



ระบบ ผู้ช่วยเสมือนจริงสำหรับการกำกับดูแลปัญญาประดิษฐ์
Artificial Intelligence Supervisory Virtual Assistant

นาย ณัฐวุฒิ เตชสกุลวุฒิชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ
คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2565

ปริญญานิพนธ์เรื่อง : ระบบผู้ช่วยเสมือนจริงสำหรับการกำกับดูแลปัญญาประดิษฐ์
Artificial Intelligence Supervisory Virtual Assistant
โดย : นาย ญัฐวุฒิ เตชสกุลวุฒิชัย
สาขาวิชา : วิทยาการคอมพิวเตอร์
ภาควิชา : วิทยาการคอมพิวเตอร์และสารสนเทศ
คณะ : วิทยาศาสตร์ประยุกต์
อาจารย์ที่ปรึกษา : อาจารย์ ดร.สรร รัตนสัญญา
ปีการศึกษา : 2565

คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์

.....ประธานกรรมการ
(อาจารย์ ดร.สรร รัตนสัญญา)

.....กรรมการ
(รองศาสตราจารย์ ดร.ปรวัฒน์ วิสูตรศักดิ์)

.....กรรมการ
(รองศาสตราจารย์ ดร.เบญจพร ลิ้มธรรมมาภรณ์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุวัจชัย กมลสันติโรจน์)

.....หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ
(ผู้ช่วยศาสตราจารย์ ดร.อัครา ประโยชน์)

ลิขสิทธิ์ของภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2565

บทคัดย่อ

ในปัจจุบันผู้ควบคุมเครื่องจักรในโรงงาน สามารถควบคุมเครื่องจักรโดยต้องการใช้พลังงานต่ำที่สุดซึ่งช่วยประหยัดค่าใช้จ่ายพลังงานของโรงงาน โดยอาศัยคำแนะนำจากระบบ AI supervisory ที่ถูกพัฒนาจากเทคนิค machine learning ซึ่งเป็นการนำข้อมูลในอดีตมาทำการเรียนรู้ และ แนะนำ (recommend) วิธีการเดินเครื่องจักรที่ใช้พลังงานต่ำสุดให้กับผู้ควบคุมเครื่องจักร แต่ปัญหา คือ ผู้ที่ใช้ AI supervisory ได้แก่ AI engineer และ Operation engineer จะต้องทำการเปิดดูกราฟ Benefit Gap และวิเคราะห์ loss ในระบบ Plant information เอง เพื่อนำข้อมูลไปสื่อสารกับผู้ควบคุมเครื่องจักรต่อไป ซึ่งไม่สะดวก อีกทั้งในการปรับปรุง (maintenance) ตัว AI จะต้องทำการตรวจสอบ ข้อมูลการเดินเครื่องจักรเพื่อให้มั่นใจว่าข้อมูลทั้งหมดนั้นอยู่ในย่านที่เคยนำมาให้ AI ทำการเรียนรู้ หากข้อมูลการ run เครื่องจักรในปัจจุบันมีค่าเกินย่านของข้อมูลที้นำมาสอน AI อาจทำให้ AI ทำการ recommend การเดินเครื่องจักรผิดเพี้ยนไปได้ อาจสร้างความเสียหายให้กับกระบวนการผลิตเป็นอย่างมาก

ดังนั้นผู้จัดโครงการทำจึงได้สร้างระบบ AI supervisory virtual assistant เพื่อทำการสรุปข้อมูลที่ต้องการจาก AI supervisory รวมทั้ง การแจ้งเตือน (Notification) ข้อมูล การ run เครื่องจักรในปัจจุบันที่มีค่าเกินย่านของข้อมูลที้นำมาสอน AI ทำให้ประหยัดเวลา และสะดวกมากยิ่งขึ้น

โดยใช้เครื่องมืออาทิเช่น Notebook (เครื่องของบริษัท) และ ฐานข้อมูลในการจัดเก็บข้อมูล นอกจากนี้ยังใช้เทคโนโลยี Webhook ในการส่งข้อความผ่านทางเครื่องมือสื่อสาร คือ Microsoft Team เป็นต้น ซึ่งการจัดทำโครงการนี้มีการใช้ทฤษฎีของ Machine Learning และ Database ในการออกแบบการทำงาน และ พัฒนาระบบด้วยภาษาโปรแกรม Python โดยมีเป้าหมายว่าระบบที่พัฒนาจะสามารถนำไปใช้งานได้อย่างมีประสิทธิภาพ และเป็นตัวช่วยวิเคราะห์ข้อมูลเพื่อนำไปต่อยอดในการพัฒนาทางด้านอุตสาหกรรมอื่นๆ

หลังจากการพัฒนาระบบ AI supervisory virtual assistant ทำให้ได้ระบบที่สามารถทำงานได้ตามที่ผู้ใช้งานต้องการ โดยสามารถ Notification ข้อมูล สรุปข้อมูลที่ต้องการจาก AI supervisory นอกจากนี้ระหว่างการพัฒนาได้พบเจอปัญหาที่ส่งผลให้เกิดการพัฒนาในส่วนของ Anomaly Detection โดยจะช่วยตรวจความผิดปกติของข้อมูลที่ได้รับอีกด้วย

กิตติกรรมประกาศ

การจัดทำระบบผู้ช่วยเสมือนจริงสำหรับการกำกับดูแลปัญญาประดิษฐ์ โดยผู้จัดทำมีความตั้งใจเป็นอย่างยิ่งที่จะพัฒนาโปรแกรมให้ประสบความสำเร็จ มีประสิทธิภาพและเป็นไปตามความต้องการของผู้ใช้งานมากที่สุด

ปริญญานิพนธ์ฉบับนี้สามารถประสบความสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์และความช่วยเหลือจากอาจารย์ที่ปรึกษาโครงการ ดร.สรร รัตนสัญญา และ พี่เลี้ยง นาย อรรถพล ถาวร ที่ได้กรุณาสละเวลาอันมีค่าให้คำปรึกษาและได้แนะนำแนวทางต่าง ๆ ในการพัฒนาโปรแกรมมาโดยตลอด รวมถึงขอขอบพระคุณอาจารย์ทุกท่านในภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือที่อบรมสั่งสอนถ่ายทอดประสบการณ์ความรู้ทั้งในตำราและนอกตำราซึ่งส่งผลให้ปริญญานิพนธ์นี้เสร็จสมบูรณ์ไปด้วยดีขอขอบพระคุณทุกท่านที่ได้ให้คำปรึกษา คำแนะนำและให้ความช่วยเหลือจนปริญญานิพนธ์นี้ สำเร็จลุล่วงไปได้ด้วยดีและ การสนับสนุนอุปกรณ์จากทางบริษัท เอสซีจี เคมิคอลส์ จำกัด (มหาชน)

สุดท้ายนี้ ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และบุคคลในครอบครัวที่ได้ให้คำปรึกษา ให้การสนับสนุนให้กำลังใจ และคอยช่วยเหลือในทุกด้านตั้งแต่เริ่มต้นทำให้ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

