บทที่ 4

ผลการดำเนินงาน

ผลการดำเนินงาน ของโครงงานเรื่อง การสร้างโมเดลการพยากรณ์ ผลกระทบของปัจจัยทางสภาพภูมิอากาศ ต่อความเข้มข้นฝุ่นละอองขนาด 2.5 ไมครอน ในเขตอำเภอเมือง จังหวัดเชียงใหม่ โดยใช้การเรียนรู้ของเครื่อง (MACHINE LEARNING)

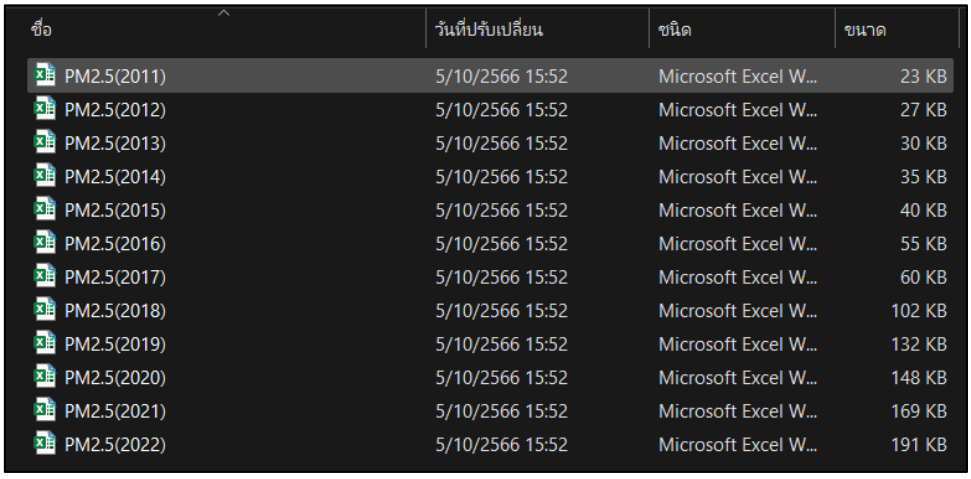
4.1 กระบวนการศึกษาทำความเข้าใจธุรกิจ (Business Understanding)

กระบวนการศึกษาทำความเข้าใจธุรกิจ (Business Understanding) เป็นขั้นตอนแรกของกระบวนการ ที่มุ่งเน้นไปที่การทำความเข้าใจกระบวนการทางธุรกิจโดยรวมเป็นขั้นตอนแรกของกระบวนการ ที่มุ่งเน้นไปที่การทำความเข้าใจกระบวนการทางธุรกิจโดยรวม ผู้วิเคราะห์ข้อมูลทำความเข้าใจกับปัญหาให้อยู่ในรูปของการวิเคราะห์ข้อมูลทาง Data Mining โดยการวิเคราะห์ข้อมูลในประเด็นนี้ ข้อมูลจะถูกจัดเก็บโดยหน่วยงานราชการ ซึ่งมีทั้งหมด 2 หน่วยงาน 1. กรมควบคุมมลพิษ 2. ศูนย์อุตุนิยมวิทยาภาคเหนือ ซึ่งข้อมูลจะถูกจัดเก็บไว้จากทั้งประเทศโดยแยกออกเป็นหน่วยพื้นที่ต่าง ๆ

4.2 การทำความเข้าใจข้อมูล (Data Understanding)

การทำความเข้าใจข้อมูล (Data Understanding) ขั้นตอนการจัดเก็บและรวบรวม ข้อมูล ตลอดจนการพิจารณาตรวจสอบความถูกต้องของข้อมูลที่ได้รับ โดยเลือกว่าจะใช้ข้อมูล ทั้งหมดหรือบางส่วนในการวิเคราะห์ให้สอดรับกับวัตถุประสงค์ที่กำหนดไว้

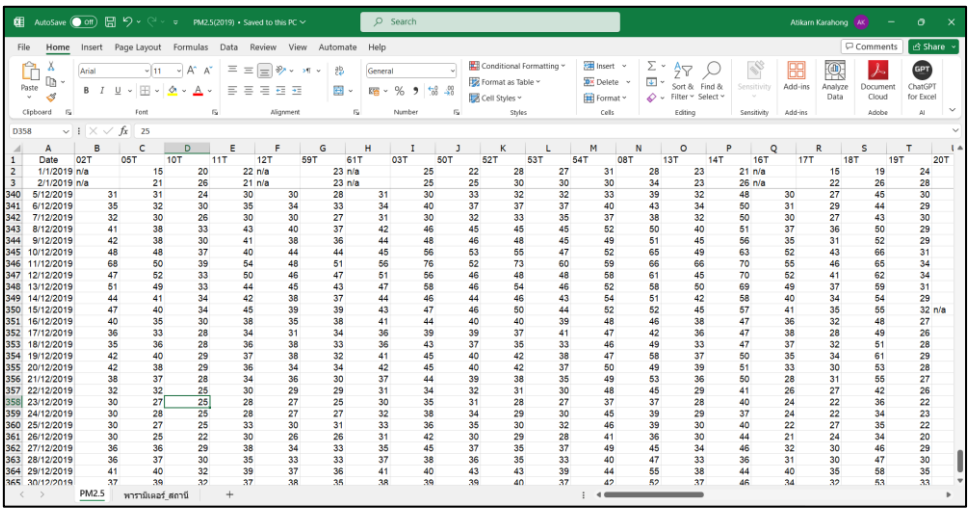
ข้อมูล PM2.5 ตั้งแต่ปี 2561 – 2565 โดยจะมีข้อมูลแยกออกเป็นปีละไฟล์



ชื่อ	วันที่ปรับเปลี่ยน	ชนิด	ขนาด
PM2.5(2011)	5/10/2566 15:52	Microsoft Excel W...	23 KB
PM2.5(2012)	5/10/2566 15:52	Microsoft Excel W...	27 KB
PM2.5(2013)	5/10/2566 15:52	Microsoft Excel W...	30 KB
PM2.5(2014)	5/10/2566 15:52	Microsoft Excel W...	35 KB
PM2.5(2015)	5/10/2566 15:52	Microsoft Excel W...	40 KB
PM2.5(2016)	5/10/2566 15:52	Microsoft Excel W...	55 KB
PM2.5(2017)	5/10/2566 15:52	Microsoft Excel W...	60 KB
PM2.5(2018)	5/10/2566 15:52	Microsoft Excel W...	102 KB
PM2.5(2019)	5/10/2566 15:52	Microsoft Excel W...	132 KB
PM2.5(2020)	5/10/2566 15:52	Microsoft Excel W...	148 KB
PM2.5(2021)	5/10/2566 15:52	Microsoft Excel W...	169 KB
PM2.5(2022)	5/10/2566 15:52	Microsoft Excel W...	191 KB

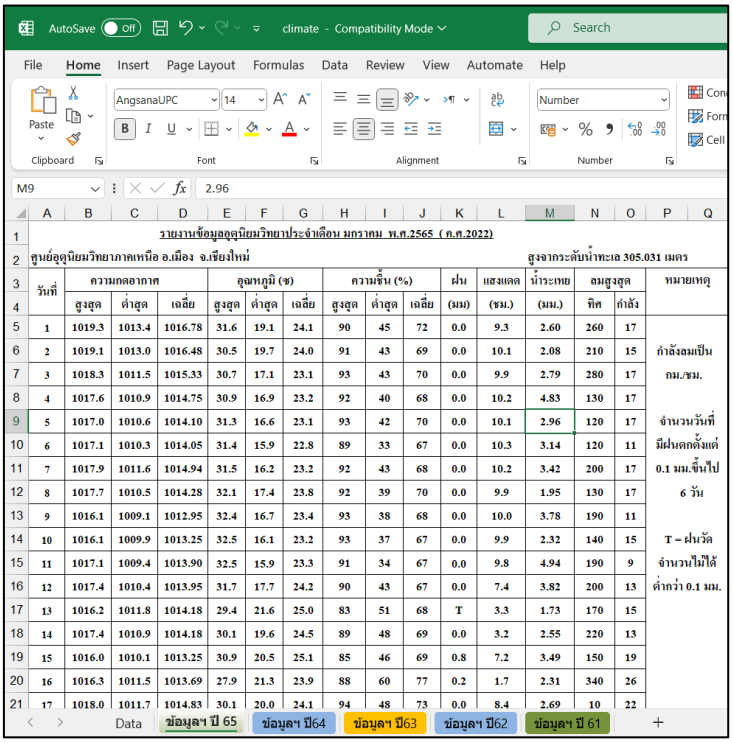
ภาพที่ 4.1 ข้อมูล PM2.5 ที่ได้มาจากกรมควบคุมมลพิษ

