Machine learning project - Final Report

Dataset 4 Power Consumption

Group 5

1) Dechatorn	Bumrungchoo	Student ID 63070501207
2) Siravit	Tachaprosopchai	Student ID 63070501220
3) Chansinee	Mueangnu	Student ID 63070501221

To

Dr. Teema Leangarun

This report is part of the subject

INC 492/690 - Introduction to Data Science

Developed by Control Systems and Instrumentation Engineering Students

King Mongkut's University of Technology Thonburi

Semester 2/2022

Table of Contents

Chapter 1 About dataset

1.1 Dataset

Dataset ที่เลือกใช้ คือ Power Consumption ซึ่งเป็น Dataset ที่เกี่ยวกับการใช้พลังงานไฟฟ้าในพื้นที่ 3 แห่ง ของเมือง Tetouan ประเทศ Morocco โดยที่ข้อมูลจาก Dataset นี้จะเป็นข้อมูลในทุก ๆ 10 นาที โดยข้อมูลที่ส่งผลกับการใช้พลังงานไฟฟ้า ประกอบไปด้วย ข้อมูลที่สภาพอากาศ และข้อมูลการไหลของน้ำใต้พื้นดิน ซึ่งเราได้นำมาใช้ในการทำนายการใช้พลังงานไฟฟ้าของทั้ง 3 พื้นที่ รายละเอียดของปัจจัยที่ส่งผลต่อการใช้พลังงานไฟฟ้า มีรายละเอียดดังนี้

1.1.1 ข้อมูลสภาพอากาศ

ข้อมูลที่เป็นปัจจัยต่างๆ ที่ส่งผลต่อการใช้พลังงานไฟฟ้า ในส่วนของสภาพอากาศ ตั้งแต่อุณหภูมิ ความชื้น ความเร็วลม ซึ่งค่าทางสภาพอากาศเหล่านี้ถูกเก็บจากระบบ SCADA (Supervisory Control And Data Acquisition System)

1.1.2 ข้อมูลการไหลของน้ำใต้พื้นดิน

เป็นข้อมูลที่เป็นปัจจัยในการวิเคราะห์การใช้พลังงานไฟฟ้าเช่นเดียวกันกับสภาพอากาศ ประกอบไปด้วย general diffuse flows และ diffuse flow ซึ่งเป็นข้อมูลที่ใช้วิเคราะห์การไหลของน้ำใต้ดิน เนื่องจากสภาพอากาศมีผลต่อปริมาณน้ำที่เข้าสู่ระบบน้ำใต้ดิน ตัวอย่างเช่น ฝนตกหนักจะมีผลให้ปริมาณของน้ำใต้ดินเพิ่มสูงขึ้น และในช่วงหน้าร้อนปริมาณของน้ำใต้ดินจะลดลง

1.2 Objective

เพื่อคาดการณ์การใช้พลังงานไฟฟ้าของเมือง Tetouan ประเทศ Morocco และนำไปวิเคราะห์การใช้พลังงานและการจัดการการผลิตพลังงานไฟฟ้าที่ยั่งยืนและมีประสิทธิภาพที่สุด

Chapter 2 Exploratory Data Analysis (EDA)

2.1 Basic information about data

Dataset Power Consumption นี้จะเป็นข้อมูลในทุก ๆ 10 นาที ประกอบไปด้วย ข้อมูลการใช้พลังงานไฟฟ้าของทั้ง 3 พื้นที่ในเมือง Tetouan และข้อมูลที่เป็นปัจจัยต่าง ๆ ซึ่งส่งผลต่อการใช้พลังงานไฟฟ้าเป็นไปดัง Figure 2.1

	DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption
0	2017-01-01 00:00:00	6.559	73.8	0.083	0.051	0.119	34055.69620	16128.87538	20240.96386
1	2017-01-01 00:10:00	6.414	74.5	0.083	0.070	0.085	29814.68354	19375.07599	20131.08434
2	2017-01-01 00:20:00	6.313	74.5	0.080	0.062	0.100	29128.10127	19006.68693	19668.43373
3	2017-01-01 00:30:00	6.121	75.0	0.083	0.091	0.096	28228.86076	18361.09422	18899.27711
4	2017-01-01 00:40:00	5.921	75.7	0.081	0.048	0.085	27335.69620	17872.34043	18442.40964
5	2017-01-01 00:50:00	5.853	76.9	0.081	0.059	0.108	26624.81013	17416.41337	18130.12048
6	2017-01-01 01:00:00	5.641	77.7	0.080	0.048	0.096	25998.98734	16993.31307	17945.06024
7	2017-01-01 01:10:00	5.496	78.2	0.085	0.055	0.093	25446.07595	16661.39818	17459.27711
8	2017-01-01 01:20:00	5.678	78.1	0.081	0.066	0.141	24777.72152	16227.35562	17025.54217
9	2017-01-01 01:30:00	5.491	77.3	0.082	0.062	0.111	24279.49367	15939.20973	16794.21687

Figure 2.1 ข้อมูล 10 แถวแรกของ Dataset Power Consumption

ข้อมูลสรุปเบื้องต้นและค่าสถิติพื้นฐาน ได้แก่ จำนวนข้อมูลทั้งหมด ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน ค่าควอไทล์ ค่าสูงสุด และค่าต่ำสุด แสดงได้ดัง Figure 2.2

	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption
count	52416.000000	52416.000000	52416.000000	52416.000000	52416.000000	52416.000000	52416.000000	52416.000000
mean	18.810024	68.259518	1.959489	182.696614	75.028022	32344.970564	21042.509082	17835.406218
std	5.815476	15.551177	2.348862	264.400960	124.210949	7130.562564	5201.465892	6622.165099
min	3.247000	11.340000	0.050000	0.004000	0.011000	13895.696200	8560.081466	5935.174070
25%	14.410000	58.310000	0.078000	0.062000	0.122000	26310.668692	16980.766032	13129.326630
50%	18.780000	69.860000	0.086000	5.035500	4.456000	32265.920340	20823.168405	16415.117470
75%	22.890000	81.400000	4.915000	319.600000	101.000000	37309.018185	24713.717520	21624.100420
max	40.010000	94.800000	6.483000	1163.000000	936.000000	52204.395120	37408.860760	47598.326360

Figure 2.2 ข้อมูลสถิติเบื้องต้น

Figure 2.3 เป็นข้อมูลสรุปเกี่ยวกับ Dataset Power Consumption ประกอบไปด้วย จำนวนแถว จำนวนคอลัมน์ ชื่อคอลัมน์ ชนิดข้อมูลของแต่ละคอลัมน์ จำนวนค่าไม่ว่าง (non-null) และการใช้หน่วยความจำของ DataFrame