นาย ณัฐวุฒิ เทชสกุลวุฒิชัย

23 ธันวาคม 2565

สารบัญ

	หน้า
บทคัดย่อ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญภาพ	ฉ
สารบัญตาราง	ช
สารบัญตาราง (ต่อ)	ซ
สารบัญตาราง (ต่อ)	ฌ
สารบัญตาราง (ต่อ)	ญ
บทที่	
1 บทนำ	1
1.1. ความเป็นมา และ ความสำคัญ	1
1.2. ปัญหาเพิ่มเติมที่เจอระหว่างการพัฒนา	4
1.3. วัตถุประสงค์	4
1.4. ขอบเขต	5
1.5. ประโยชน์ที่คาดว่าจะได้รับ	6
2 ทฤษฎีที่เกี่ยวข้อง	7
2.1. Machine learning	7
2.2. Database	10
3 ขั้นตอนและวิธีการดำเนินงาน	13
3.1. ภาพรวมของระบบ	13
3.2. การทำงาน	14
3.3. Entity relational Diagram (ER Diagram)	16
3.4. Data Dictionary	17
4 การพัฒนาระบบ	26
4.1. การออกแบบและเครื่องมือต่างๆ	26

สารบัญ (ต่อ)		หน้า
บทที่		
	4.2. การพัฒนาระบบของโครงการ	29
	4.3. ตารางการปฏิบัติงานระยะเวลา 6 เดือน	92
5	ผลการดำเนินงาน	94
	5.1. ส่วนผลลัพธ์ที่ผู้ใช้งานได้รับ	94
	5.2. ส่วนของฐานข้อมูล	98
6	บทสรุปและแนวทางในการพัฒนาต่อ	99
	6.1. สรุปผลการดำเนินงาน	99
	6.2. ปัญหาที่พบในการดำเนินงาน	99
	6.3. แนวทางในการแก้ปัญหา	100
	6.4. แนวทางการพัฒนาต่อ	100
	บรรณานุกรม	101
	บรรณานุกรม (ต่อ)	102
	ภาคผนวก ก	103

สารบัญภาพ

	หน้า
ภาพที่ 1-1 เป็นภาพตัวอย่างห้องควบคุมโรงงาน	1
ภาพที่ 1-2 กราฟ AI benefit gap	2
ภาพที่ 1-3 กราฟ AI feature drift	3
ภาพที่ 2-1 แผนภาพอธิบาย การทำงานของ machine learning	7
ภาพที่ 2-2 การแบ่งกลุ่ม cluster แบบ centroid-based	9
ภาพที่ 2-3 สมการ Euclidean distance	9
ภาพที่ 2-4 สมการ Euclidean distance หลังแทนค่า	10
ภาพที่ 2-5 โครงสร้างโมเดลเชิงสัมพันธ์	11
ภาพที่ 2-6 การสร้างคีย์หลัก	12
ภาพที่ 2-7 แสดงตัวอย่างคีย์นอก	12
ภาพที่ 3-1 สถาปัตยกรรมของระบบ	13
ภาพที่ 3-2 แผนภาพการทำงานทุกๆ 12 ชม	14
ภาพที่ 3-3 แผนภาพการทำงานทุกๆ 7 วัน	15
ภาพที่ 3-4 แผนภาพ ER	16
ภาพที่ 4-1 ผลลัพธ์จากคำสั่ง df.head(5)	67
ภาพที่ 4-2 ผลลัพธ์จากคำสั่ง df.describe	67
ภาพที่ 4-3 ผลลัพธ์จากคำสั่ง X	69
ภาพที่ 4-4 ผลลัพธ์จากคำสั่ง kmeans.labels	70
ภาพที่ 4-5 ผลลัพธ์จากคำสั่ง kmeans.cluster_centers	71

สารบัญตาราง

	หน้า
ตารางที่ 1-1 ตัวอย่างย่านข้อมูล AI	4
ตารางที่ 3-1 feature name	17
ตารางที่ 3-2 name	18
ตารางที่ 3-3 plant	19
ตารางที่ 3-4 report	20
ตารางที่ 3-5 report2	22
ตารางที่ 3-6 tag	23
ตารางที่ 3-7 feature	24
ตารางที่ 4-1 ส่วน library ที่ใช้ใน key vault	29
ตารางที่ 4-2 ส่วน ค่ากำหนดเริ่มต้นที่ใช้ใน key vault	30
ตารางที่ 4-3 ส่วน function ในการ เรียก key มาใช้ในการ ติดต่อ piview	30
ตารางที่ 4-4 ส่วน main	31
ตารางที่ 4-5 ส่วน libery ที่ใช้ใน AI_12hr_feature_drift_webhook	32
ตารางที่ 4-6 กำหนดค่าเริ่มต้นสำหรับการทำงาน	33
ตารางที่ 4-7 สร้าง obj จาก class module	33
ตารางที่ 4-8 ประกาศค่าสำหรับ piwebapi	34
ตารางที่ 4-9 กำหนดค่าสำหรับการrequest piwebapi	34
ตารางที่ 4-10 ทำการ request piwebapi	35
ตารางที่ 4-11 ทำการ คำนวนค่าและลบcolumnไม่จำเป็น	35
ตารางที่ 4-12 loop สำหรับการเพิ่มค่าประเภท X ,Y ลงlist	36
ตารางที่ 4-13 loopสำหรับเช็ค min-max ในและส่งค่าไปยังDB	37
ตารางที่ 4-14 เช็คเงื่อนไขว่า มี 1 หรือ -1 ใน ans หรือไม่	38
ตารางที่ 4-15 loopสำหรับจัดเตรียมข้อมูลต่างๆ ในการสร้าง message	39
ตารางที่ 4-16 loopสำหรับจัดเตรียมข้อมูลต่างๆ ในการสร้าง message (ต่อ)	40
ตารางที่ 4-17 loop สำหรับการสร้าง message	42
ตารางที่ 4-18 ประกาศโครงสร้างของ payload card โดยใส่ message เข้าไป	44
ตารางที่ 4-19 ทำการส่ง messagecard ไปยัง income service	45
ตารางที่ 4-20 ส่วน libery ที่ใช้ใน AI_12hr_gap_benefit_webhook	46

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 4-21 ประกาศ list ของ tag filename	47
ตารางที่ 4-22 กำหนดค่าเริ่มต้นสำหรับการทำงาน	47
ตารางที่ 4-23 ทำการrequest piwebapi 10 นาที	47
ตารางที่ 4-24 ประกาศค่า column และ drop time	48
ตารางที่ 4-25 loop check ว่า ตัวไหนมีค่าเกิน 100000	48
ตารางที่ 4-26 กำหนดค่าสำหรับการrequest piwebapi	49
ตารางที่ 4-27 ทำการrequest piwebapi 12 hr	50
ตารางที่ 4-28 loop เพื่อกำหนดค่า 0 ให้กลับ list ans	51
ตารางที่ 4-29 การสร้าง message	52
ตารางที่ 4-30 loop สำหรับการสร้าง message	53
ตารางที่ 4-31 ทำการเพิ่มค่าเข้าไปใน DB	55
ตารางที่ 4-32 ประกาศโครงสร้างของ payload card	56
ตารางที่ 4-33 ทำการส่ง messagecard ไปยัง income service	57
ตารางที่ 4-34 ส่วน libery ที่ใช้ใน AI_weekly_feature_drift_webhook	57
ตารางที่ 4-35 ประกาศค่าเริ่มต้นสำหรับการอ่านค่า	58
ตารางที่ 4-36 loopสำหรับการแยก X,Y และ นับยอดทั้งหมด	59
ตารางที่ 4-37 loopสำหรับการสร้าง message	60
ตารางที่ 4-38 ประกาศโครงสร้างของ payload card โดยใส่ message เข้าไป	62
ตารางที่ 4-39 ส่วน library ที่ใช้ใน AI_weekly_gap_benefit_webhook	62
ตารางที่ 4-40 ประกาศค่าเริ่มต้นสำหรับการ เช็คเงื่อนไข และ การสร้างmessage	63
ตารางที่ 4-41 การสร้าง message	63
ตารางที่ 4-42 loopสำหรับการสร้าง message	64
ตารางที่ 4-43 ประกาศโครงสร้างของpayload card	65
ตารางที่ 4-44 ทำการส่งmessagecardไปยัง income service	65
ตารางที่ 4-45 ส่วนการนำเข้าไฟล์	66
ตารางที่ 4-46 ส่วนการจัดการ colume	67
ตารางที่ 4-47 ส่วนการแปลง Dataframe to numpy	68
ตารางที่ 4-48 ส่วนการเตรียม min max scaler	68