ในแต่ละไฟล์จะมีการเก็บข้อมูลแบ่งย่อยออกตามสถานีเก็บข้อมูล ซึ่งอำเภอเมืองจังหวัดเชียงใหม่จะเป็นสถานี T34 และ T35



ภาพที่ 4.2 ข้อมูล PM2.5 ที่ถูกจัดเก็บและบันทึกไว้

ข้อมูลสภาพอากาศถูกเก็บไว้ในไฟล์เดียว โดยทำการแบ่ง Sheet ออกเป็นแต่ละปี โดยมีข้อมูลตั้งแต่ปี 2561-2565 โดยมีคอลลัมที่ต้องการใช้ในการเทรนโมเดลทำนายครั้งนี้มีทั้งหมด 5 คอลัม คือ 1.ความกดอากาศ 2.อุณหภูมิ 3.ความชื้น 4.ฝน 5.ความเร็วลมสูงสุด



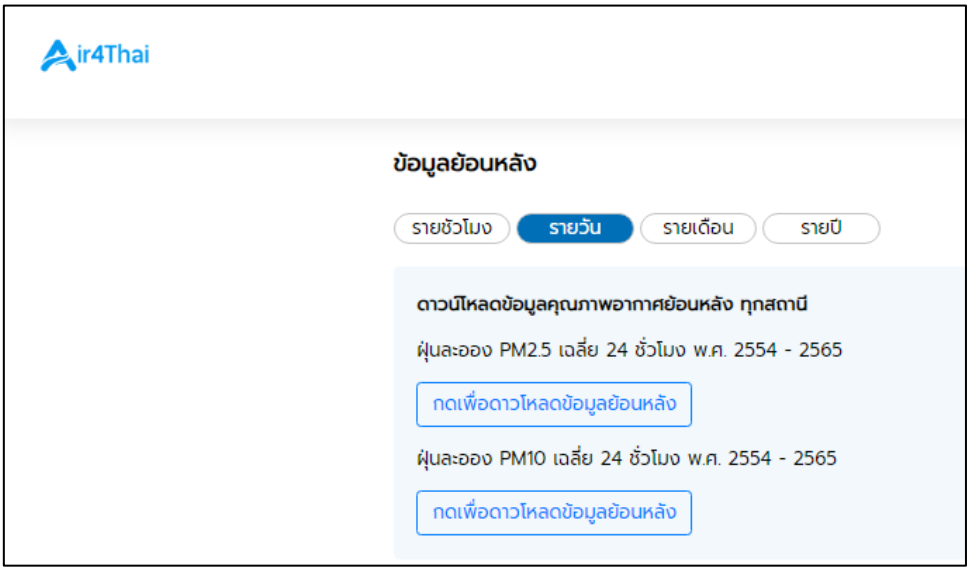
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	รายงานข้อมูลอุตุนิยมวิทยาประจำเดือน มกราคม พ.ศ.2565 (ค.ศ.2022)																
2	ศูนย์อุตุนิยมวิทยาภาคเหนือ อ.เมือง จ.เชียงใหม่														สูงจากระดับน้ำทะเล 305.031 เมตร		
3	วันที่	ความกดอากาศ			อุณหภูมิ (ซ)			ความชื้น (%)			ฝน	แสงแดด	น้ำระเหย	ลมสูงสุด		หมายเหตุ	
4		สูงสุด	ต่ำสุด	เฉลี่ย	สูงสุด	ต่ำสุด	เฉลี่ย	สูงสุด	ต่ำสุด	เฉลี่ย	(มม)	(ชม.)	(มม.)	ทิศ	กำลัง		
5	1	1019.3	1013.4	1016.78	31.6	19.1	24.1	90	45	72	0.0	9.3	2.60	260	17		
6	2	1019.1	1013.0	1016.48	30.5	19.7	24.0	91	43	69	0.0	10.1	2.08	210	15	กำลังลมเป็น	
7	3	1018.3	1011.5	1015.33	30.7	17.1	23.1	93	43	70	0.0	9.9	2.79	280	17	กม./ชม.	
8	4	1017.6	1010.9	1014.75	30.9	16.9	23.2	92	40	68	0.0	10.2	4.83	130	17		
9	5	1017.0	1010.6	1014.10	31.3	16.6	23.1	93	42	70	0.0	10.1	2.96	120	17	จำนวนวันที่	
10	6	1017.1	1010.3	1014.05	31.4	15.9	22.8	89	33	67	0.0	10.3	3.14	120	11	มีฝนตกตั้งแต่	
11	7	1017.9	1011.6	1014.94	31.5	16.2	23.2	92	43	68	0.0	10.2	3.42	200	17	0.1 มม.ขึ้นไป	
12	8	1017.7	1010.5	1014.28	32.1	17.4	23.8	92	39	70	0.0	9.9	1.95	130	17	6 วัน	
13	9	1016.1	1009.1	1012.95	32.4	16.7	23.4	93	38	68	0.0	10.0	3.78	190	11		
14	10	1016.1	1009.9	1013.25	32.5	16.1	23.2	93	37	67	0.0	9.9	2.32	140	15	T = ฝนวัด	
15	11	1017.1	1009.4	1013.90	32.5	15.9	23.3	91	34	67	0.0	9.8	4.94	190	9	จำนวนไม่ได้	
16	12	1017.4	1010.4	1013.95	31.7	17.7	24.2	90	43	67	0.0	7.4	3.82	200	13	ต่ำกว่า 0.1 มม.	
17	13	1016.2	1011.8	1014.18	29.4	21.6	25.0	83	51	68	T	3.3	1.73	170	15		
18	14	1017.4	1010.9	1014.18	30.1	19.6	24.5	89	48	69	0.0	3.2	2.55	220	13		
19	15	1016.0	1010.1	1013.25	30.9	20.5	25.1	85	46	69	0.8	7.2	3.49	150	19		
20	16	1016.3	1011.5	1013.69	27.9	21.3	23.9	88	60	77	0.2	1.7	2.31	340	26		
21	17	1018.0	1011.7	1014.83	30.1	20.0	24.1	94	48	73	0.0	8.4	2.69	10	22		

ภาพที่ 4.3 ข้อมูลสภาพอากาศที่ได้จากศูนย์อุตุนิยมวิทยาภาคเหนือ

4.3 การเตรียมข้อมูล (Data Preparation)

4.3.1 การสกัดข้อมูลที่ต้องการ (Extract)

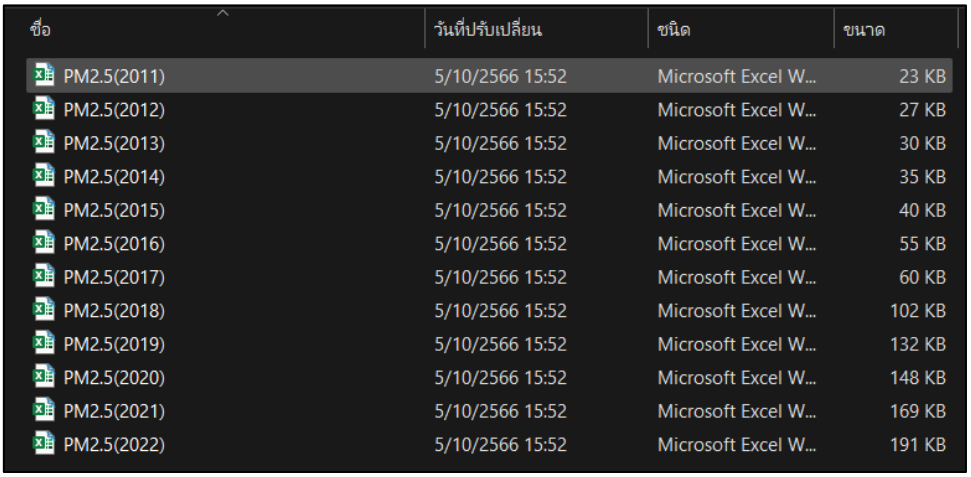
ข้อมูล PM2.5 ทำการรวบรวมมาจากเว็บไซต์ของ กรมควบคุมมลพิษ เข้าถึงได้จาก http://air4thai.pcd.go.th ซึ่งเป็นข้อมูลที่เปิดเผยต่อสาธารณะ