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52416 entries, 0 to 52415
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   DateTime               52416 non-null  datetime64[ns]
 1   Temperature            52416 non-null  float64
 2   Humidity               52416 non-null  float64
 3   Wind Speed             52416 non-null  float64
 4   general diffuse flows  52416 non-null  float64
 5   diffuse flows          52416 non-null  float64
 6   Zone 1 Power Consumption  52416 non-null  float64
 7   Zone 2  Power Consumption  52416 non-null  float64
 8   Zone 3  Power Consumption  52416 non-null  float64
dtypes: datetime64[ns](1), float64(8)
memory usage: 3.6 MB
```

Figure 2.3 ข้อมูลสรุปเกี่ยวกับ Dataset

2.2 Data Visualization

เนื่องจากชุดข้อมูลของเราแสดงอยู่ในรูปตัวเลข ซึ่งยากต่อการสรุปและทำความเข้าใจ เราจึงได้ใช้วิธีนำข้อมูลมาแสดงผลในรูปแบบกราฟฟิก และกราฟแทน ดังนี้

2.2.1 แนวโน้มความสัมพันธ์ของตัวแปรต่าง ๆ ใน Dataset

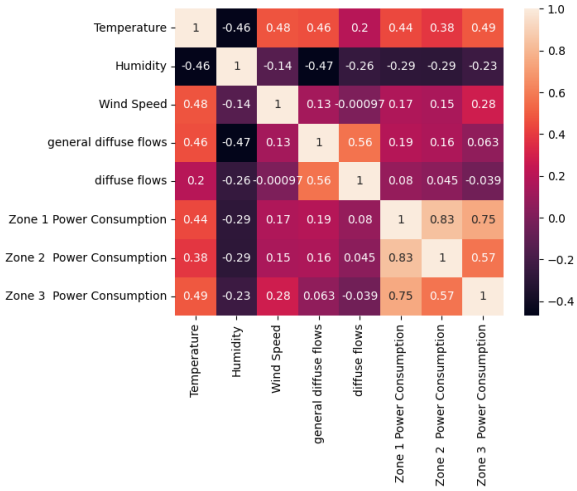

Figure 2.4 ตารางค่าสัมประสิทธิ์สหพันธ์

เนื่องจากเราสนใจที่จะทำนายการใช้พลังงานไฟฟ้า (Zone1-3 Power Consumption) หากพิจารณาจาก Figure 2.4 ในภาพรวม จะเห็นได้ว่า Temperature มีความสัมพันธ์เชิงบวกกับ Power Consumption มากที่สุด อาจประเมินได้คร่าว ๆ ว่าหากสภาพอากาศมี Temperature มาก (อากาศร้อน) ก็จะส่งผลให้มีการใช้พลังงานไฟฟ้าที่มากขึ้นตาม ดัง Figure 2.5 ส่วน Humidity มีความสัมพันธ์เชิงลบกับ Power Consumption มากที่สุดเช่นกัน และอาจประเมินได้คร่าว ๆ ว่าหากสภาพอากาศมี Humidity น้อย จะส่งผลให้มีการใช้พลังงานไฟฟ้าที่มากขึ้นตาม Figure 2.6

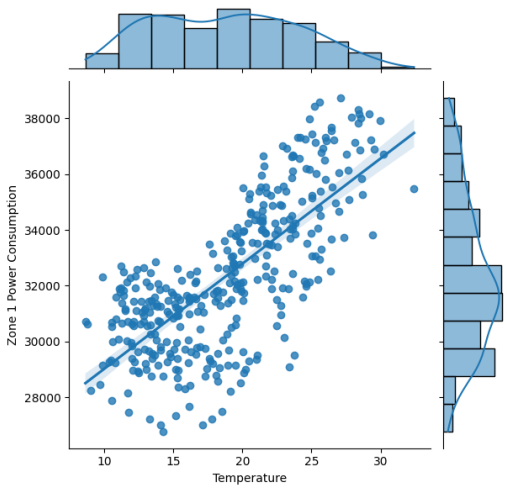

Figure 2.5 ความสัมพันธ์ระหว่าง Temperature กับ Power Consumption

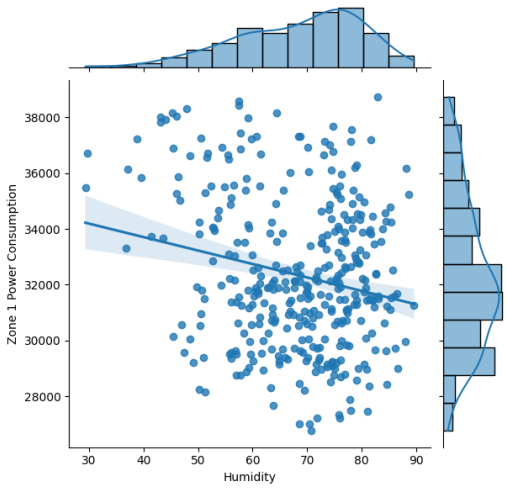

Figure 2.6 ความสัมพันธ์ระหว่าง Humidity กับ Power Consumption

2.2.2 เปรียบการใช้พลังงานไฟฟ้าของแต่ละพื้นที่

การที่จะสร้างแบบจำลองเพื่อคาดการณ์การใช้พลังงานไฟฟ้าของแต่ละพื้นที่ จำเป็นต้องศึกษาลักษณะการใช้พลังงานไฟฟ้าของพื้นที่นั้น ๆ เพื่อให้สามารถเลือกหาตัวแปรที่มีความสำคัญในการนำไปคาดการณ์การใช้พลังงานไฟฟ้าได้อย่างแม่นยำ ดังนั้นเราจึงศึกษาข้อมูลการใช้พลังงานไฟฟ้าในช่วงเวลาต่าง ๆ ดังนี้

1) ข้อมูลการใช้พลังงานไฟฟ้าทุก ๆ 10 นาที ใน 1 ปี

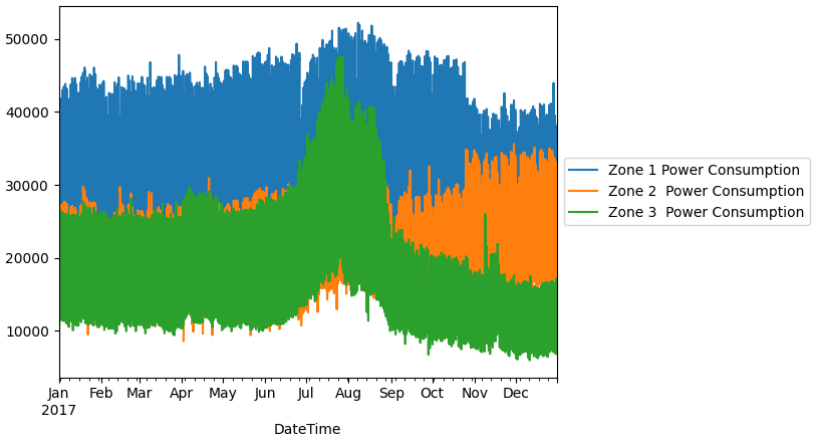

Figure 2.7 การใช้พลังงานไฟฟ้าของแต่ละพื้นที่ (ข้อมูลทุก ๆ 10 นาที)

การใช้พลังงานไฟฟ้าของแต่ละพื้นที่เป็นระยะเวลา 1 ปี แสดงได้ดัง Figure 2.4 ซึ่งเป็นข้อมูลทุก ๆ 10 นาที จะเห็นได้คร่าว ๆ ว่าใน Zone 1 และ Zone 3 มีการใช้พลังงานไฟฟ้าสูงที่สุดในช่วงเดือนกรกฎาคม ถึงเดือนกันยายน

2) ข้อมูลการใช้พลังงานไฟฟ้าเฉลี่ยรายวัน ใน 1 ปี

เนื่องจาก Figure 2.4 เป็นกราฟที่พลอทโดยข้อมูลทุก ๆ 10 นาที ทำให้เห็นภาพไม่ชัดเจน (มีการทับกันของกราฟระหว่าง Zone 2 และ Zone 3) เราจึงได้เฉลี่ยการใช้พลังงานงานในแต่ละวันขึ้นมาใหม่ ตาม Figure 2.5

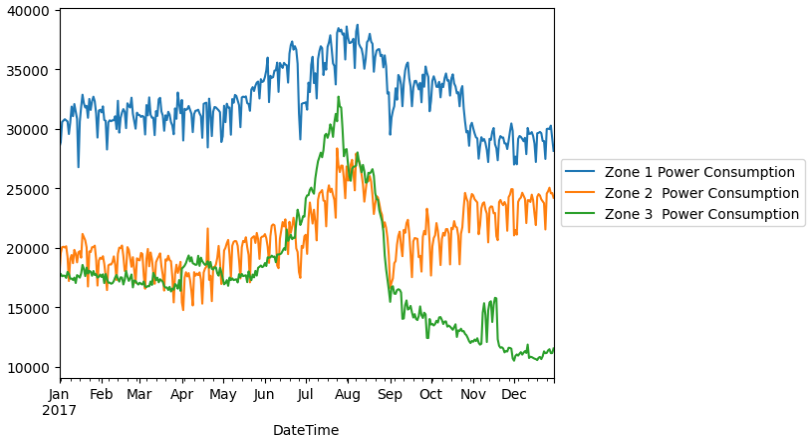

Figure 2.8 การใช้พลังงานไฟฟ้าของแต่ละพื้นที่ (ข้อมูลเฉลี่ยรายวัน)

จาก Figure 2.5 จะเห็นได้ว่าช่วงเดือนกรกฎาคม ถึงเดือนกันยายน เป็นช่วงที่มีการใช้พลังงานไฟฟ้ามากที่สุดในรอบปีของพื้นที่ทั้ง 3 Zone ซึ่งอาจสันนิษฐานได้ว่าช่วง 3 เดือนนี้เป็นช่วง summer ของที่นั่น ส่งผลให้อากาศร้อนมากขึ้นกว่าปกติ ก็เลยมีการใช้พลังงานไฟฟ้าที่มากขึ้นตามไปด้วย

3) ข้อมูลการใช้พลังงานไฟฟ้าเฉลี่ยแต่ละวันของสัปดาห์ ใน 1 ปี

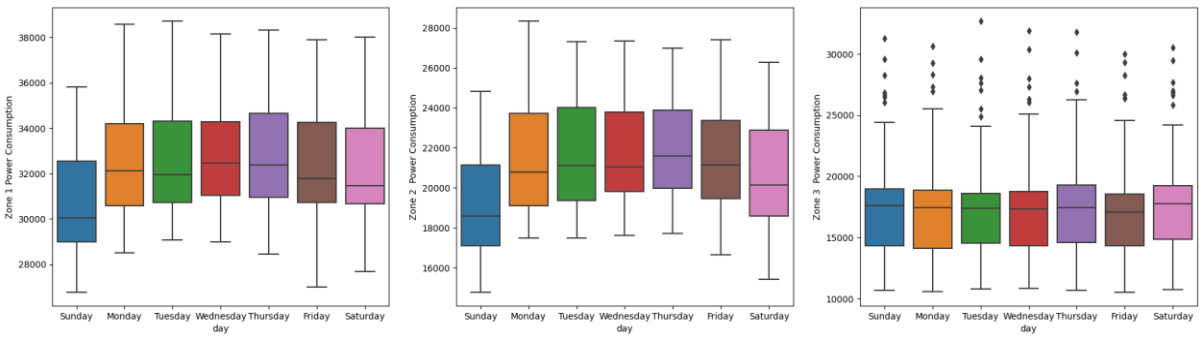

Figure 2.9 การใช้พลังงานไฟฟ้าของแต่ละพื้นที่ (ข้อมูลเฉลี่ยรายวัน)

Figure 2.6 เป็น Boxplot เปรียบเทียบการใช้พลังงานไฟฟ้ารายสัปดาห์ของแต่ละ Zone จะเห็นได้ว่า Zone 1 และ Zone 2 มีการใช้พลังงานไฟฟ้าเฉลี่ยของวันจันทร์ถึงวันศุกร์ (สันนิษฐานว่าเป็นวันทำงาน) เยอะกว่าวันหยุดเสาร์อาทิตย์อย่างชัดเจน นั่นคือ Zone 1 และ Zone 2 อาจเป็นพื้นที่ที่มีการตั้งอยู่ของภาคครัวเรือนและภาคโรงงานรวมกัน ทำให้วันทำงานคนส่วนใหญ่ก็ไปใช้ไฟฟ้าจากที่ทำงาน แต่เมื่อถึงวันหยุด ผู้คนส่วนใหญ่