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 4-49 ส่วนการเตรียมข้อมูล	69
ตารางที่ 4-50 ส่วนการทำ kmean clustering	70
ตารางที่ 4-51 ส่วนตรวจสอบความแม่นยำ model	70
ตารางที่ 4-52 ส่วนการsave model	71
ตารางที่ 4-53 ส่วน libery ที่ใช้ใน Unix_AD	71
ตารางที่ 4-54 กำหนดค่าสำหรับ piwebapi	72
ตารางที่ 4-55 ประกาศค่าเริ่มต้นสำหรับการอ่านค่า	72
ตารางที่ 4-56 ทำการ request piwebapi และ drop columnที่ไม่จำเป็น	72
ตารางที่ 4-57 ทำการ load model min-max scaler และทำการ prepare data	73
ตารางที่ 4-58 ทำการ load model kmean และทำการ predict data	73
ตารางที่ 4-59 ทำการเอาข้อมูลมาเช็คเงื่อนไข	74
ตารางที่ 4-60 หากเข้าเงื่อนไขจะทำการสร้างmessageและsave status ลง DB	74
ตารางที่ 4-61 โครงสร้างของpayload card	76
ตารางที่ 4-62 ส่วนไม่เข้าเงื่อนไข	76
ตารางที่ 4-63 ส่วนที่ทำการส่งmessage cardไปยัง income service	77
ตารางที่ 4-64 ส่วน libery ที่ใช้ใน dbreader	78
ตารางที่ 4-65 class DB	78
ตารางที่ 4-66 class ชื่อ database	79
ตารางที่ 4-67 function read	80
ตารางที่ 4-68 function inserts 2	81
ตารางที่ 4-69 function read 2	82
ตารางที่ 4-70 function getallplantname	83
ตารางที่ 4-71 function readtag	84
ตารางที่ 4-72 function getmax	85
ตารางที่ 4-73 function getmin	86
ตารางที่ 4-74 function getunit	87
ตารางที่ 4-75 function readname	88
ตารางที่ 4-76 function getplantname	89

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 4-77 function findplantindex	90
ตารางที่ 4-78 function findplantindex	91
ตารางที่ 4-79 ตารางการปฏิบัติงาน	92

บทที่ 1

บทนำ

1.1. ความเป็นมา และ ความสำคัญ

AI supervisory เป็นโปรแกรมระบบ recommendation system ที่ช่วยให้ผู้ควบคุมเครื่องจักรในโรงงานมาบตาพุดโอเลฟินส์ สามารถควบคุมเครื่องจักรโดยใช้พลังงานต่ำที่สุดได้ ซึ่งช่วยประหยัดค่าใช้พลังงานของโรงงาน AI supervisory เป็นโปรแกรมที่ถูกสร้างจากเทคนิค machine learning ซึ่งเป็นการนำข้อมูลในอดีตจำนวนมากๆมาทำให้โปรแกรมทำการเรียนรู้ว่าควรที่จะปรับเครื่องจักรอย่างไรจึงจะใช้พลังงานต่ำที่สุด หลังจากโปรแกรมทำการเรียนรู้เรียบร้อยแล้ว เราจะนำโปรแกรมนี้มาใช้ในการ recommend วิธีการเดินเครื่องจักรที่ใช้พลังงาน โดยลักษณะจะเป็นดังภาพที่ 1-1

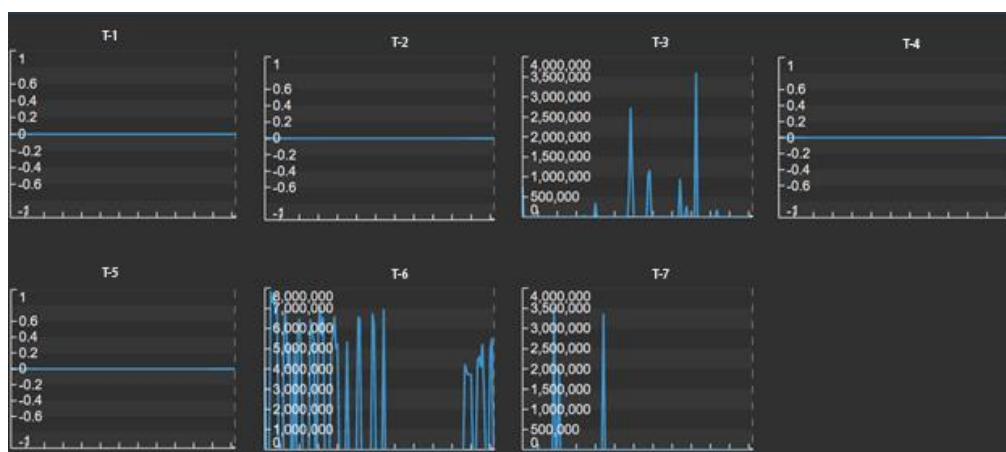


ภาพที่ 1-1 เป็นภาพตัวอย่างห้องควบคุมโรงงาน

ปัญหาคือ ผู้ที่ใช้ AI supervisory คือ AI engineer และ Operation engineer จะต้องคอย monitor ประสิทธิภาพของ AI ให้ดีอยู่ตลอดเวลา โดยจะต้อง monitor key parameter 2 ตัวได้แก่

1.1.1. Benefit Gap

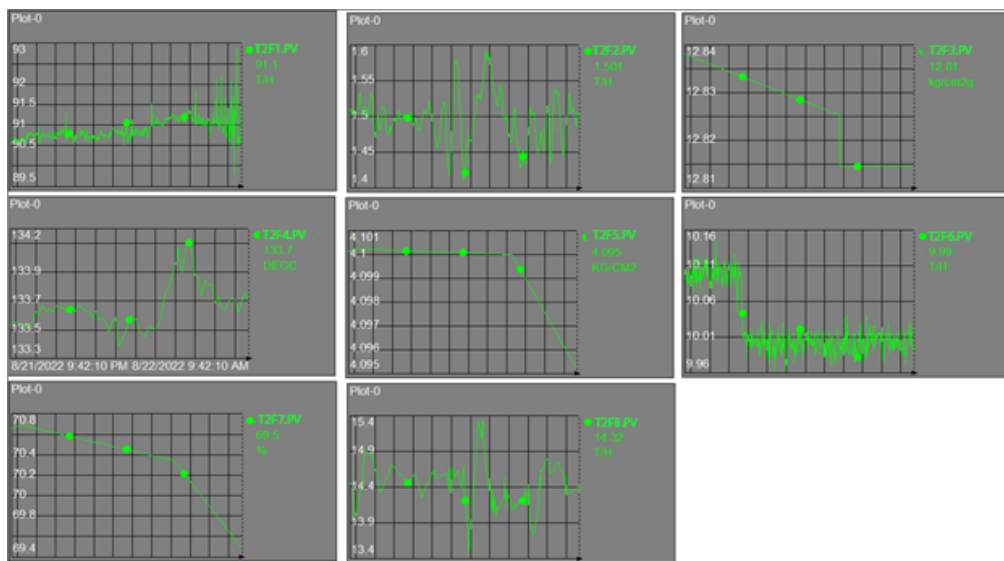
Benefit Gap: เป็น parameter ที่บ่งบอกว่าผู้ปฏิบัติงานได้ปรับเครื่องจักรตามที่ AI แนะนำหรือไม่ หากผู้ปฏิบัติงานไม่ได้ทำการปรับเครื่องจักรตามที่ AI แนะนำ จะทำให้มีตัวเลข loss ของ Benefit Gap ออกมาเป็นจำนวนเงินดัง ภาพที่ 1-2