ภาพที่ 4.4 หน้าเว็บไซต์ของกรมควบคุมมลพิษ

(ที่มา : http://air4thai.pcd.go.th)

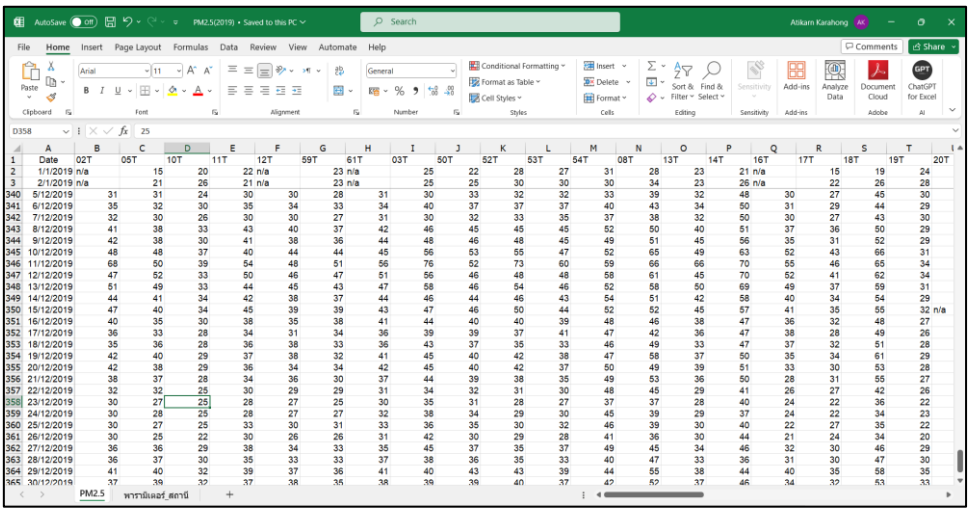
โดยข้อมูลที่ได้จะเป็นข้อมูล PM2.5 ตั้งแต่ปี 2561 – 2565 โดยจะมีข้อมูลแยกออกเป็นปีละไฟล์



ชื่อ	วันที่ปรับเปลี่ยน	ชนิด	ขนาด
PM2.5(2011)	5/10/2566 15:52	Microsoft Excel W...	23 KB
PM2.5(2012)	5/10/2566 15:52	Microsoft Excel W...	27 KB
PM2.5(2013)	5/10/2566 15:52	Microsoft Excel W...	30 KB
PM2.5(2014)	5/10/2566 15:52	Microsoft Excel W...	35 KB
PM2.5(2015)	5/10/2566 15:52	Microsoft Excel W...	40 KB
PM2.5(2016)	5/10/2566 15:52	Microsoft Excel W...	55 KB
PM2.5(2017)	5/10/2566 15:52	Microsoft Excel W...	60 KB
PM2.5(2018)	5/10/2566 15:52	Microsoft Excel W...	102 KB
PM2.5(2019)	5/10/2566 15:52	Microsoft Excel W...	132 KB
PM2.5(2020)	5/10/2566 15:52	Microsoft Excel W...	148 KB
PM2.5(2021)	5/10/2566 15:52	Microsoft Excel W...	169 KB
PM2.5(2022)	5/10/2566 15:52	Microsoft Excel W...	191 KB

ที่ 4.5 ข้อมูล ภาพPM2.5 ที่ได้มาจากกรมควบคุมมลพิษ

ในแต่ละไฟล์จะมีการเก็บข้อมูลแบ่งย่อยออกตามสถานีเก็บข้อมูล ซึ่งอำเภอเมืองจังหวัดเชียงใหม่จะเป็นสถานี T34 และ T35



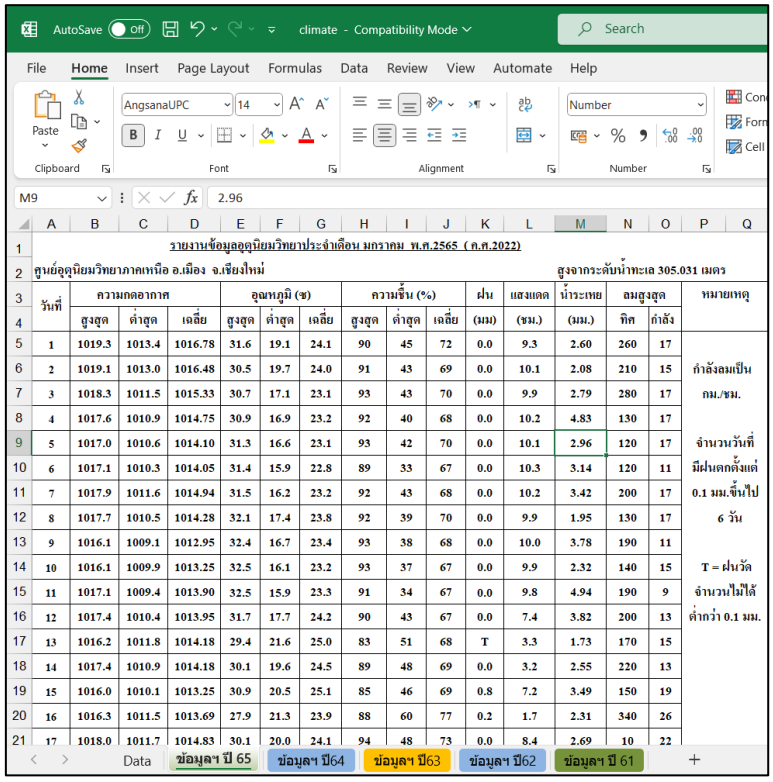
ภาพที่ 4.6 ข้อมูล PM2.5 ที่ถูกจัดเก็บและบันทึกไว้

ผู้วิเคราะห์ทำการรวบรวมข้อมูล เพื่อทำการวิเคราะห์ ข้อมูลความกดอากาศ ข้อมูลความชื้น ข้อมูลความเร็วลม ข้อมูลปริมาณน้ำฝน ข้อมูลอุณหภูมิ ทำการรวบรวมมาจากเว็บไซต์ของ ศูนย์อุตุนิยมวิทยาภาคเหนือ เข้าถึงได้จาก http://www.cmmet.tmd.go.th ซึ่งเป็นข้อมูลเปิดเผยต่อสาธารณะ

ภาพที่ 4.7 ศูนย์อุตุนิยมวิทยาภาคเหนือ

(ที่มา : http://www.cmmet.tmd.go.th)

ข้อมูลสภาพอากาศถูกเก็บไว้ในไฟล์เดียว โดยทำการแบ่ง Sheet ออกเป็นแต่ละปี โดยมีข้อมูลตั้งแต่ปี 2561–2565 โดยมีคอลลัมที่ต้องการใช้ในการเทรนโมเดลทำนายครั้งนี้มีทั้งหมด 5 คอลัม คือ 1.ความกดอากาศ 2.อุณหภูมิ 3.ความชื้น 4.ฝน 5.ความเร็วลมสูงสุด



AutoSave Off | climate - Compatibility Mode | Search

File Home Insert Page Layout Formulas Data Review View Automate Help

Paste | AngsanaUPC | 14 | Number | Clipboard | Font | Alignment | Number

M9 | 2.96

รายงานข้อมูลอุตุนิยมวิทยาประจำเดือน มกราคม พ.ศ.2565 (ค.ศ.2022)