อยู่บ้าน แน่นอนว่าก็จะต้องพยายามใช้ไฟฟ้าให้ลดน้อยลง ส่วน zoneที่ 3 การใช้พลังงานไฟฟ้าเฉลี่ยของวันทำงานกับวันหยุดแทบจะไม่แตกต่างกัน อาจเป็นผลมาจากที่นั่นมีการตั้งอยู่ของภาคครัวเรือนเป็นส่วนใหญ่ การใช้ไฟฟ้าของวันทำงานกับวันหยุดก็เลยไม่แตกต่างกันอย่างชัดเจนเหมือน Zone 1 และ Zone2 อีกทั้งยังมีการใช้พลังงานไฟฟ้าโดยภาพรวมน้อยกว่า Zone 1 และ Zone 2

4) ข้อมูลการใช้พลังงานไฟฟ้าเฉลี่ยรายชั่วโมง ใน 1 ปี

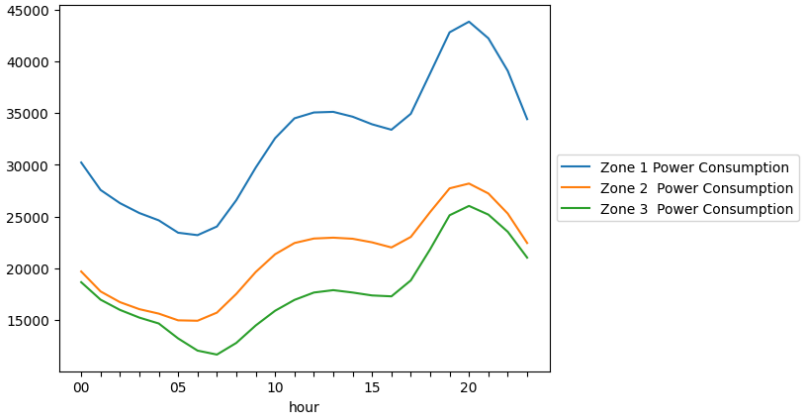

Figure 2.10 การใช้พลังงานไฟฟ้าของแต่ละพื้นที่ (ข้อมูลเฉลี่ยรายชั่วโมง)

จาก Figure 2.7 เป็นการใช้พลังงานไฟฟ้าโดยเฉลี่ยเป็นรายชั่วโมงในทั้งปี ก็เห็นได้ว่าตั้งแต่ช่วงประมาณ 7 โมงเช้าเป็นต้นไป จะมีการใช้พลังงานไฟฟ้าเพิ่มขึ้นเรื่อย ๆ ตลอดทั้งวัน และจะมีการใช้พลังงานไฟฟ้าสูงเพิ่มขึ้นอย่างรวดเร็วตั้งแต่ช่วงประมาณ 5 โมงเย็น ถึง 2 ทุ่ม อาจสันนิษฐานได้ว่าเป็นช่วงเวลาพักจากการทำงาน มีการใช้อุปกรณ์ไฟฟ้าในเกือบทุกครัวเรือน ส่งผลให้มีการใช้พลังงานไฟฟ้าในภาพรวมที่มากขึ้น

2.3) Fixing missing data OR Data normalization

จาก Figure 2.3 จะเห็นได้ว่าข้อมูลไม่มี missing data และจาก Figure 2.1 ถ้าหากจะสร้าง model เพื่อคาดการณ์การใช้พลังงานไฟฟ้าจะมี Feature ต่าง ๆ ที่เกี่ยวข้อง 5 Features ได้แก่ อุณหภูมิ ความชื้น ความเร็วลม General diffuse flow และ Diffuse flow

จากการทำ Exploratory Data Analysis ข้างต้น พบว่ามี Feature ที่ส่งผลกับการใช้พลังงานไฟฟ้าเพิ่มเติม คือ ชั่วโมงในแต่ละวัน (00-23) และวันในแต่ละสัปดาห์ (Monday-Sunday) จึงได้เพิ่มทั้ง 2 Features นี้เข้ามาในข้อมูล ดังแสดงดัง Figure 2.8

DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption	hour	day_of_week
2017-01-01 00:00:00	6.559	73.8	0.083	0.051	0.119	34055.69620	16128.87538	20240.96386	0	0
2017-01-01 00:10:00	6.414	74.5	0.083	0.070	0.085	29814.68354	19375.07599	20131.08434	0	0
2017-01-01 00:20:00	6.313	74.5	0.080	0.062	0.100	29128.10127	19006.68693	19668.43373	0	0
2017-01-01 00:30:00	6.121	75.0	0.083	0.091	0.096	28228.86076	18361.09422	18899.27711	0	0
2017-01-01 00:40:00	5.921	75.7	0.081	0.048	0.085	27335.69620	17872.34043	18442.40964	0	0
...	...	...	...	...	...	...	...	...	...	...
2017-12-30 23:10:00	7.010	72.4	0.080	0.040	0.096	31160.45627	26857.31820	14780.31212	23	6
2017-12-30 23:20:00	6.947	72.6	0.082	0.051	0.093	30430.41825	26124.57809	14428.81152	23	6
2017-12-30 23:30:00	6.900	72.8	0.086	0.084	0.074	29590.87452	25277.69254	13806.48259	23	6
2017-12-30 23:40:00	6.758	73.0	0.080	0.066	0.089	28958.17490	24692.23688	13512.60504	23	6
2017-12-30 23:50:00	6.580	74.1	0.081	0.062	0.111	28349.80989	24055.23167	13345.49820	23	6

52416 rows × 10 columns

Figure 2.11 DataFrame ที่มีการเพิ่ม 2 Feature เข้ามา

จาก Figure 2.8 ก็จะเห็นได้ว่า มี Features ทั้งหมด 7 Features ซึ่งในข้อมูลอาจมีคอลัมน์ที่เกินความจำเป็นหรือมีการซ้ำซ้อน ส่งผลเสียคือ ใช้ทรัพยากรเครื่องสูง ใช้เวลา Train นาน ดังนั้นเราจึงพยายามลดมิติของ Features ลง แต่ยังคงให้ Information เดิมของชุดข้อมูลให้มากที่สุด โดยใช้เทคนิค PCA ซึ่งมีหลักการ คือ ทำให้มิติของ Features น้อยที่สุด แต่สามารถใช้เป็นตัวแทนหรือให้ Information ของข้อมูลต้นฉบับมากที่สุด

ซึ่งการทำ PCA ที่ให้ความสำคัญกับตัวแปรที่มีค่าความแปรปรวนสูง ๆ ก่อน ดังนั้น เพื่อไม่ให้สเกลของข้อมูลที่ห่างกันมากมีผลกับกระบวนการ PCA จึงต้องเปลี่ยนให้ข้อมูลทุกคอลัมน์มาเป็นสเกลเดียวกันด้วยการทำ Standard Scale (Z-Score) ซึ่งจะทำให้ข้อมูลทุกคอลัมน์มีค่าเฉลี่ยเป็น 0 และค่าส่วนเบี่ยงเบนมาตรฐานเป็น 1

2.4) Selected input features

ในที่นี้เราเลือกจำนวน PCA จากค่า Cumulative Explained Variance 89% (มีความสามารถเป็นตัวแทนข้อมูลจริงได้ 89%) ดัง Figure 2.9 ทำให้เราเลือกจำนวน PCA ทั้งหมด 5 ตัว (ใช้ PCA1-5) เท่ากับว่าลด Features ลงไปได้ถึง 28.57% (จาก 7 Features เหลือ 5 Features)