ภาพที่ 1-2 กราฟ AI benefit gap

ปัญหาคือ Operation Engineer จะต้องทำการเปิดดูกราฟ Benefit Gap และวิเคราะห์ loss ในระบบ Plant information เอง วันละ 2 ครั้ง (กะเช้าและกะดึก) เช่น loss ที่ unit ไหนบ้าง, loss ที่บาท กี่ชั่วโมง เป็นต้น

เพื่อนำข้อมูลไปสื่อสารกับผู้ควบคุมเครื่องจักรต่อไป ซึ่งไม่สะดวก หากมีระบบ Virtual Assistant มาทำการสรุปข้อมูล Benefit Gap แล้วส่ง Notification ให้ operation engineer จะทำให้ประหยัดเวลามากขึ้น



ภาพที่ 1-3 กราฟ AI feature drift

ตารางที่ 1-1 ตัวอย่างข้อมูล AI

T-2				
X	tag	unit	min	max
Feed from fea5	T2F1.PV	T/H	76.3	120.71
Feed from fea6	T2F2.PV	T/H	0.3	5.53
Feed Temp	T2F3.PV	C	112.2	131.4
T-2 Btm Level	T2F4.PV	%	66	90
fea7 Ovhd Product	T2F5.PV	kscg	20	21.78
Y	tag	unit	min	max
fea8	T2F6.PV	T/H	12.1	17.21

1.2. ปัญหาเพิ่มเติมที่เจอระหว่างการพัฒนา

1.2.1. พบ feature drift มาด้วยจำนวนความถี่สูง ที่ UNIT4 เนื่องจาก Unit shut down แล้วเราไม่ทราบ ทำให้ค่าข้อมูลการ run เครื่องจักรในปัจจุบันมีค่าเกินย่านของข้อมูลที่นำมาสอน AI

1.2.2. เดิมทีระบบ AI supervisory virtual assistant ถูกพัฒนาให้เก็บข้อมูลผ่าน csv file ที่ถูก save ไว้ใน Azure Kubernetes service แต่พบปัญหาว่า เมื่อ Kubernetes cluster ล่ม ข้อมูลที่เก็บอยู่ใน csv file จะหายไป ส่งผลให้ไม่สามารถนำข้อมูลไปทำ weekly report ต่อได้

1.3. วัตถุประสงค์

โดยหลังการสามารถสรุปปัญหาได้แล้วจะสามารถกำหนดวัตถุประสงค์ว่าระบบที่จะจัดทำขึ้นนั้นมีวัตถุประสงค์เพื่ออะไร ดังต่อไปนี้

- 1.3.1. ระบบทำการสรุป insight และ notify ผ่าน MS Teams
- 1.3.2. ระบบจะทำการตรวจเงื่อนไขของข้อมูลจาก plant information
- 1.3.3. ระบบนำโมเดลที่มีมาทำการวิเคราะห์ความผิดปกติของข้อมูลได้

1.4. ขอบเขต

จากนั้นหลังจากได้วัตถุประสงค์มาแล้ว ก็จะเป็นส่วนของการกำหนดขอบเขตของระบบที่สามารถทำได้โดยแบ่งเป็น 5 ส่วนโดยมีรายละเอียดดังนี้

1.4.1. Virtual Assistant ของ AI Supervisory (UNIT3+UNIT4)

เป็นไฟล์ที่มีหน้าที่ เรียกดูข้อมูลจาก plant information แล้วทำการสรุปข้อมูล Gap benefit ตามเงื่อนไขที่กำหนดจากนั้นส่ง notification และเก็บข้อมูลลง Database ซึ่งสรุปเป็นขอบเขตดังนี้

- 1) เรียกดูข้อมูล จาก plant information
- 2) ทำการสรุปข้อมูล Gap benefit อาศัยเงื่อนไขที่กำหนด
- 3) ส่ง notification
- 4) เก็บข้อมูลจำนวนครั้งที่เกิด ลง Database สำหรับการทำให้ weekly report

1.4.2. Virtual Assistant ของ AI Supervisory (ML Feature Drift)

เป็นไฟล์ที่มีหน้าที่ เรียกดูข้อมูลจาก plant information แล้วทำการสรุปข้อมูล ML Feature Drift ตามเงื่อนไขที่กำหนดจากนั้นส่ง notification และเก็บข้อมูลลง Database ซึ่งสรุปเป็นขอบเขตดังนี้

- 1) เรียกดูข้อมูล จาก plant information
- 2) ทำการสรุปข้อมูล ML Feature Drift อาศัยเกณฑ์ที่กำหนด
- 3) ส่ง notification
- 4) เก็บข้อมูลจำนวนครั้งที่เกิน ลง Database สำหรับการทำให้ weekly report

1.4.3. weekly report AI Supervisory Virtual Assistant

เป็นไฟล์ที่มีหน้าที่ เรียกดูข้อมูลจาก Database แล้วทำการสรุปข้อมูลจากนั้นส่ง notification ซึ่งสรุปเป็นขอบเขตดังนี้

- 1) เรียกดูข้อมูล จาก ที่จัดเก็บข้อมูล
- 2) ทำการสรุปข้อมูล
- 3) ส่ง notification

1.4.4. anomaly detection

เป็นไฟล์ที่มีหน้าที่ เรียกดูข้อมูลจาก plant information แล้วทำการเตรียมข้อมูลด้วยการ ทำmin max scale ข้อมูลจากนั้นpredictด้วย model ตามเงื่อนไขที่กำหนด ส่ง notification ซึ่งสรุปเป็นขอบเขตดังนี้

- 1) ทำ min-max scaler ได้
- 2) นำ model ML มา ทำการ predict
- 3) เช็คเงื่อนไขที่กำหนด

1.4.5. DB reader

เป็นไฟล์ที่มีหน้าที่ เป็นส่วนในการติดต่อ Database โดยมีฟังก์ชันรองรับการ insert `หรือ read data แบบต่างๆ ซึ่งสรุปเป็นขอบเขตดังนี้

- 1) มีฟังก์ชันรองรับการ insert data แบบต่างๆ
- 2) มีฟังก์ชันรองรับการ read data แบบต่างๆ

1.5. ประโยชน์ที่คาดว่าจะได้รับ

จากสองข้อที่กล่าวมาข้างต้นทำให้สามารถสรุปประโยชน์ที่คาดว่าจะได้รับ โดยทั้งหมดล้วนเกิดผลต่อผู้ใช้งานและผู้นำไปต่อยอดได้โดยสรุปได้ดังนี้