ศูนย์อุตุนิยมวิทยาภาคเหนือ อ.เมือง จ.เชียงใหม่ — สูงจากระดับน้ำทะเล 305.031 เมตร

วันที่	ความกดอากาศ			อุณหภูมิ (ซ)			ความชื้น (%)			ฝน	แสงแดด	น้ำระเหย	ลมสูงสุด		หมายเหตุ
	สูงสุด	ต่ำสุด	เฉลี่ย	สูงสุด	ต่ำสุด	เฉลี่ย	สูงสุด	ต่ำสุด	เฉลี่ย	(มม)	(ชม.)	(มม.)	ทิศ	กำลัง	
1	1019.3	1013.4	1016.78	31.6	19.1	24.1	90	45	72	0.0	9.3	2.60	260	17	
2	1019.1	1013.0	1016.48	30.5	19.7	24.0	91	43	69	0.0	10.1	2.08	210	15	กำลังลมเป็น
3	1018.3	1011.5	1015.33	30.7	17.1	23.1	93	43	70	0.0	9.9	2.79	280	17	กม./ชม.
4	1017.6	1010.9	1014.75	30.9	16.9	23.2	92	40	68	0.0	10.2	4.83	130	17	
5	1017.0	1010.6	1014.10	31.3	16.6	23.1	93	42	70	0.0	10.1	2.96	120	17	จำนวนวันที่
6	1017.1	1010.3	1014.05	31.4	15.9	22.8	89	33	67	0.0	10.3	3.14	120	11	มีฝนตกตั้งแต่
7	1017.9	1011.6	1014.94	31.5	16.2	23.2	92	43	68	0.0	10.2	3.42	200	17	0.1 มม.ขึ้นไป
8	1017.7	1010.5	1014.28	32.1	17.4	23.8	92	39	70	0.0	9.9	1.95	130	17	6 วัน
9	1016.1	1009.1	1012.95	32.4	16.7	23.4	93	38	68	0.0	10.0	3.78	190	11	
10	1016.1	1009.9	1013.25	32.5	16.1	23.2	93	37	67	0.0	9.9	2.32	140	15	T = ฝนวัด
11	1017.1	1009.4	1013.90	32.5	15.9	23.3	91	34	67	0.0	9.8	4.94	190	9	จำนวนไม่ได้
12	1017.4	1010.4	1013.95	31.7	17.7	24.2	90	43	67	0.0	7.4	3.82	200	13	ต่ำกว่า 0.1 มม.
13	1016.2	1011.8	1014.18	29.4	21.6	25.0	83	51	68	T	3.3	1.73	170	15	
14	1017.4	1010.9	1014.18	30.1	19.6	24.5	89	48	69	0.0	3.2	2.55	220	13	
15	1016.0	1010.1	1013.25	30.9	20.5	25.1	85	46	69	0.8	7.2	3.49	150	19	
16	1016.3	1011.5	1013.69	27.9	21.3	23.9	88	60	77	0.2	1.7	2.31	340	26	
17	1018.0	1011.7	1014.83	30.1	20.0	24.1	94	48	73	0.0	8.4	2.69	10	22	

Data | ข้อมูลฯ ปี 65 | ข้อมูลฯ ปี64 | ข้อมูลฯ ปี63 | ข้อมูลฯ ปี62 | ข้อมูลฯ ปี 61

ภาพที่ 4.8 ข้อมูลสภาพอากาศที่ได้จากศูนย์อุตุนิยมวิทยาภาคเหนือ

4.3.2 การเปลี่ยนรูปข้อมูลให้เหมาะสม (Transform)

4.3.2.1 ทำการจัดเรียงข้อมูลที่ต้องการ

นำข้อมูล PM2.5 และ ข้อมูลตัวแปรสภาพอากาศ ที่คัดแยกออกมา ทำการรวมเข้าด้วยกันโดยใช้ตารางวันที่ เป็นตัวเชื่อมทั้งสองข้อมูล

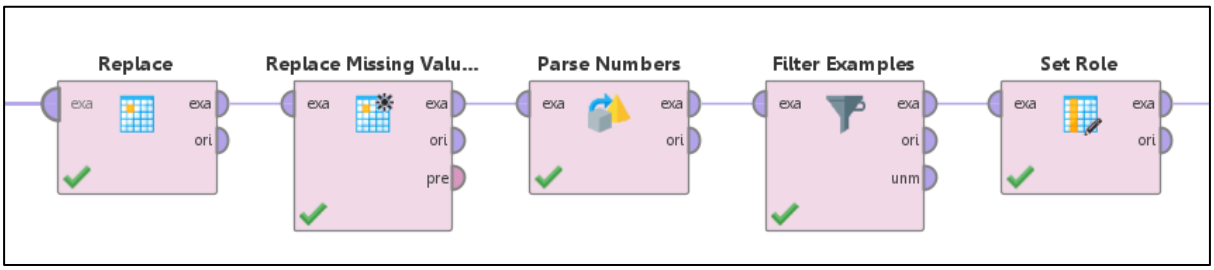
Date	AirPes	Humi	Temper	Rain	Wind	10	2.5
1/1/2018	1013.54	23.5	79	0.0	17	44	23.5
2/1/2018	1012.87	23.1	76	0.0	24	39.5	
3/1/2018	1012.28	23.5	74	0.3	17	41	22
4/1/2018	1011.77	24.2	77	0.0	17	44	25.5
5/1/2018	1011.57	23.7	80	0.0	15	37	22.5
6/1/2018	1010.11	24.3	81	0.0	9	51	34.5
7/1/2018	1010.14	25.9	77	0.0	24	55.5	35.5
8/1/2018	1011.39	26.5	74	0.0	11	50	30.5
9/1/2018	1012.21	27.1	74	0.0	13	41	24
10/1/2018	1013.05	26.9	72	0.0	22	38.5	22.5
11/1/2018	1014.28	25.4	72	0.0	17	48.5	33.5
12/1/2018	1018.75	19.3	75	T	19	26	
13/1/2018	1019.26	18.0	75	0.0	17	31.5	20
14/1/2018	1016.91	21.4	71	0.0	7	41.5	29

ภาพที่ 4.9 การรวมข้อมูล PM2.5 และข้อมูลตัวแปรสภาพอากาศ

4.3.2.2 ทำความสะอาดข้อมูล

1) นำเข้าข้อมูลสู่โปรแกรม Rapid miner และทำการแทนค่าข้อมูลที่หายไปด้วยค่าเฉลี่ย และ ทำการแปลงข้อมูลปริมาณน้ำฝนเป็นข้อมูลประเภทตัวเลข
2) ทำการแทนที่ข้อมูลในตัวแปร ปริมาณน้ำฝนที่เป็นข้อมูล T ซึ่งหมายถึงอุปกรณ์วัดข้อมูลในวันนั้นทำงานผิดพลาด โดยแทนที่ด้วยค่าว่าง
3) แปลงประเภทข้อมูลของตัวแปร ปริมาณน้ำฝน ให้เป็น Number
4) ทำการลบแถวที่มีค่าว่างออกทั้งหมด

5) ตั้งค่าตัวแปร วันที่ ให้เป็น ID และ ตั้งค่าตัวแปร PM2.5 ให้เป็น Label

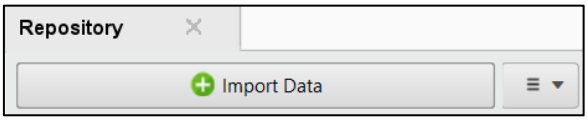


ภาพที่ 4.10 การทำความสะอาดข้อมูลโดยใช้โปรแกรม RapidMiner

4.3.3 การโหลดข้อมูล (Load)

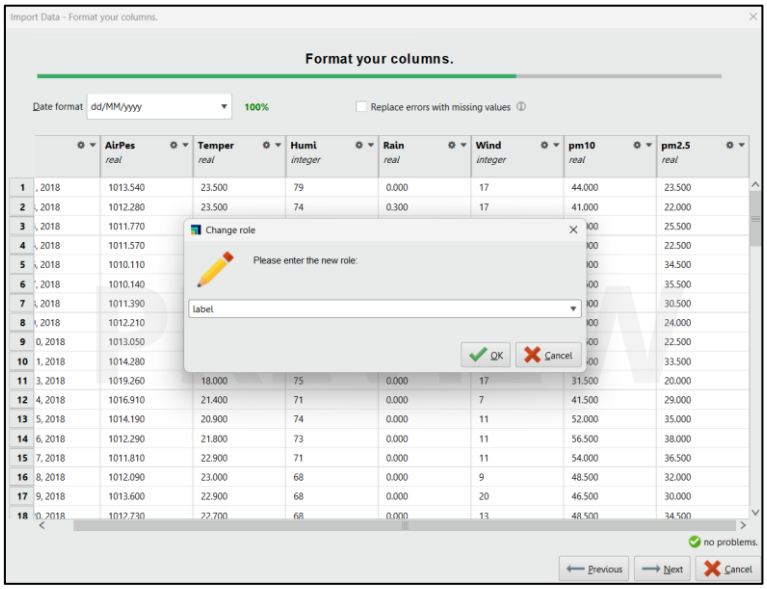
4.3.3.1 การโหลดข้อมูลเข้าสู่โปรแกรม RapidMiner

