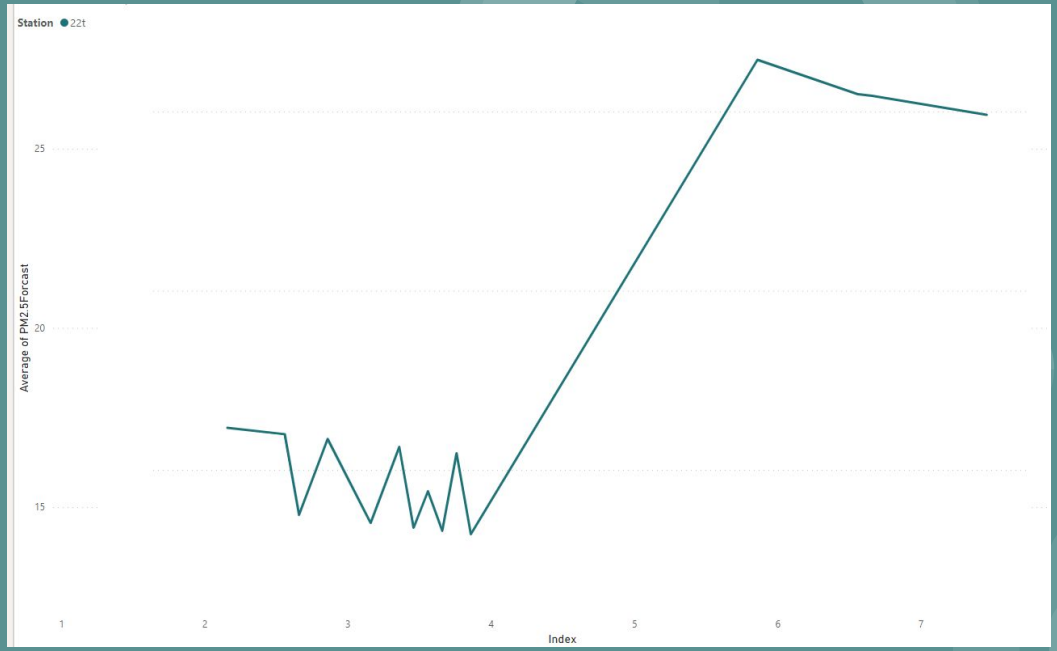
Time

-ช่วงเวลาไม่ค่อยส่งผลต่อการเปลี่ยนแปลงของ PM2.5 แต่มีความคลาดเคลื่อนตรงที่ว่า เราจับ Data แค่วันเดียว ถ้าหาก จับเป็นสัปดาห์อาจเห็น การเปลี่ยนแปลงมากกว่านี้



Index

-จากการเทียบพบว่าช่วงค่า index ต่ำ pm2.5 จะต่ำ แต่พอรถติด = PM2.5 สูง ค่ากราฟก็จะสูงตามด้วย



Summary

-Wind Speed มาก PM2.5 ต่ำ ส่วนใหญ่ Station จะอยู่ที่ 0

-Humidity มาก PM2.5 กระจายตัวน้อยลง กระจุกมากขึ้น โดย Trend ค่า PM2.5 จะลดลง

-Temp ยังมีปัญหาของการที่ค่าที่เก็บส่วนใหญ่อยู่ในช่วงกลางคืน ทำให้อุณหภูมิกระจุก กันอยู่ใน Time เดียวกัน แต่จะส่งผลให้เกิด ค่า PM2.5 มากสุดที่ช่วง 25-27 C