```
pca = PCA(0.89) #ต้องการ 89%
pca.fit(X_sc) #ทำการคำนวณ PCA
```

```
PCA(n_components=0.89)
```

```
pca.n_components_  #จะต้องใช้กี่ component ผลลัพธ์คือ 5
```

```
5
```

Figure 2.12 เลือก Components ในการทำ PCA

สรุปได้ว่า Features ที่เลือกใช้คือ General diffuse flows Diffuse flows Humidity Hour และ Temperature ซึ่งเลือกจากการทำ PCA โดยที่ PCA1 (General diffuse flows) จะมี variance เยอะที่สุด

Chapter 3 Modelling

3.1) Selected model

ตัดสินใจใช้ RDF เป็นวิธีการสร้างโมเดล เนื่องจากชุดข้อมูลของเรามีความซับซ้อนและหลากหลาย Features ซึ่ง RDF สามารถจัดบความสำคัญของ Features เพื่อนำไปใช้ ในการตัดสินใจเมื่อนำไปทำนาย อีกทั้งยังป้องกันการ overfitting และมีความแม่นยำสูง เนื่องจากมีการใช้เทคนิคการสุ่มตัวอย่างและการสุ่มคุณลักษณะในแต่ละต้นไม้การตัดสินใจ

โดยจากการที่วิเคราะและจัดการกับข้อมูล เราจะทำการสร้างโมเดลและทำการทดสอบเพื่อเปรียบเทียบประสิทธิภาพระหว่างโมเดล 3 โมเดล คือ
โมเดลที่ 1 จะสร้างโมเดลที่ใช้เพียงตัวแปรจากข้อมูลสภาพอากาศและข้อมูลการไหลของน้ำใต้พื้นดิน
แบบจำลองที่ 2 จะสร้างโมเดลที่ใช้ตัวแปรของชั่วโมงในแต่ละวันมาเป็นตัวแปรสำหรับเงื่อนไขในการคาดการณ์ร่วมด้วย
แบบจำลองที่ 3 จะสร้างโมเดลโดยใช้ตัวแปรทั้งหมดในชุดเดียวกันกับแบบจำลองที่ 2 โดยจะมีการใช้วิธีการลดมิติด้วยองค์ประกอบหลัก (Principle Components Analysis) เพื่อวิเคราะห์หาองค์ประกอบหลักในการนำมาใช้เป็นตัวแปรเพื่อเพิ่มประสิทธิภาพในการคาดการณ์การใช้พลังงาน

3.2) Training the Model

สำหรับการสร้างโมเดลนั้น จะต้องทำการ train model เพื่อให้โมเดลสามารถเรียนรู้และทำนายผลลัพธ์ของข้อมูลได้ถูกต้อง ซึ่งจะเริ่มต้นจากการแบ่งข้อมูลเป็นชุด train และ ชุด test

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.30)
X2_train, X2_test, y2_train, y2_test = train_test_split(X1, y2, test_size=0.30)
X3_train, X3_test, y3_train, y3_test = train_test_split(X1, y3, test_size=0.30)
```

Figure 3.1 ตัวอย่างการทำ the train-test split เพื่อแบ่งข้อมูล

โดยจะแบ่งสัดส่วนระหว่างข้อมูลชุด train และ ชุด test คือ 0.3 ดัง Figure 3.1 ซึ่งจะได้ผลลัพธ์ดังนี้

DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows
2017-09-07 04:50:00	22.13	89.60	0.331	0.069	0.119
2017-09-09 15:00:00	26.39	59.55	0.261	682.400	52.390
2017-04-24 14:50:00	22.98	42.91	0.090	550.700	346.500
2017-12-24 23:50:00	9.97	85.00	0.085	0.059	0.152
2017-07-14 09:20:00	25.29	76.20	4.913	469.800	183.800
...	...	...	...	...	...
2017-02-05 19:30:00	8.86	53.09	0.088	0.044	0.115
2017-05-28 17:40:00	27.06	44.24	0.083	417.800	214.100
2017-12-18 17:00:00	17.63	69.03	0.077	111.500	133.100
2017-11-15 17:20:00	22.41	48.17	0.086	68.530	74.100
2017-03-02 11:20:00	16.39	76.30	0.080	569.800	61.280

36691 rows × 5 columns

Figure 3.2 ตัวอย่างข้อมูลชุด X_train

```
DateTime
2017-04-12 08:50:00    29023.20775
2017-12-01 19:50:00    35193.91635
2017-12-25 07:50:00    23002.28137
2017-12-11 19:20:00    39872.24335
2017-08-08 07:30:00    31005.54939
                          ...
2017-02-25 04:20:00    21813.55932
2017-01-16 00:10:00    28076.96203
2017-02-03 22:10:00    38324.74576
2017-07-28 21:30:00    48688.90365
2017-06-28 19:00:00    33313.90728
Name: Zone 1 Power Consumption, Length: 36691, dtype: float64
```