1) ใช้ปุ่ม Import Data เพื่อนำเข้าข้อมูลจากภายในเครื่อง



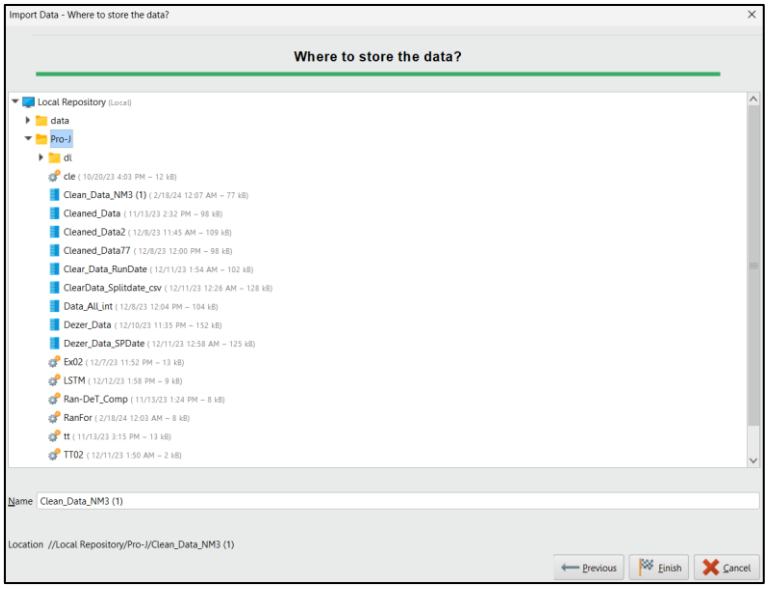
ภาพที่ 4.11 ปุ่ม Import Data

2) ทำการกำหนดค่าตัวแปร PM2.5 ให้เป็น Label



ภาพที่ 4.12 การกำหนดค่า Label

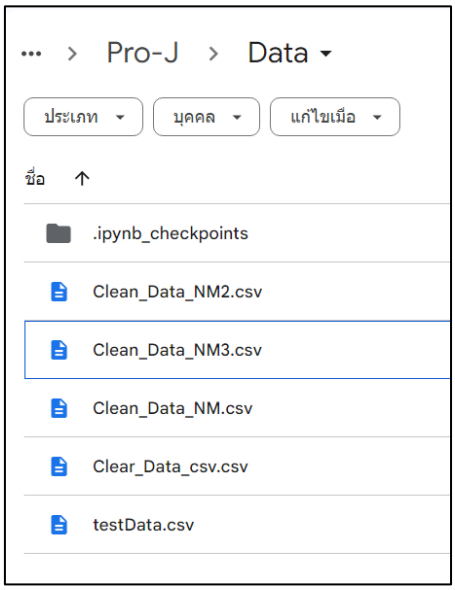
3) ทำการเลือกพื้นที่จัดเก็บภายในโปรแกรม และกด Finish



ภาพที่ 4.13 การเลือกพื้นที่จัดเก็บไฟล์ข้อมูล

4.3.3.2 การโหลดข้อมูลเข้าสู่เว็บไซต์ Google Drive

1) ทำการอัปโหลดข้อมูลสำหรับการทำนายไว้ในน Google Drive เพื่อความสะดวกในการใช้งาน



ภาพที่ 4.14 การอัปโหลดข้อมูลขึ้น Google Drive

4.4 การสร้างโมเดล (Modeling)

4.4.1 การสร้างโมเดลการทำนายโดยใช้โมเดล LSTM

1) การอ่านและการเตรียมข้อมูล : นำเข้าไลบรารีที่จำเป็น NumPy, Pandas, Matplotlib, และ TensorFlow สร้างฟังก์ชัน parse สำหรับแปลงข้อมูลวันที่ อ่านข้อมูล PM2.5 จากไฟล์ CSV และทำการ parse วันที่ด้วยฟังก์ชัน parse ตรวจสอบค่าว่าง, แทนที่ค่าว่าง, ลบข้อมูลที่ไม่จำเป็น ทำการ normalize ข้อมูล PM2.5 ให้อยู่ในช่วง 0–1 ด้วย Min–Max Scaler

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
from tensorflow.keras import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error

# ฟังก์ชันสำหรับแปลงข้อมูลวันที่
def parse(x):
    return datetime.strptime(x, '%d/%m/%Y')

# อ่านข้อมูล PM2.5 จากไฟล์ CSV
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Pro-J/Data/C

# ... (ตรวจสอบค่าว่างในข้อมูล, แทนที่ค่าว่าง, ลบข้อมูลที่ไม่จำเป็น)

# ทำการ normalize ข้อมูลให้อยู่ในช่วง 0-1
sc = MinMaxScaler()
data_sc = sc.fit_transform(df[['pm2.5']].values)
```

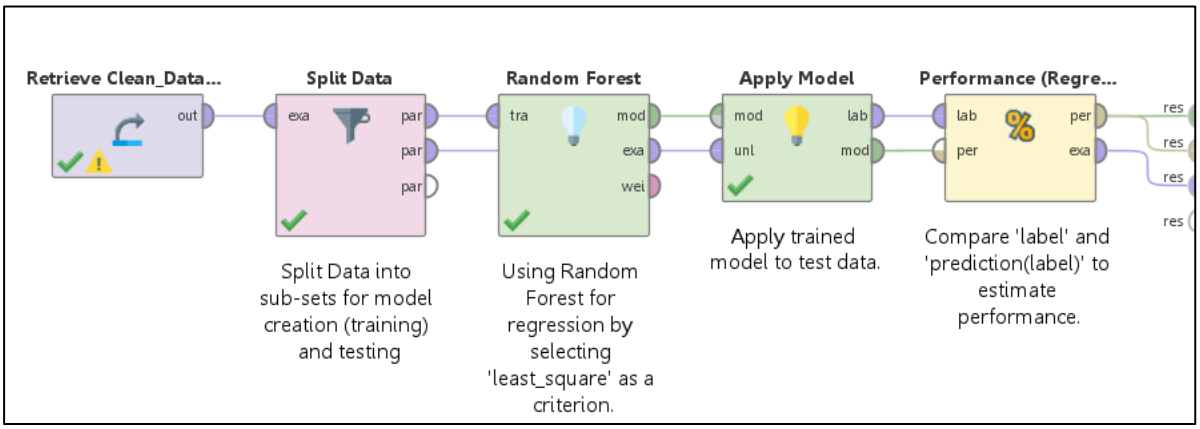
ภาพที่ 4.15 การนำเข้าไลบรารีที่จำเป็น และ การอ่าน เตรียมข้อมูล

2) การสร้างและคอมไพล์โมเดล LSTM : สร้างคลาส DeepModelTS ที่มีเมทอดต่าง ๆ สำหรับการสร้างข้อมูล, การคำนวณเมตริก, และการสร้างและคอมไพลโมเดล LSTM

```
class DeepModelTS():
    # ... (ตัวแปรและเมทอดต่าง ๆ สำหรับคลาส)

    def create_data_for_NN(self):
        # ... (สร้างข้อมูลสำหรับ Neural Network)

    def calculate_metrics(self, y_true, y_pred):
        # ... (คำนวณเมตริก)

    def LSTMModel(self):
        # ... (สร้างและคอมไพล์โมเดล LSTM)