- 1.5.1. ประหยัดเวลามากขึ้นในการตรวจสอบข้อมูล gap benefit ของ Unit4 กับ Unit3
- 1.5.2. ตัว AI supervisory มีความแม่นยำมากขึ้น ส่งผลให้การปรับตัวแปร unit ช่วยประหยัดพลังงานมีประสิทธิภาพ
- 1.5.3. สามารถนำไปปรับใช้และต่อยอดกับplant อื่นๆได้
- 1.5.4. ข้อมูลที่ถูกเก็บสามารถนำไปพัฒนางานอื่นๆต่อได้ เนื่องจากระบบ PostgreSQL ยืดหยุ่น

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

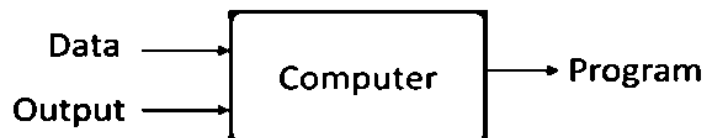
การที่จะจัดทำโครงการนี้ นอกจากอาศัยเครื่องมือแล้ว ยังต้องอาศัยความรู้จากทฤษฎีต่างๆ มาเป็นแนวทางการออกแบบ และปฏิบัติ โดยทฤษฎีได้จากการเรียนและศึกษาจากตำรา ซึ่งช่วยให้การดำเนินโครงการสามารถทำได้อย่างราบรื่น โดยทฤษฎีที่จะกล่าวถึงได้แก่ machine learning (K mean clustering) และ Database(relational database)

2.1. Machine learning

2.1.1. Machine learning

Machine learning คือการศึกษา algorithms (อัลกอริทึม) ที่คอมพิวเตอร์ ใช้ในการแก้ปัญหา หรือ ตัดสินใจ โดยอัตโนมัติ โดยอาศัยข้อมูลที่มีอยู่

Machine Learning



ภาพที่ 2-1 แผนภาพอธิบาย การทำงานของ machine learning

ดังภาพที่ 2-1 คอมพิวเตอร์ที่ได้รับการ training ด้วยข้อมูลที่เรียกว่า feature data หรือ training set และ label หรือ output คอมพิวเตอร์จะทำการเรียนรู้ algorithms ที่จะสร้าง model (โมเดล) ที่อิงจาก training data โดยเราเรียกทั้งหมดนี้ว่า machine learning.

machine learning นั้นมี อยู่หลายตัวด้วยกันแต่แบ่งออกเป็นกลุ่มใหญ่ๆได้อยู่ 3 ชนิด คือ

2.1.2. Supervised machine learning โดยที่คอมพิวเตอร์จะได้รับ feature data และ output จากนั้นทำการเรียนรู้ในการ mapping feature data และ output

2.1.3. Unsupervised machine learning โดยที่คอมพิวเตอร์จะได้รับ feature data จากนั้นทำการเรียนรู้ในการ feature data และกำหนด output เอง

2.1.4. Reinforcement machine learning โดยที่ คอมพิวเตอร์ จะได้รับ state (สถานะ) และ action(การกระทำ) ซึ่ง action จะมีผลต่อ state ที่จะได้รับ จากนั้นทำการเรียนรู้ในการ เพิ่มรางวัลให้มากที่สุดจาก state ด้วยการเลือกขั้นตอนในการ action

จาก machine learning ชนิดที่ 2 “Unsupervised” หมายความว่า คอมพิวเตอร์จะไม่ได้ รับ output สำหรับการเรียนรู้ หมายความว่า Unsupervised machine learning ยากเกินกว่าที่จะ evaluate(การวัดประสิทธิภาพ) แต่ปัญหานี้จะแก้ด้วยการทำ Clustering

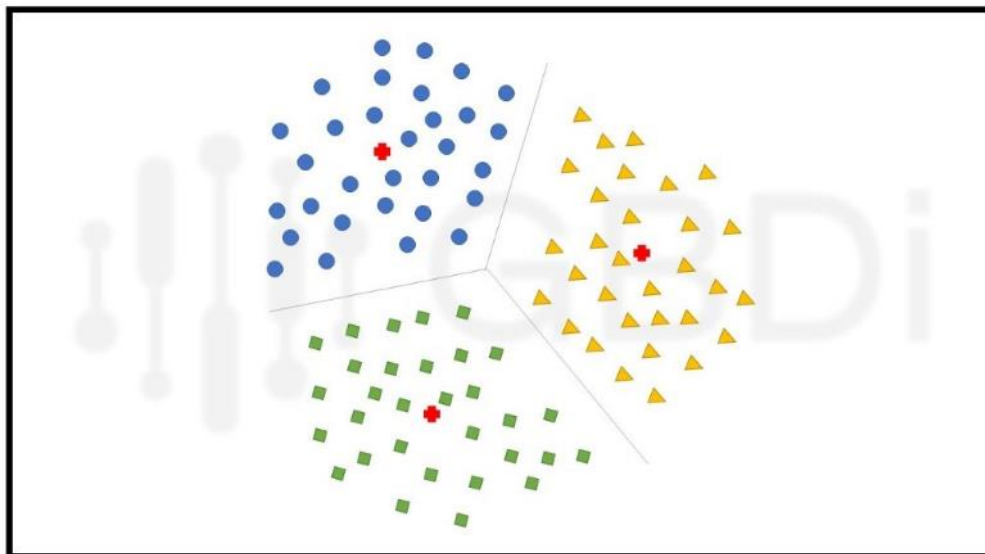
2.1.5. Clustering

Clustering จัดอยู่ในประเภทของการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เมื่อ เราได้กลุ่มข้อมูลที่เครื่องแยกไว้ให้เราแล้ว เราก็ค่อยเข้าไปดูลักษณะของแต่ละกลุ่มและกำหนดชื่อให้ กลุ่มนั้น ๆ

Centroid-based Clustering การแบ่งกลุ่ม cluster แบบ centroid-based

การแบ่งกลุ่มของข้อมูลเกิดจากจุดข้อมูล (Data Point) ที่กระจุกตัวใกล้กับจุดกึ่งกลางของกลุ่ม (Centroid) มากที่สุด การแบ่งกลุ่มแบบนี้มีประสิทธิภาพแต่ก็มีความอ่อนไหวขึ้นกับเงื่อนไขตั้ง ต้นและข้อมูลที่เป็น outlier ซึ่ง algorithm ที่นิยมใช้และถูกกล่าวถึงบ่อยครั้ง คือ k-means เพราะ เป็น algorithm ที่ง่ายและมีประสิทธิภาพ โดยมีการทำงานแบบ iterative คือวนซ้ำ ๆ เพื่อคำนวณ ระยะห่าง หรือ Euclidean Distance ของ data point กับ centroid ในแต่ละกลุ่ม ผลที่ได้คือ data point ที่อยู่ใกล้กับ centroid ไหนมากที่สุด ก็จะถูกจัดให้อยู่กลุ่มเดียวกันกับ centroid นั้นมีข้อควร

ระลึกไว้คือ ผลลัพธ์จากการใช้ k-means ในแต่ละครั้งจะไม่เหมือนกันเนื่องจากการกำหนด centroid ในขั้นแรกสุดนั้นเป็นการสุ่ม ดังภาพที่ 2-2 (Srimarong, 2020)



ภาพที่ 2-2 การแบ่งกลุ่ม cluster แบบ centroid-based

2.1.6. K mean clustering

เพราะด้วยความเรียบง่ายของโมเดล และประสิทธิภาพของการแบ่งกลุ่มที่อยู่ในเกณฑ์ที่ยอมรับได้ k-means จึงเป็นที่นิยมและมักจะถูกเรียกใช้งานอยู่บ่อยครั้ง