Figure 3.3 ตัวอย่างข้อมูลชุด y_train

DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows
2017-09-17 22:50:00	21.00	78.20	4.917	0.091	0.093
2017-09-09 01:00:00	21.32	87.50	0.360	0.073	0.133
2017-03-30 18:10:00	24.71	38.30	4.924	206.000	231.400
2017-12-11 16:20:00	16.07	69.14	0.075	144.800	140.200
2017-06-03 23:10:00	20.39	83.80	0.065	0.040	0.152
...	...	...	...	...	...
2017-06-10 20:50:00	21.02	60.32	0.079	0.326	0.322
2017-04-14 16:00:00	18.42	70.20	0.073	116.800	102.300
2017-03-04 14:00:00	14.99	61.85	4.921	302.400	295.500
2017-04-23 18:30:00	19.97	60.13	0.086	79.500	66.700
2017-06-13 00:00:00	18.76	57.12	0.081	0.051	0.078

15725 rows × 5 columns

Figure 3.4 ตัวอย่างข้อมูลชุด X_train

```
DateTime
2017-12-19 16:30:00    31531.55894
2017-04-21 06:40:00    22370.37675
2017-06-24 15:30:00    37153.90728
2017-09-06 02:30:00    27270.79646
2017-12-09 23:20:00    30831.93916
                          ...
2017-10-26 11:10:00    33028.62144
2017-08-11 03:30:00    29656.64817
2017-11-26 09:40:00    25286.15385
2017-05-13 00:20:00    29725.37705
2017-08-28 15:40:00    38402.13097
Name: Zone 1 Power Consumption, Length: 15725, dtype: float64
```

Figure 3.5 ตัวอย่างข้อมูลชุด y_test

หลังจากนั้นจะต้องทำการ train model จากโมเดลที่เราทำการเลือกไว้ ซึ่งก็คือ ทำการสร้าง Random Forest model โดยจะทำการกำหนดจำนวนต้นไม้อยู่ 100 ต้น

```
regr1 = RandomForestRegressor(n_estimators=100, bootstrap=True)
regr2 = RandomForestRegressor(n_estimators=100, bootstrap=True)
regr3 = RandomForestRegressor(n_estimators=100, bootstrap=True)
```

Figure 3.6 ตัวอย่างการสร้าง model Random Forest

ทำการ fit model ซึ่งจะเป็นกระบวนการในการฝึกสอนโมเดล โดยใช้ข้อมูลชุด train เพื่อให้โมเดลเรียนรู้และปรับปรุงตัวเองให้มีประสิทธิภาพในการทำงานกับข้อมูลใหม่ได้ดีขึ้น /// เพื่อปรับพารามิเตอร์ของโมเดลให้เหมาะสมกับข้อมูลนั้นๆ

```
regr1.fit(X1_train, y1_train)
regr2.fit(X2_train, y2_train)
regr3.fit(X3_train, y3_train)
```

Figure 3.7 ตัวอย่างการ fit model กับข้อมูลชุด train

3.3) Evaluating the Model

ทำการทดสอบเพื่อหาประสิทธิภาพของโมเดล โดยเราจะหาประสิทธิภาพของข้อมูล 2 ส่วน คือ ประสิทธิภาพของข้อมูลชุดทดสอบ (test data) สำหรับทดสอบโมเดลและวัดประสิทธิภาพของข้อมูลที่ไม่เคยใช้ในกระบวนการฝึกสอน และอีกส่วนคือ ทดสอบจากชุดข้อมูลทั้งหมด เพื่อนำไปใช้เปรียบเทียบกับข้อมูลจริง

```
# Predict X
regr1_pred = regr1.predict(X1)
regr2_pred = regr2.predict(X2)
regr3_pred = regr3.predict(X3)

# Predict X_test
regr1_pred_test = regr1.predict(X1_test)
regr2_pred_test = regr2.predict(X2_test)
regr3_pred_test = regr3.predict(X3_test)
```

Figure 3.8 ตัวอย่าง การทดสอบข้อมูลด้วยข้อมูลทั้งหมดและการทดสอบข้อมูลด้วยข้อมูลชุดทดสอบ

```
fig = plt.figure(figsize=(30,15))
fig.add_subplot(511)
plt.plot(df_try.index, df_try['Zone 1 Power Consumption'], label='Actual Zone 1 Power Consumption')
plt.plot(df_try.index ,df_try['zone1_pred'], label='Zone 1 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
fig.add_subplot(512)
plt.plot(df_try.index, df_try['Zone 2  Power Consumption'], label='Actual Zone 2 Power Consumption')
plt.plot(df_try.index ,df_try['zone2_pred'], label='Zone 2 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
fig.add_subplot(513)
plt.plot(df_try.index, df_try['Zone 3  Power Consumption'], label='Actual Zone 3 Power Consumption')
plt.plot(df_try.index ,df_try['zone3_pred'], label='Zone 3 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
```

Figure 3.9 การนำข้อมูลชุดทดสอบทั้งหมดไปเปรียบเทียบกับข้อมูลจริง

Model 2 : Wether Data + Time feature

```
df_model_2 = df
df_model_2["hour"] = df_model_2["DateTime"].apply(lambda x: x.strftime("%H"))
df_model_2["day_of_week"] = df_model_2["DateTime"].apply(lambda x: x.strftime("%A"))
df_model_2['day_of_week'] = df_model_2['day_of_week'].replace({'Sunday': 0, 'Monday': 1,'Tuesday':2,'Wednesday':3,'Thursday':4,'Friday':5,'Saturday':6})
df_model_2['hour'] = df_model_2['hour'].astype(int)
df_model_2 = df_model_2.set_index('DateTime')

# Define X and y
X1_model_2 = df_model_2.drop(['Zone 1 Power Consumption',
       'Zone 2  Power Consumption',
       'Zone 3  Power Consumption'], axis=1)
y1_model_2 = df_model_2['Zone 1 Power Consumption']

X2_model_2 = df_model_2.drop(['Zone 1 Power Consumption',
       'Zone 2  Power Consumption',
       'Zone 3  Power Consumption'], axis=1)
y2_model_2 = df_model_2['Zone 2  Power Consumption']

X3_model_2 = df_model_2.drop(['Zone 1 Power Consumption',
       'Zone 2  Power Consumption',
       'Zone 3  Power Consumption'], axis=1)
y3_model_2 = df_model_2['Zone 3  Power Consumption']

# Split arrays into random train and test subsets (70:30)
X1_train_model_2, X1_test_model_2, y1_train_model_2, y1_test_model_2 = train_test_split(X1_model_2, y1_model_2, test_size=0.30)
X2_train_model_2, X2_test_model_2, y2_train_model_2, y2_test_model_2 = train_test_split(X2_model_2, y2_model_2, test_size=0.30)
X3_train_model_2, X3_test_model_2, y3_train_model_2, y3_test_model_2 = train_test_split(X3_model_2, y3_model_2, test_size=0.30)

regr1_model_2 = RandomForestRegressor(n_estimators=100, bootstrap=True)
regr2_model_2 = RandomForestRegressor(n_estimators=100, bootstrap=True)
regr3_model_2 = RandomForestRegressor(n_estimators=100, bootstrap=True)

# Train (fit) rfc (X_train,y_train)
regr1_model_2.fit(X1_train_model_2, y1_train_model_2)
regr2_model_2.fit(X2_train_model_2, y2_train_model_2)
regr3_model_2.fit(X3_train_model_2, y3_train_model_2)

# Predict X for predict graph
regr1_pred_model_2 = regr1_model_2.predict(X1_model_2)
regr2_pred_model_2 = regr2_model_2.predict(X2_model_2)
regr3_pred_model_2 = regr3_model_2.predict(X3_model_2)
```

```
# Predict X_test for evaluate model
regr1_pred_test_model_2 = regr1_model_2.predict(X1_test_model_2)
regr2_pred_test_model_2 = regr2_model_2.predict(X2_test_model_2)
regr3_pred_test_model_2 = regr3_model_2.predict(X3_test_model_2)

df_model_2 = df_model_2.reset_index('DateTime')
df_model_2['zone1_pred_model_2'] = pd.DataFrame(regr1_pred_model_2, columns=['zone1_pred_model_2'])
df_model_2['zone2_pred_model_2'] = pd.DataFrame(regr2_pred_model_2, columns=['zone2_pred_model_2'])
df_model_2['zone3_pred_model_2'] = pd.DataFrame(regr3_pred_model_2, columns=['zone3_pred_model_2'])

# Graph Predict vs Real Power Consumption [ Model 2 ]
fig = plt.figure(figsize=(30,15))
fig.add_subplot(511)
plt.plot(df_model_2.index, df_model_2['Zone 1 Power Consumption'], label='Actual Zone 1 Power Consumption')
plt.plot(df_model_2.index ,df_model_2['zone1_pred_model_2'], label='Zone 1 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
fig.add_subplot(512)
plt.plot(df_model_2.index, df_model_2['Zone 2  Power Consumption'], label='Actual Zone 2 Power Consumption')
plt.plot(df_model_2.index ,df_model_2['zone2_pred_model_2'], label='Zone 2 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
fig.add_subplot(513)
plt.plot(df_model_2.index, df_model_2['Zone 3  Power Consumption'], label='Actual Zone 3 Power Consumption')
plt.plot(df_model_2.index ,df_model_2['zone3_pred_model_2'], label='Zone 3 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
```