# สร้างอินสแตนซ์ของคลาส
deep_learner = DeepModelTS(
    data=df,
    Y_var='pm2.5',
    lag=5,
    LSTM_layer_depth=50,
    epochs=20,
    batch_size=30,
    train_test_split=0.15
)
```

ภาพที่ 4.16 การสร้าง และ สอน โมเดล

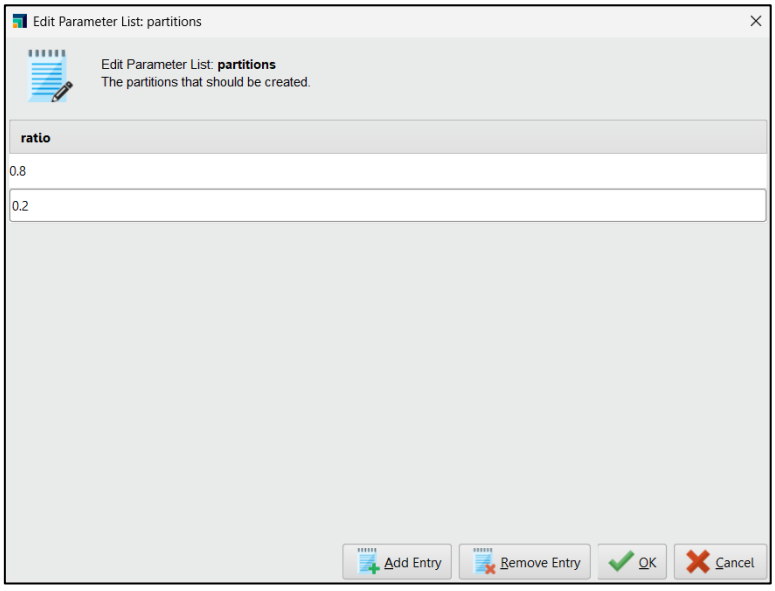
4.4.2 การสร้างโมเดลการทำนายโดยใช้โมเดล Random Forest



ภาพที่ 4.17 ขั้นตอนการสร้าง Model Random forest

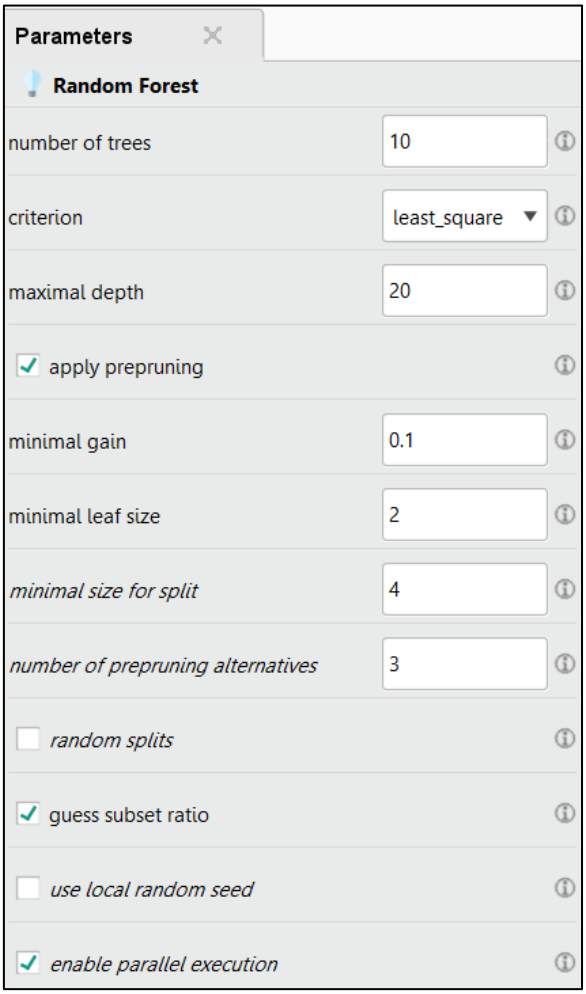
1) ทำการดึงข้อมูลที่ได้โหลดไว้แล้วด้วยตัวดำเนินการ retrieve

2) ทำการแบ่งข้อมูลออกเป็น 2 ชุดเพื่อนำไปสร้างชุดข้อมูลสอน และชุดข้อมูลทดสอบ โดยแบ่งอัตราส่วน 0.8:0.2 กฤติกา ทิพย์คำมี (2566)



ภาพที่ 4.18 การกำหนดอัตราส่วนของข้อมูลชุด Test และ ชุด Train

3) ทำการสร้างโมเดล Random Forest โดยทำการตั้งค่าตัวแปรดังภาพที่ 3.15



ภาพที่ 4.19 การตั้งค่า Parameter Random Forest

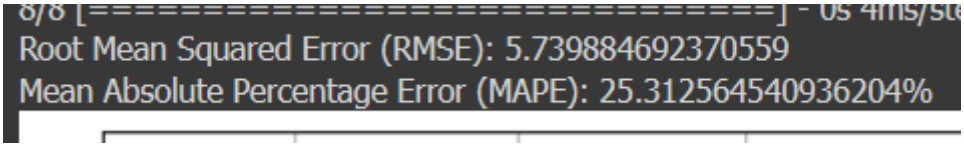
4.5 การวัดประสิทธิภาพของโมเดล (Evaluation)

4.5.1 การวัดประสิทธิภาพของโมเดล LSTM

ใช้โมเดลที่ได้จากการคอมไพล์เพื่อทำนายค่า PM2.5 คำนวณค่าความแม่นยำโดยใช้ค่า RMSE และ MAE

```
# ทำนายและคำนวณค่าความแม่นยำ
X_train, X_test, Y_train, Y_test = deep_learner.create_data_for_NN()
yhat = deep_learner.model.predict(X_test)
rmse, mae = deep_learner.calculate_metrics(Y_test.values, yhat.flatten())