โดยทั่วไปแล้ว k-means มีวิธีการทำงานโดยเริ่มจากการกำหนดจุดศูนย์กลางของกลุ่ม (centroid) มาจำนวนจุด คือจำนวนกลุ่มที่คาดว่าจะได้จากการแบ่งกลุ่ม จากนั้นคำนวณระยะห่างระหว่างข้อมูลในแต่ละแถวกับ centroid เพื่อใช้ในการจัดกลุ่มข้อมูล โดยระยะห่างนั้นสามารถคำนวณโดยใช้ สมการ Euclidean distance ภาพที่ 2-3

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

ภาพที่ 2-3 สมการ Euclidean distance

โดยที่ x และ y เป็นข้อมูลแต่ละแถวที่มีความยาว 12 หรือมี 10 ค่า ซึ่งสามารถเข้าถึงแต่ละค่าได้จาก n index ; ตัวอย่างเช่น การคำนวณ Euclidean distance สำหรับข้อมูลที่มีความยาว $x = [-1, 3]$ และ $y = [2, 7]$ ทำได้ดังภาพที่ 2-4

$$\begin{aligned}
 d(\mathbf{x}, \mathbf{y}) &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \\
 &= \sqrt{(-1 - 2)^2 + (3 - 7)^2} \\
 &= 5
 \end{aligned}$$

ภาพที่ 2-4 สมการ Euclidean distance หลังแทนค่า

เมื่อจัดกลุ่มครบทุกแถวข้อมูลแล้วจะมีการปรับตำแหน่งของ centroid ในแต่ละกลุ่มใหม่ และทำวนซ้ำไปเรื่อย ๆ จนกว่าจะตรงตามเงื่อนไขในการหยุดการทำงาน (ธรรมเจริญพร, 2020)

2.2. Database

ฐานข้อมูล (Database) ประกอบด้วยรายละเอียดข้อมูลที่มีความเกี่ยวข้องกัน ซึ่งจะถูกนำมาใช้งานในด้านต่างๆ โดยข้อมูลเหล่านี้จะถูกจัดเก็บไว้ให้เป็นศูนย์กลางอย่างมีระบบซึ่งสามารถเรียกใช้งานร่วมกันได้ เพื่อประโยชน์ในการจัดการและการเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ

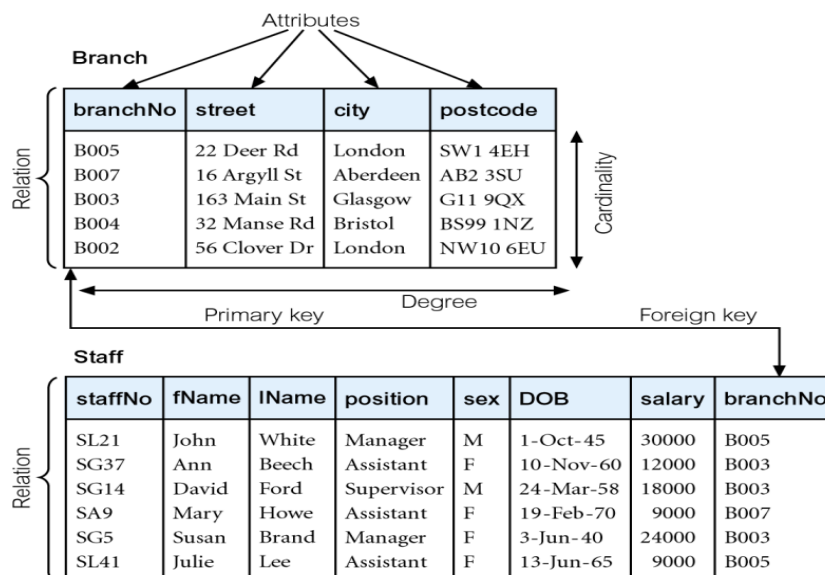
องค์ประกอบของระบบฐานข้อมูลมี ฮาร์ดแวร์ (Hardware) , ซอฟต์แวร์ (Software) หรือ โปรแกรม (Program) , ข้อมูล (Data) , บุคลากร (People) , ขั้นตอนการปฏิบัติงาน (Procedure)

2.2.1. ฐานข้อมูลแบบสัมพันธ์ (Relational Data Model)

โครงสร้างของฐานข้อมูลอยู่ในรูปของความสัมพันธ์ หรือ ตาราง ประกอบด้วย แถว (Tuple) และ คอลัมน์ (Attribute) โดยความสัมพันธ์ของฐานข้อมูลจะอยู่ในรูปแบบของตาราง

ตัวอย่างของภาษาปฏิบัติการ คือ SQL” (สุริยะฉาย, 2021)

2.2.2. โมเดลเชิงสัมพันธ์



ภาพที่ 2-5 โครงสร้างโมเดลเชิงสัมพันธ์

จากภาพที่ 2-5 ชื่อของตารางคือ ชื่อของ ความสัมพันธ์ (Relation)

โดยที่แต่ละคอลัมน์ของตารางความสัมพันธ์ เรียกว่า แอททริบิว(Attribute) ของความสัมพันธ์

2.2.3. Primary Key and Foreign Key

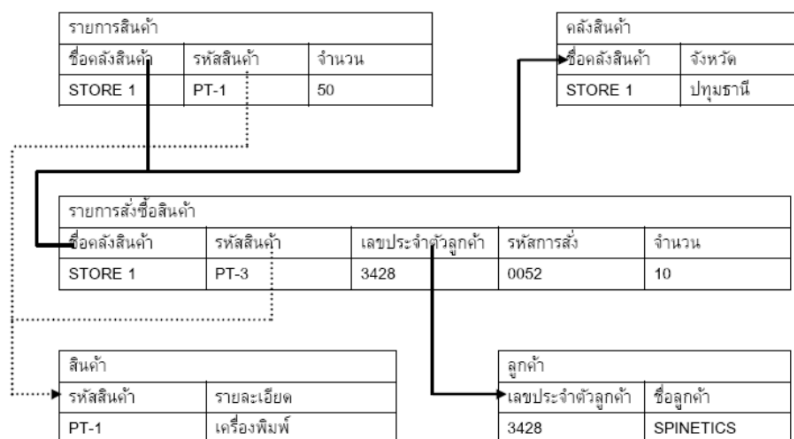
“คีย์หลัก (Primary Key) คือ คีย์ที่ใช้ในการอ้างอิง Entity ในฐานข้อมูล สามารถเลือกได้จาก Record ใด ๆ ก็ได้ที่ ไม่มีโอกาสซ้ำซ้อนกัน คีย์หลักเป็นข้อมูลสำคัญที่จะทำให้การเข้าถึงข้อมูลเป็นไปได้อย่างรวดเร็ว” (สุริยะฉาย, 2021)

รูปที่ภาพ 2-6 หากไม่มีข้อมูลใดเลยในฐานข้อมูลที่เหมาะสมที่จะเป็นคีย์หลักควรที่จะ กำหนด Record ใหม่ให้เป็นคีย์หลักโดยเฉพาะ

ชื่อ	นามสกุล	อายุ	เพศ	ID	ชื่อ	นามสกุล	อายุ	เพศ
สมชาย	แช่ตั้ง	25	ชาย	1	สมชาย	แช่ตั้ง	25	ชาย
สมศรี	แช่อึ้ง	42	หญิง	2	สมศรี	แช่อึ้ง	42	หญิง
สมชาย	แช่ตั้ง	25	ชาย	3	สมชาย	แช่ตั้ง	25	ชาย
สมปอง	แช่แต้	16	ชาย	4	สมปอง	แช่แต้	16	ชาย
สมชัย	แช่เล้ง	50	ชาย	5	สมชัย	แช่เล้ง	50	ชาย