Figure 3.10 การเขียนโปรแกรมการทำนายของโมเดลที่ 2

Model 3 : Wether Data + Time feature with PCA

```
df_model_3 = df.set_index('DateTime')
df_model_x_pca = df.set_index('DateTime')
X = df_model_x_pca.drop(['Zone 1 Power Consumption','Zone 2  Power Consumption','Zone 3  Power Consumption'], axis = 1)
sc = StandardScaler()
X_sc = sc.fit_transform(X)
pca = PCA(0.89)
pca.fit(X_sc)
pca.n_components_
pca = PCA(n_components = 5)
pca.fit(X_sc)

# Define X and y
X1_model_3 = df_model_3.drop(['Zone 1 Power Consumption',
       'Zone 2  Power Consumption',
       'Zone 3  Power Consumption'], axis=1)
y1_model_3 = df_model_3['Zone 1 Power Consumption']

X2_model_3 = df_model_3.drop(['Zone 1 Power Consumption',
       'Zone 2  Power Consumption',
       'Zone 3  Power Consumption'], axis=1)
y2_model_3 = df_model_3['Zone 2  Power Consumption']

X3_model_3 = df_model_3.drop(['Zone 1 Power Consumption',
       'Zone 2  Power Consumption',
       'Zone 3  Power Consumption'], axis=1)
y3_model_3 = df_model_3['Zone 3  Power Consumption']

# Split arrays into random train and test subsets (70:30)

X1_train_model_3, X1_test_model_3, y1_train_model_3, y1_test_model_3 = train_test_split(X1_model_3, y1_model_3, test_size=0.30)
X2_train_model_3, X2_test_model_3, y2_train_model_3, y2_test_model_3 = train_test_split(X2_model_3, y2_model_3, test_size=0.30)
X3_train_model_3, X3_test_model_3, y3_train_model_3, y3_test_model_3 = train_test_split(X3_model_3, y3_model_3, test_size=0.30)

X1_sc_model_3 = sc.transform(X1_model_3)
X1_pca_model_3 = pca.transform(X1_sc_model_3)
X2_sc_model_3 = sc.transform(X2_model_3)
X2_pca_model_3 = pca.transform(X2_sc_model_3)
X3_sc_model_3 = sc.transform(X3_model_3)
X3_pca_model_3 = pca.transform(X3_sc_model_3)
```

```
X1_train_sc_model_3 = sc.transform(X1_train_model_3)
X1_train_pca_model_3 = pca.transform(X1_train_sc_model_3)
X1_test_sc_model_3 = sc.transform(X1_test_model_3)
X1_test_pca_model_3 = pca.transform(X1_test_sc_model_3)

X2_train_sc_model_3 = sc.transform(X2_train_model_3)
X2_train_pca_model_3 = pca.transform(X2_train_sc_model_3)
X2_test_sc_model_3 = sc.transform(X2_test_model_3)
X2_test_pca_model_3 = pca.transform(X2_test_sc_model_3)

X3_train_sc_model_3 = sc.transform(X3_train_model_3)
X3_train_pca_model_3 = pca.transform(X3_train_sc_model_3)
X3_test_sc_model_3 = sc.transform(X3_test_model_3)
X3_test_pca_model_3 = pca.transform(X3_test_sc_model_3)

regr1_model_3 = RandomForestRegressor(n_estimators=100, bootstrap=True)
regr2_model_3 = RandomForestRegressor(n_estimators=100, bootstrap=True)
regr3_model_3 = RandomForestRegressor(n_estimators=100, bootstrap=True)

# Train (fit) rfc (X_train,y_train)
regr1_model_3.fit(X1_train_pca_model_3, y1_train_model_3)
regr2_model_3.fit(X2_train_pca_model_3, y2_train_model_3)
regr3_model_3.fit(X3_train_pca_model_3, y3_train_model_3)

# Predict X for Graph
regr1_pred_model_3 = regr1_model_3.predict(X1_pca_model_3)
regr2_pred_model_3 = regr2_model_3.predict(X2_pca_model_3)
regr3_pred_model_3 = regr3_model_3.predict(X3_pca_model_3)

# Predict X_test for evaluate model
regr1_pred_test_model_3 = regr1_model_3.predict(X1_test_pca_model_3)
regr2_pred_test_model_3 = regr2_model_3.predict(X2_test_pca_model_3)
regr3_pred_test_model_3 = regr3_model_3.predict(X3_test_pca_model_3)

df_model_3 = df_model_3.reset_index()
df_model_3['zone1_pred_model_3'] = pd.DataFrame(regr1_pred_model_3, columns=['zone1_pred_model_3'])
df_model_3['zone2_pred_model_3'] = pd.DataFrame(regr2_pred_model_3, columns=['zone2_pred_model_3'])
df_model_3['zone3_pred_model_3'] = pd.DataFrame(regr3_pred_model_3, columns=['zone3_pred_model_3'])
```

```
# Graph Predict vs Real Power Consumption [ Model 3 ]
fig = plt.figure(figsize=(30,15))
fig.add_subplot(511)
plt.plot(df_model_3.index, df_model_3['Zone 1 Power Consumption'], label='Actual Zone 1 Power Consumption')
plt.plot(df_model_3.index ,df_model_3['zone1_pred_model_3'], label='Zone 1 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
fig.add_subplot(512)
plt.plot(df_model_3.index, df_model_3['Zone 2  Power Consumption'], label='Actual Zone 2 Power Consumption')
plt.plot(df_model_3.index ,df_model_3['zone2_pred_model_3'], label='Zone 2 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
fig.add_subplot(513)
plt.plot(df_model_3.index, df_model_3['Zone 3  Power Consumption'], label='Actual Zone 3 Power Consumption')
plt.plot(df_model_3.index ,df_model_3['zone3_pred_model_3'], label='Zone 3 Power Consumption Predict')
plt.xlim(0, 2000)
plt.legend()
plt.grid()
```