print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'Mean Absolute Error (MAE): {mae}')
```

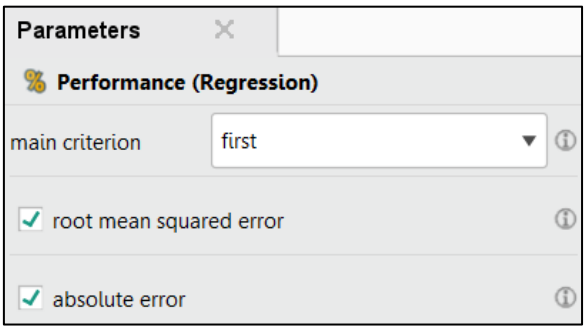
ภาพที่ 4.20 การวัดประสิทธิภาพโมเดล



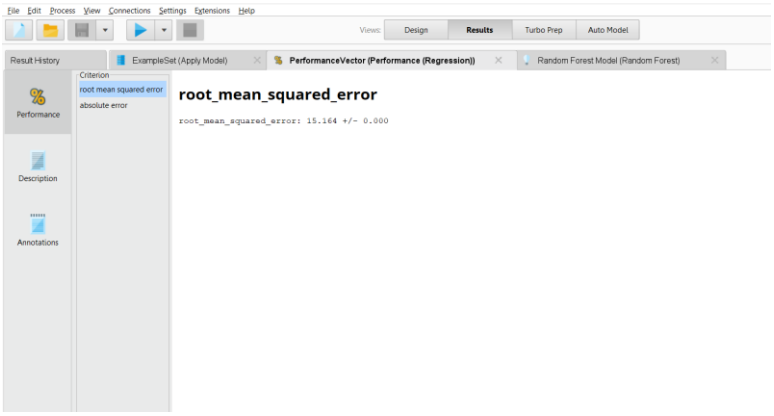
ภาพที่ 4.21 ประสิทธิภาพของโมเดลที่ได้จากการวัดผล

4.5.2 การวัดประสิทธิภาพของโมเดล Random Forest

ใช้โมเดลที่ได้จากการคอมไพล์เพื่อทำนายค่า PM2.5 คำนวณค่าความแม่นยำโดยใช้ค่า RMSE และ MAE



ภาพที่ 4.22 การตั้งค่า Parameter เพื่อทำการวัดผล



ภาพที่ 4.23 ภาพผลลัพธ์การวัดผลจากโปรแกรม RapidMiner

4.5.3 การเปรียบเทียบประสิทธิภาพระหว่าง โมเดล LSTM และ Random Forest

	RMSE	MAE
Random Forest	15.16	29.78
LSTM	5.73	25.31

ตารางที่ 4.1 ตารางเปรียบเทียบประสิทธิภาพของโมเดล

จากตาราง 4.1 ทำการเปรียบเทียบ 2 ส่วน

1) เปรียบเทียบกับค่าที่ยอมได้ ซึ่งค่า RMSE ต้องน้อยกว่า 7.00 และค่า MAE ต้องน้อยกว่า 27.00 กฤติกา ทิพย์คำมี และ คณะ (2566) พบว่าโมเดล Random Forest มีค่า RMSE และ MAE อยู่ที่ 15.16, 29.78 ตามลำดับ ซึ่งมากกว่าค่าที่ยอมรับได้ที่กำหนด และโมเดล LSTM มีค่า RMSE และ MAE อยู่ที่ 5.73, 25.31 ตามลำดับ ซึ่งต่ำกว่าค่าที่ยอมรับได้

2) เปรียบเทียบระหว่างโมเดล LSTM กับ โมเดล Random Forest โดยทำการเลือกโมเดลที่มีค่าน่าพึงพอใจที่สุด โมเดล Random Forest โดยมีค่า RMSE และ MAE อยู่ที่ 15.16, 29.78 ตามลำดับ ซึ่งโมเดล LSTM มีค่า RMSE และ MAE อยู่ที่ 5.73, 25.31 ตามลำดับ จากการเปรียบเทียบ โมเดล LSTM มีประสิทธิภาพในการทำนายมากกว่าโมเดล Random forest

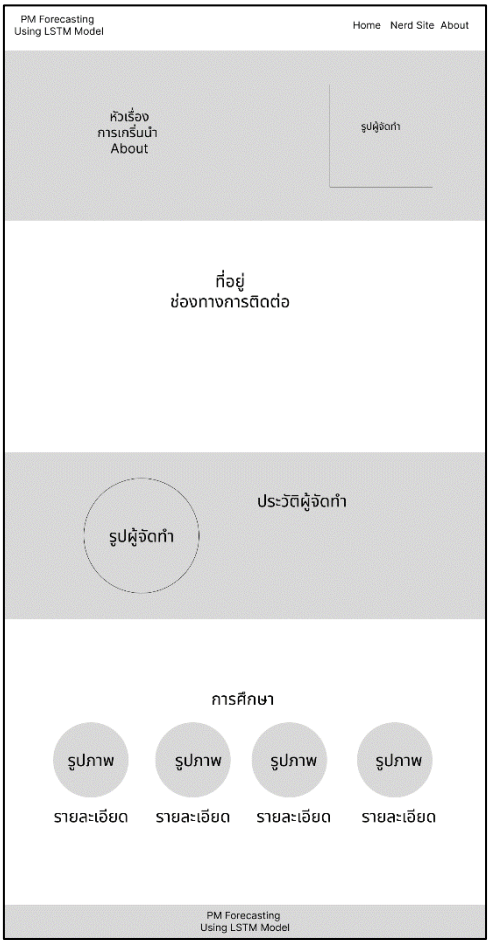
จากการเปรียบเทียบประสิทธิภาพทั้ง 2 ส่วนพบว่าโมเดล LSTM มีประสิทธิภาพในการทำนายค่าฝุ่น PM2.5 ได้แม่นยำมากกว่าโมเดล Random Forest

4.6 การนำโมเดลไปใช้งานจริง (Deployment)

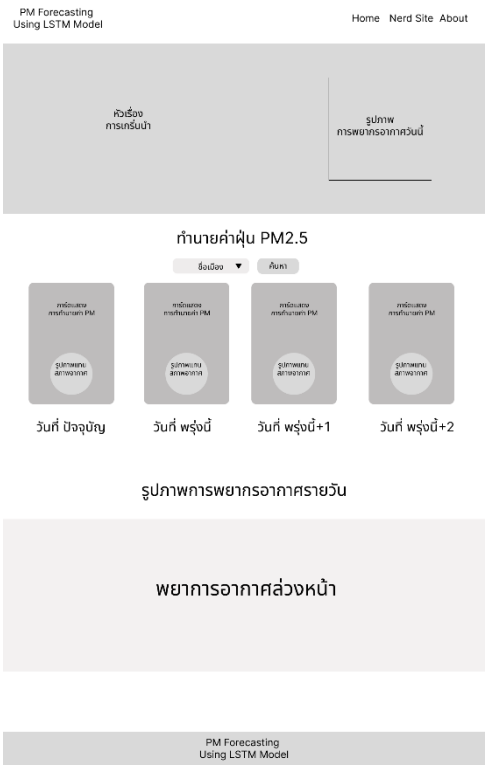
4.6.1 การสร้างหน้าเว็บไซต์สำหรับแสดงผล

กำหนดแผนการจัดลำดับ เนื้อหาสาระของเว็บไซต์ ออกแบบโครงหน้า UX/UI ของเว็บไซต์ทั้งหมด พัฒนาส่วนการใช้งานโมเดลและเว็บไซต์ส่วนแสดงผล ทำการอัปโหลดเว็บไซต์ขึ้นสู่ระบบออนไลน์ โดยมีขั้นตอนดังนี้

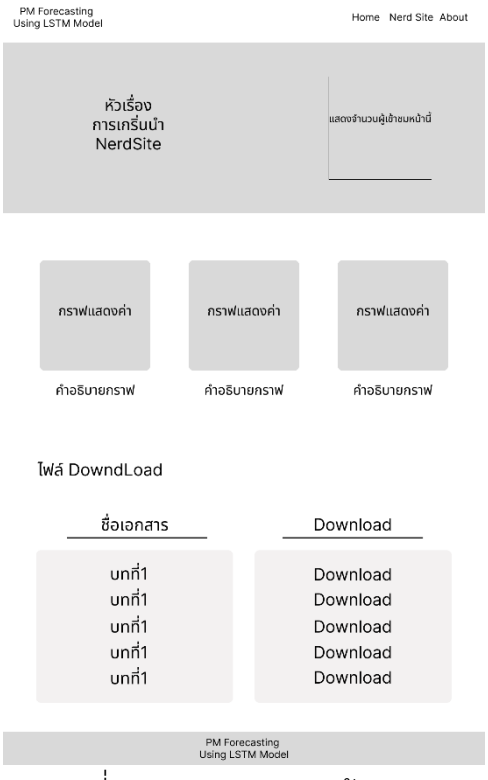
4.6.1.1 สร้างหน้าเว็บไซต์ โดยใช้โปรแกรม Nice Page ซึ่งเป็นโปรแกรมช่วยในการสร้างหน้าเพจ HTML และ CSS เบื้องต้นโดยทำการสร้างตาม Wireframe ที่ได้ออกแบบไว้



ภาพที่ 4.24 Wireframe หน้า Index



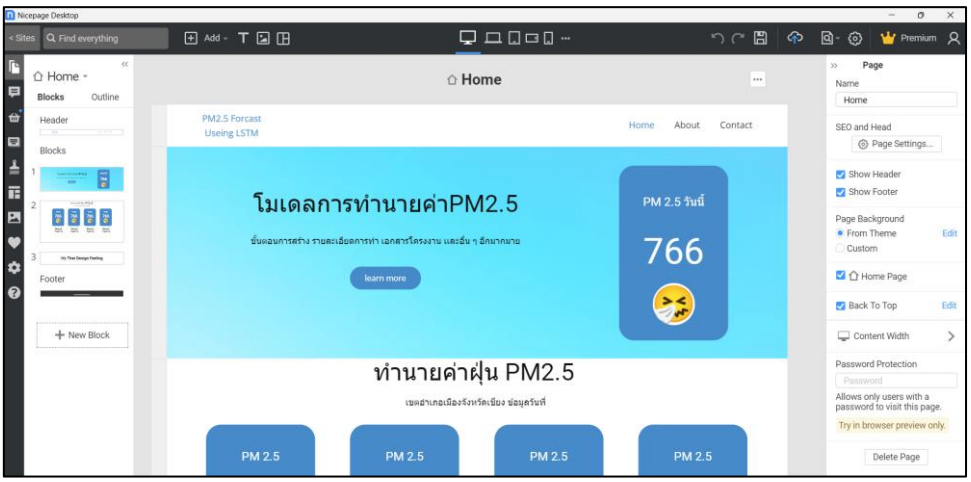
ภาพที่ 4.25 Wireframe หน้า NerdSite



ภาพที่ 4.26 Wireframe หน้า About

4.6.1.2 **จัดทำเว็บไซต์ตามที่ได้ออกแบบ**

ได้ใช้โปรแกรมสำเร็จรูปในการสร้างหน้า HTML พื้นฐานของหน้าเว็บไซต์ โดยใช้โปรแกรม Nice Page ในการสร้างหน้า HTML CSS ของเว็บไซต์



ภาพที่ 4.27 ภาพภายในโปรแกรม NicePage

4.6.1.3 **การสร้างส่วนการทำนายผลในหน้าเว็บไซต์**

1) สร้างฟังก์ชั่น Java script การทำงานของหน้าเว็บทำการกำหนด Api key และ ทำให้ฟังก์ชั่นเริ่มทำงานหลังหน้าเว็บโหลดเสร็จสมบูรณ์ และเรียกใช้ฟังก์ชั่น getCoordinates

```
let dateTime = [];
const apiKey = '058181e0ef5153da030c5d26cd04b2be';
document.addEventListener('DOMContentLoaded', () => {
    getCoordinates();
});
```

ภาพที่ 4.28 Code Java Script ทำการกำหนด Api key

2) ฟังก์ชั่น getCoordinates ทำการเรียกใช้ API จาก Open Weather Map และ เรียกใช้ฟังก์ชั่น fetchWeatherData