ภาพที่ 2-6 การสร้างคีย์หลัก

“คีย์นอก (Foreign Key) คือ คีย์เดี่ยวหรือคีย์ผสม ซึ่งเป็นคีย์ทั่วไปของความสัมพันธ์หนึ่ง แต่เป็น คีย์หลักของอีกความสัมพันธ์หนึ่ง ดังภาพที่ 2-7 ที่ ชื่อสินค้า ที่ตาราง รายการสินค้า และ คลังสินค้า เป็นคีย์นอก ของ ชื่อสินค้า ที่ตาราง รายการสั่งซื้อ” (สุริยะฉาย, 2021)



ภาพที่ 2-7 แสดงตัวอย่างคีย์นอก

จากทฤษฎีที่กล่าวมาทั้งหมดนี้ สามารถนำมาใช้งานได้จริง เช่น การนำไปใช้ในการออกแบบ ระบบฐานข้อมูล การกำหนดอัลกอริทึมการทำงาน เป็นต้น และมีประโยชน์ในการทำโครงการนี้ไม่ว่า ทางตรงและทางอ้อม ส่งผลให้งานนั้นสามารถประสบความสำเร็จได้เป็นอย่างดี

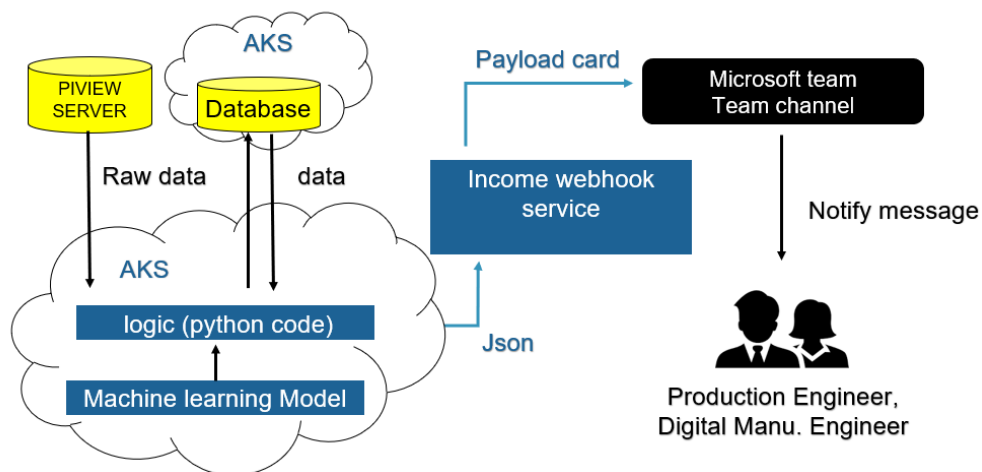
บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงในส่วนของขั้นตอนและวิธีการดำเนินงาน โดยจะอธิบายเรื่องภาพรวมระบบ ว่ามีองค์ประกอบอะไรบ้างที่ใช้ มีความสัมพันธ์อย่างไรบ้าง การทำงาน รวมไปถึง ER Diagram และ Data dictionary

3.1. ภาพรวมของระบบ

จากภาพที่ 3-1 ระบบจะมี ส่วนของตัวระบบ มี โค้ด เชิงตรรกะ(python) และมี machine learning model โดยที่สองอย่างนี้จะทำงานบน Azure Kubernetes service โดยเชื่อมต่อกับ PIVIEW SERVER และ Database บนระบบ AKS อีก cluster ซึ่งแยกออกจากอีกตัว โดยที่ระบบจะติดต่อไปยัง service คือ income webhook ที่เชื่อมต่อกับ Microsoft team โดยมีผู้ใช้งานคือ Production Engineer และ Digital Manu. Engineer

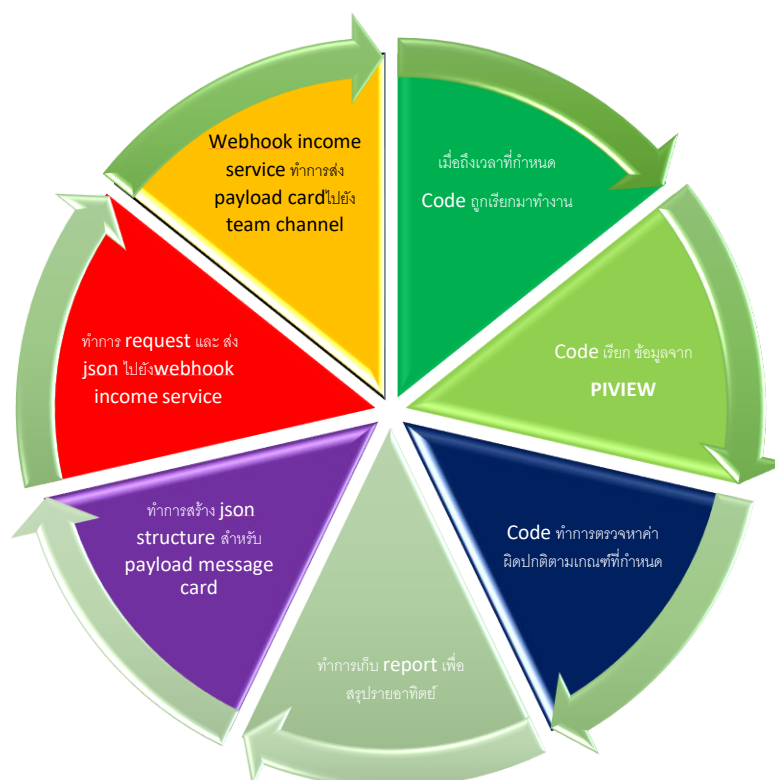


ภาพที่ 3-1 สถาปัตยกรรมของระบบ

3.2. การทำงาน

ในส่วนขั้นตอนการทำงานของระบบโดยจะมีตัวกำหนดเวลา เนื่องจากไม่ต้องการให้ทำงานตลอดเวลา โดยจะกำหนดใน file เรียกว่า CronJob เมื่อถึงกำหนดเวลา AKS จะ run ตามเวลาที่ตั้งใน file ได้

3.2.1. การทำงานทุก 12 ชม



ภาพที่ 3-2 แผนภาพการทำงานทุกๆ 12 ชม

จากภาพที่ 3-2 จะเริ่มเมื่อเวลาใน AKS ตรงกับ cornfile โดยตัว logic จะถูกเรียกมาทำงาน โดยจะเรียกข้อมูล PIVIEW จากนั้นตรวจค่าผิดปกติตามเกณฑ์ที่กำหนด และสร้าง และ เก็บ report สำหรับทำสรุปรายอาหิตย์ โดยหลังจากนั้นทำการสร้าง json structure สำหรับ payload message card และส่งผ่าน request ไปยัง webhook income service เพื่อให้ แสดง message บน channel ที่ต้องการ