Figure 3.11 การเขียนโปรแกรมการทำนายของโมเดลที่ 3

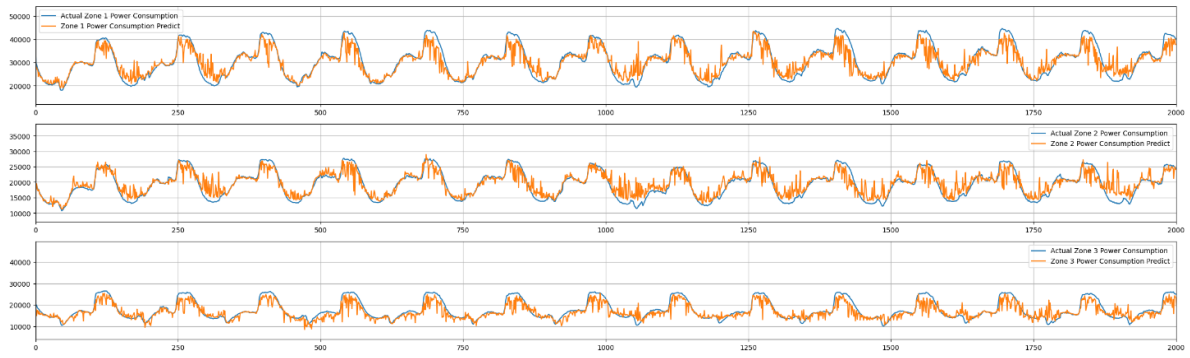

Figure 3.12 ผลลัพธ์ที่ได้จากการนำข้อมูลชุดทดสอบทั้งหมดไปเปรียบเทียบกับข้อมูลจริง โมเดลที่ 1

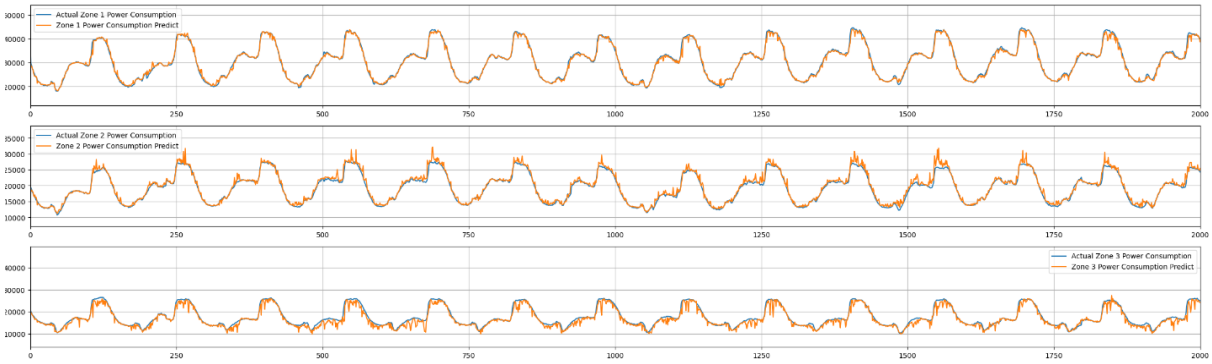

Figure 3.13 ผลลัพธ์ที่ได้จากการนำข้อมูลชุดทดสอบทั้งหมดไปเปรียบเทียบกับข้อมูลจริง โมเดลที่ 2

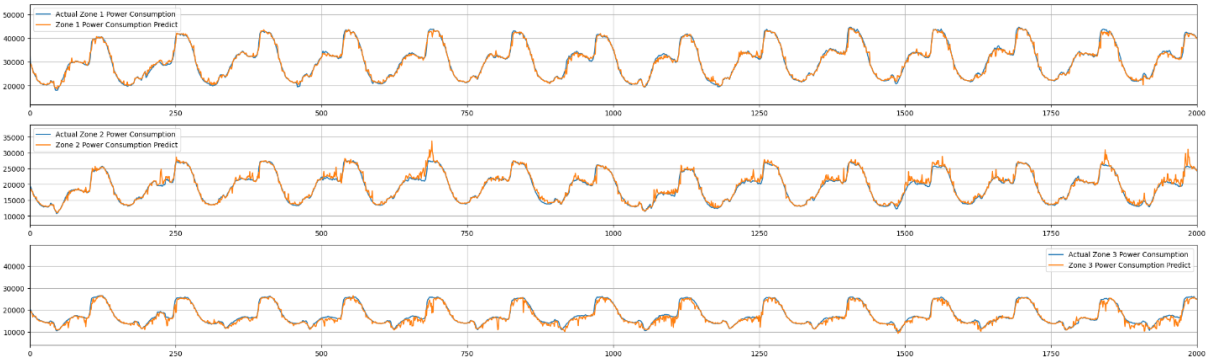

Figure 3.14 ผลลัพธ์ที่ได้จากการนำข้อมูลชุดทดสอบทั้งหมดไปเปรียบเทียบกับข้อมูลจริง โมเดลที่ 3

จากผลลัพธ์ที่ได้จากการนำข้อมูลชุดทดสอบทั้งหมดไปเปรียบเทียบกับข้อมูลจริงในแต่ละโมเดลนั้น พบว่า เมื่อมีการเพิ่ม Feature เวลา และใช้วิธีการลดมิติด้วยองค์ประกอบหลัก (Principle Components Analysis) ค่า Power Consumption ที่ได้จากการทำนายด้วยชุดข้อมูลทั้งหมด และค่า Power Consumption จริง มีค่าใกล้เคียงกัน ต่างจากโมเดล 1 ที่ค่าที่ได้จากการทำนายจะมีความคลาดเคลื่อนสูงเมื่อนำไปเทียบกับ Power Consumption จริง

โดยสำหรับการประเมินประสิทธิความแม่นยำของโมเดล เราจะใช้ข้อมูลชุดทดสอบไปประเมินด้วยวิธีการต่าง ๆ ดังนี้

1.MAE (Mean Absolute Error) เป็นการหาค่าเฉลี่ยของระยะความคลาดเคลื่อนทั้งหมด ซึ่งค่านี้จะมีความอ่อนไหวกับ Outlier เหมาะสมกับการประเมินข้อมูลที่มี Outliner เยอะ

2.MSE (Mean Square Error) เป็นการประเมินประสิทธิภาพโดยนำค่าความคลาดเคลื่อนไปยกกำลังสอง ซึ่งค่านี้จะ Sensitive กับ Outlier มาก ทำให้ไม่เหมาะกับข้อมูลที่มี Outlier เยอะ

3.RMSE (Root Mean Square Error) เป็นการนำ MSE มาถอดรากที่สอง ซึ่งอาจทำให้ตีความง่ายกว่าเนื่องจากหน่วยของค่า Error จะไม่มีเลขยกกำลัง 2

4.R2 (R Square) เป็นการคำนวณหาค่าสัมประสิทธิ์การตัดสินใจ ค่าวัดความเป็นสมบูรณ์ของการอธิบายของตัวแปรตาม (dependent variable) ด้วยตัวแปรอิสระ (independent variables) โดยค่า R-squared จะอยู่ในช่วง 0-1 หากมีค่าที่ใกล้เคียง 1 จะแสดงถึงการอธิบายข้อมูลตามโมเดลที่ดีมาก ในขณะที่ค่าใกล้เคียง 0 จะแสดงถึงการอธิบายข้อมูลตามโมเดลที่ไม่เหมาะสมเลย

```
from sklearn import metrics
```

```
from sklearn.metrics import r2_score
```

```
print('MAE1:', metrics.mean_absolute_error(y1_test, regr1_pred_test))
print('MSE1:', metrics.mean_squared_error(y1_test, regr1_pred_test))
print('RMSE1:', np.sqrt(metrics.mean_squared_error(y1_test, regr1_pred_test)))
print('r2_1:', r2_score(y1_test, regr1_pred_test))
```

```
print('MAE2:', metrics.mean_absolute_error(y2_test, regr2_pred_test))
print('MSE2:', metrics.mean_squared_error(y2_test, regr2_pred_test))
print('RMSE2:', np.sqrt(metrics.mean_squared_error(y2_test, regr2_pred_test)))
print('r2_2:', r2_score(y2_test, regr2_pred_test))
```

```
print('MAE3:', metrics.mean_absolute_error(y3_test, regr3_pred_test))
print('MSE3:', metrics.mean_squared_error(y3_test, regr3_pred_test))
print('RMSE3:', np.sqrt(metrics.mean_squared_error(y3_test, regr3_pred_test)))
print('r2_3:', r2_score(y3_test, regr3_pred_test))
```