```
function getCoordinates() {
    const selectedCity = 'Chiang Mai';
    const apiUrl = `https://api.openweathermap.org/geo/1.0/direct?q=${selectedCity},TH&limit=1&appid=${apiKey}`;

    fetch(apiUrl)
        .then(response => response.json())
        .then(data => {
            const coordinates = data[0].lat + ', ' + data[0].lon;
            console.log('Coordinates:', coordinates);
            fetchWeatherData(data[0].lat, data[0].lon);
        })
        .catch(error => {
            console.error('Error fetching data:', error);
            alert('Error fetching data');
        });
}
```

ภาพที่ 4.29 Code Java Script ส่วนที่ getCoordinates

3) ฟังก์ชั่น fetchWeatherData ทำการรับค่า ละติจูด ลองติจูด จากฟังก์ชั่น getCoordinates และทำการเรียกใช้ API ของ Open Weather API เพื่อขอข้อมูลสภาพอากาศ จากนั้นทำการแยกเฉพาะข้อมูลที่ต้องการ และ เก็บข้อมูลลงในตัวแปร และ ทำการเรียกใช้ฟังก์ชั่น predict

```
function fetchWeatherData(lat, lon) {
    const weatherUrl = `https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lon}&appid=${apiKey}&cnt=32&units=metric`;

    fetch(weatherUrl)
        .then(response => response.json())
        .then(data => {
            const cityName = data.city.name;
            const filteredForecast = data.list.filter(entry => entry.dt_txt.includes(' 00:00:00'));

            const forecastData = filteredForecast.map(entry => {
                const { main, wind, rain, dt_txt } = entry;
                return {
                    dateTime: dt_txt,
                    atmosphericPressure: main.pressure,
                    temperature: main.temp,
                    humidity: main.humidity,
                    rainfall: rain ? rain['3h'] || 0 : 0,
                    windSpeed: wind.speed
                };
            });

            const weatherDataDiv = document.getElementById('weatherData');

            predict(forecastData);
        })
        .catch(error => {
            console.error('Error making weather forecast request:', error);
        });
}
```

ภาพที่ 4.30 Code Java Script ฟังก์ชั่น fetchWeatherData

4) ฟังก์ชั่น predict ทำการโหลดโมเดลจาก GitHub จากนั้นแปลงข้อมูลให้อยู่ในรูปแบบที่พร้อมสำหรับการทำนาย และ ทำการเรียกใช้ฟังก์ชั่นการทำนาย เก็บข้อมูลที่ในตัวแปร Prediction และทำการเรียกใช้ฟังก์ชั่น displayvalue

```
async function predict(weatherData) {
    try {
        const modelPath = 'https://raw.githubusercontent.com/8elllA8ell/Pro-3/90c7fb827f94b7b4641352d3f60e678b72fff464/model/model.json';
        const model = await tf.loadLayersModel(modelPath);

        let i = 0;
        predictedValue = [];

        weatherData.forEach(entry => {
            const { atmosphericPressure, temperature, humidity, rainfall, windSpeed } = entry;

            const inputData = tf.tensor3d([[
                [atmosphericPressure],
                [temperature],
                [humidity],
                [rainfall],
                [windSpeed],
                [0]
            ]]);

            const predictions = model.predict(inputData);

            predictedValue[i] = Array.from(predictions.dataSync());
            displayValues0();
            i++;
        });
    } catch (error) {
        console.error('Error loading the model:', error);
    }
}
```

ภาพที่ 4.31 Code Java Script ฟังก์ชั่น predict

5) ฟังก์ชั่น displayvalue ทำการรับค่า parameter และ ดึงข้อมูลในตัวแปร Predicvalue index ตามค่า parameter ที่รับมา จากนั้นทำการแปลงข้อมูลให้เป็นจำนวนเต็ม และ เก็บไว้ในตัวแปร intvalue ทำการกำหนด Element เป้าหมายจาก ID และทำการเปรียบเทียบค่า PM2.5 ในวันนั้น ๆ ว่ามีค่าอยู่ในระดับไหน ทำการเปลี่ยนข้อมูลและรูปภาพ ที่ปรากฏตามค่าของวันนั้น ๆ

```
function displayValues(valueIndex, containerIndex, imageIndex) {
  let intValue = Math.floor(predictedValue[valueIndex]/15);
  const resultContainer = document.getElementById(`myElement${containerIndex}`);
  const resultContainerimg = document.getElementById(`myimgElement${imageIndex}`);
  resultContainer.innerHTML = `<p>${intValue}</p>`;

  if (intValue >= 0 && intValue <= 25) {
      resultContainerimg.innerHTML = `<img class="u-image u-image-circle u-image-1" src="images/emoji_0.svg"
  } else if (intValue >= 26 && intValue <= 50) {
      resultContainerimg.innerHTML = `<img class="u-image u-image-circle u-image-1" src="images/emoji_1.svg"
  } else if (intValue >= 51 && intValue <= 90) {
      resultContainerimg.innerHTML = `<img class="u-image u-image-circle u-image-1" src="images/emoji_2.svg"
  } else if (intValue >= 51 && intValue <= 90) {
    resultContainerimg.innerHTML = `<img class="u-image u-image-circle u-image-1" src="images/emoji_3.svg" al
  } else {
      resultContainerimg.innerHTML = `<img class="u-image u-image-circle u-image-1" src="images/emoji_4.svg"
  }
```

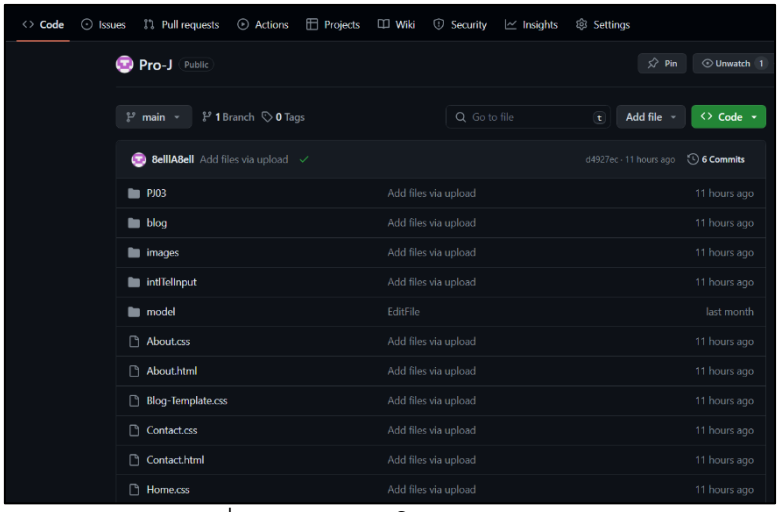
ภาพที่ 4.32 Code Java Script ฟังก์ชั่น displayvalue

6) อัปโหลด Website ทำการอัปโหลดไฟล์ Website ไปยัง Hosting โดยในโครงงานนี้เลือกใช้เป็น GitHub เนื่องจากเป็นบริการฟรี และรองรับการใช้งานได้อย่างครบถ้วน

4.6.3 การอัปโหลดเว็บไซต์และเผยแพร่

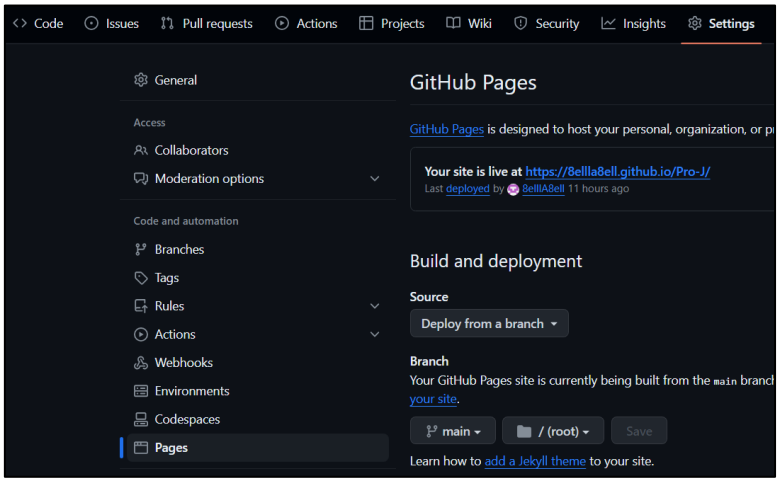
อัปโหลด Website ทำการอัปโหลดไฟล์ Website ไปยัง Hosting โดยในโครงงานนี้เลือกใช้เป็น GitHub เนื่องจากเป็นบริการฟรี และรองรับการใช้งานได้อย่างครบถ้วน

1) การอัปโหลดไฟล์ไปยังหน้า Project



ภาพที่ 4.33 การอัปโหลดไฟล์ไปยังหน้า Project

2) การตั้งค่าให้หน้าเว็บสามารถเผยแพร่ได้



ภาพที่ 4.34 การตั้งค่าให้หน้าเว็บสามารถเผยแพร่ได้