Figure 3.15 ตัวอย่างการประเมินประสิทธิภาพของโมเดลในแต่ละZone

Table 3.1 ตารางแสดงผลลัพธ์จากการประเมินประสิทธิภาพของแต่ละโมเดลในแต่ละZone

Model		Zone 1	Zone 2	Zone 3
1	**MAE**	3233.640130387504	2393.1856428254882	2470.7624865361363
	MSE	22116671.308542468	11677545.348920867	12667753.506316928
	RMSE	4702.836517309789	3417.24236028422	3559.17876852469
	R^2	0.559792702321623	0.5712739131203344	0.7106514659090891
2	**MAE**	1005.626374118035	948.0474092511624	1209.6280717493019
	MSE	200011763.55493486	1946933.187147342	3356146.6749313488
	RMSE	1529.2593841685893	1395.3254771369086	1831.9788958749903
	R^2	0.9540303596873547	0.927983435742032	0.9228047984906916
3	**MAE**	1160.5930723009537	1080.936281012624	1370.0767103828416
	MSE	2985146.65846965	2544661.1557541443	4424015.14658485
	RMSE	1727.7576966894549	1595.1994094012648	2103.33429263749
	R^2	0.9421299730628002	0.9056061766576832	0.898332878251896

จาก Table 3.1 จะเห็นได้ว่าโมเดลที่ใช้ชุดของโมเดลที่ 1 การสร้างโมเดลที่ใช้เพียงตัวแปรจากข้อมูลสภาพอากาศและข้อมูลการไหลของน้ำใต้พื้นดินมีประสิทธิภาพของโมเดลต่ำ และเมื่อมีการเพิ่ม Feature เวลา และใช้วิธีการลดมิติด้วยองค์ประกอบหลัก (Principle Components Analysis) ผลลัพธ์ที่ได้มาจากโมเดลที่ 2 มีประสิทธิภาพของโมเดลสูงที่สุด ซึ่งมีค่าใกล้เคียงกันกับโมเดลที่ 3

Chapter 4 Conclusion

สำหรับการคาดการณ์การใช้พลังงานไฟฟ้าของเมือง Tetouan ประเทศ Morocco เพื่อวิเคราะห์การใช้พลังงานและเพื่อการจัดการการผลิตพลังงานไฟฟ้าที่ยั่งยืนและมีประสิทธิภาพที่สุด ซึ่งจากการสร้างแบบจำลองเพื่อคาดการณ์การใช้พลังงานในแต่ละโซนของเมือง Tetouan ซึ่งประกอบไปด้วย โซน Quads โซน Smir และโซน Boussafou โดยใช้ข้อมูลจากระบบ SCADA ประกอบไปด้วยอุณหภูมิ ความชื้น ความเร็วลม อัตราการไหลของน้ำใต้พื้นดิน รวมไปถึงค่าพลังงานที่ถูกใช้ในแต่ละโซน โดยข้อมูลเหล่านี้ถูกเก็บทุก ๆ 10 นาที เป็นระยะเวลา 1 ปี ตั้งแต่วันที่ 1 มกราคม ค.ศ.2017 ถึงวันที่ 31 ธันวาคม ค.ศ.2017 โดยแบบจำลองการเรียนรู้ของเครื่องที่ใช้ในการคาดการณ์การใช้พลังงานของเมือง Tetouan ในที่นี้จะใช้อัลกอริทึม Random Forest ซึ่งเป็นแบบอัลกอริทึมของแบบจำลองการเรียนรู้ของเครื่องที่เลือกจากการเปรียบเทียบกันกับอัลกอริทึมอื่น ๆ ได้แก่ แบบจำลองการถดถอยเชิงเส้น แบบจำลองซัพพอร์ตเวกเตอร์ แบบจำลองต้นไม้ตัดสินใจ และแบบจำลองการสุ่มป่าไม้ หรือ Random Forest ซึ่งมีประสิทธิภาพดีที่สุดพิจารณาจากผลลัพธ์ของการคาดการณ์ของแต่ละแบบจำลอง ซึ่งแบบจำลองการเรียนรู้ของเครื่องจะใช้เพื่อคาดการณ์การใช้พลังงานในทุก ๆ 10 นาที จากตัวแปรที่กล่าวในข้างต้น รวมถึงตัวแปรชั่วโมงในแต่ละวันและวันของสัปดาห์ ซึ่งส่งผลต่อการคาดการณ์ของแบบจำลอง โดยจะทำการทดสอบความสามารถในการสร้างแบบจำลองที่เหมาะสมในการคาดการณ์การใช้พลังงานไฟฟ้าทั้งหมด 4 แบบจำลอง เพื่อทดสอบความสามารถของแบบจำลองในการคาดการณ์บนตัวแปรและการปรับปรุงไฮเปอร์พารามิเตอร์ของอัลกอริทึมที่เลือกและเลือกแบบจำลองการเรียนรู้ของเครื่องที่มีประสิทธิภาพดีที่สุดในการนำมาเพื่อใช้คาดการณ์การใช้พลังงานไฟฟ้า โดยจะเริ่มจากแบบจำลองที่ 1 จะสร้างแบบจำลองที่ใช้เพียงตัวแปรจากข้อมูลสภาพอากาศและข้อมูลการไหลของน้ำใต้พื้นดิน แบบจำลองที่ 2 จะสร้างแบบจำลองที่ใช้ตัวแปรของชั่วโมงในแต่ละวันมาเป็นตัวแปรสำหรับเงื่อนไขในการคาดการณ์ร่วมด้วย แบบจำลองที่ 3 จะสร้างแบบจำลองโดยใช้ตัวแปรทั้งหมดในชุดเดียวกันกับแบบจำลองที่ 2 และจะมีการใช้วิธีการลดมิติด้วยองค์ประกอบหลัก (Principle Components Analysis) เพื่อวิเคราะห์หาองค์ประกอบหลักในการนำมาใช้เป็นตัวแปรเพื่อเพิ่มประสิทธิภาพในการคาดการณ์การใช้พลังงาน และแบบจำลองที่ 4 จะเป็นการปรับปรุงไฮเปอร์พารามิเตอร์พารามิเตอร์ของแบบจำลองที่ 3 ที่ผ่านการใช้วิธีลดมิติด้วยองค์ประกอบหลักแล้ว ซึ่งจะได้ผลการประเมินประสิทธิภาพของแบบจำลองแต่ละแบบดังตารางที่ 3.1 โดยจะพบว่าแบบจำลองการเรียนรู้ของเครื่องที่ 4 ซึ่งใช้อัลกอริทึมการสุ่มป่าไม้ หรือ Random Forest ที่ใช้การลดมิติด้วยองค์ประกอบหลักและปรับปรุงไฮเปอร์พารามิเตอร์ของอัลกอริทึมจะให้ผลในการคาดการณ์บนตัวแปรดังที่กล่าวมาข้างต้น และมีประสิทธิภาพในการใช้เพื่อคาดการณ์การใช้พลังงานเพื่อเพิ่มประสิทธิภาพของการผลิตพลังงานไฟฟ้าและจัดการการผลิตไฟฟ้าที่ยั่งยืน

Chapter 5 Reference

1.Salam, A., & El Hibaoui, A. (2018, December). Comparison of Machine Learning Algorithms for the Power Consumption Prediction:-Case Study of Tetouan cityâ€“. In 2018 6th International Renewable and Sustainable Energy Conference (IRSEC) (pp. 1-5).

2.ผศ.ดร.กอบเกียรติ สระอุบล, เรียนรู้ Data Science และ Al:Machine Learning, หน้า 321-594.