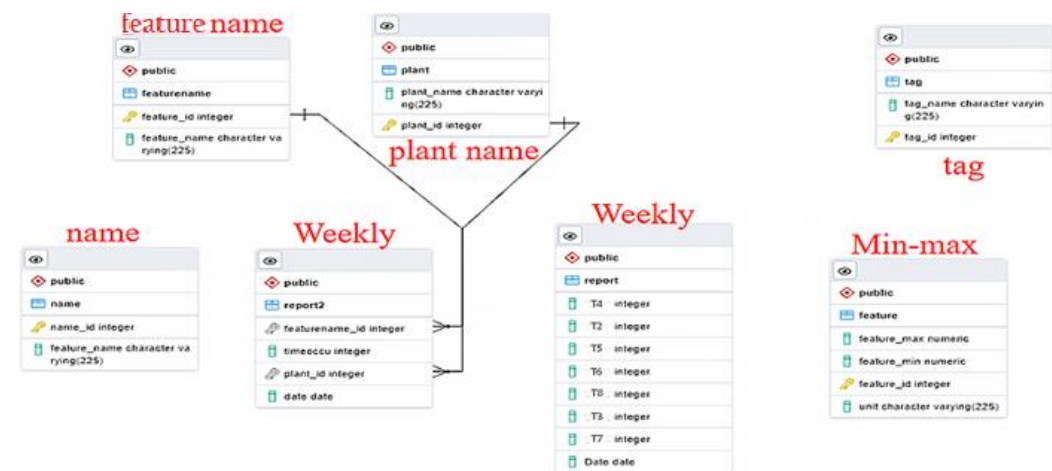
3.2.2. การทำงานทุก 7 วัน



ภาพที่ 3-3 แผนภาพการทำงานทุกๆ 7 วัน

จากภาพที่ 3-3 จะเริ่มเมื่อเวลาใน AKS ตรงกับ cornfile โดยตัว logic จะถูกเรียกมาทำงาน โดยจะเรียกและ อ่าน report สรุปรายอาทิตย์ โดยหลังจากนั้นทำการสร้าง json structure สำหรับ payload message card และส่งผ่าน request ไปยัง webhook income service เพื่อให้ แสดง message บน channel ที่ต้องการ

3.3. Entity relational Diagram (ER Diagram)



ภาพที่ 3-4 แผนภาพ ER

เนื่องจากโครงงานมีการใช้ฐานข้อมูลดังนั้นในส่วน ER Diagram จะอธิบายในเรื่องความสัมพันธ์ระหว่าง ตาราง โดย จากภาพที่ 3-4 นั้น

ตาราง feature name และ ตาราง plant name สัมพันธ์กับ ตาราง Weekly ของ feature ชื่อ feature_id แบบ 1 to many

ตารางอื่นๆ จะแยกออกจากกันโดยแต่ละตารางมีหน้าที่ดังนี้

ตาราง feature name มีหน้าที่ เก็บ feature name แบบตัดตัว feature ที่ไม่ใช่

ตาราง name มีหน้าที่ เก็บ feature name ทุกตัวไม่มีการตัดบางตัวออก

ตาราง plant name มีหน้าที่ เก็บชื่อ plant

ตาราง report2 name มีหน้าที่เก็บข้อมูล report ของ opportunity loss ที่เกิดขึ้นแต่ละค่า เพื่อใช้ใน weekly report

ตาราง report name มีหน้าที่เก็บข้อมูล report ของ feature drift ที่เกิดขึ้นแต่ละค่า เพื่อใช้ใน weekly report

ตาราง tag มีหน้าที่ เก็บชื่อ tag Column สำหรับใช้ในการกำหนดชื่อ Colome Data frame

ตาราง feature มีหน้าที่ เก็บค่า min และ max สำหรับใช้เป็นเกณฑ์ในการเปรียบเทียบข้อมูลที่ต้องการ

3.4. Data Dictionary

ในส่วนนี้ จะเป็นการลงรายละเอียดข้อมูลที่ใช้ฐานข้อมูลทั้งหมดโดยมีรายละเอียด ประเภท และหน้าที่ การกำหนดค่า ที่มีความสำคัญในการสร้างฐานข้อมูล โดยมีดังนี้

3.4.1. featurename

ตารางที่ 3-1 feature name

Attribute Name	description	data type	length	key type	nullable
feature_id	AUTO_INCREMENT feature id	integer	32	PK	NOT NULL
feature_name	ชื่อ feature	character varying	225		NULL

ตารางที่ 3-1 ประกอบด้วย 2 Attribute ซึ่งจะประกอบด้วย

feature_id ตัวแปร integer บันทึก AUTO_INCREMENT เป็น PRIMARY key ไม่เป็นค่า NULL

feature_name ตัวแปร character varying บันทึก ชื่อ feature เป็นค่า NULL ได้

featurename_id มีหน้าที่ มีหน้าที่เก็บข้อมูล id ของแต่ละ featurename เพื่อใช้เป็น PRIMARY key สำหรับเชื่อมตาราง report

feature_name มีหน้าที่เก็บข้อมูลชื่อของแต่ละ featurename เพื่อใช้ในการระบุชื่อ feature ในการระบุชื่อ column

3.4.2. name

ตารางที่ 3-2 name

Attribute Name	description	data type	length	key type	nullable
featurename_id	AUTO_INCREMENT feature name id	integer	32	PK	NOT NULL
feature_name	ชื่อ feature	character varying	225		NOT NULL

ตารางที่ 3 - 2 ประกอบด้วย 2 Attribute ซึ่งจะประกอบด้วย

featurename_id ตัวแปร integer บันทึก AUTO_INCREMENT เป็น PRIMARY key ไม่เป็นค่า NULL

feature_name ตัวแปร character varying บันทึก ชื่อ feature ไม่เป็นค่า NULL

feature_id มีหน้าที่เก็บข้อมูล id ของแต่ละ name เพื่อใช้เป็น PRIMARY key

feature_name มีหน้าที่เก็บข้อมูลชื่อของแต่ละ feature เพื่อใช้ในการระบุชื่อ feature ในการสร้าง json

3.4.3. plant

ตารางที่ 3-3 plant

Attribute Name	description	data type	length	key type	nullable
plant_name	ชื่อ plant	character varying	225		NOT NULL
plant_id	AUTO_INCREMENT Plant id	integer	32	PK	NOT NULL

ตารางที่ 3-3 ประกอบด้วย 2 Attribute ซึ่งจะประกอบด้วย

plant_name ตัวแปร character varying บันทึก ชื่อ plant ไม่เป็นค่า NULL

plant_id ตัวแปร integer บันทึก AUTO_INCREMENT เป็น PRIMARY key ไม่เป็นค่า NULL

plant_name มีหน้าที่เก็บชื่อ plant สำหรับการสร้าง json

plant_id มีหน้าที่เก็บข้อมูล id ของแต่ละ feature เพื่อใช้เป็น PRIMARY key สำหรับเชื่อม

ตาราง report

3.4.4. report

ตารางที่ 3-4 report

Attribute Name	description	data type	length	key type	nullable
T4	จำนวนครั้ง ของ T4	integer	32		NOT NULL
T2	จำนวนครั้ง ของ T2	integer	32		NOT NULL
T5	จำนวนครั้ง ของ T5	integer	32		NOT NULL
T6	จำนวนครั้ง ของ T6	integer	32		NOT NULL
T8	จำนวนครั้ง ของ T8	integer	32		NOT NULL
T3	จำนวนครั้ง ของ T3	integer	32		NOT NULL
T7	จำนวนครั้ง ของ T7	integer	32		NOT NULL
Date	วันที่	date	3		NOT NULL

ตารางที่ 3-4 ประกอบด้วย 8 Attribute ซึ่งจะประกอบด้วย

T4 ตัวแปร integer บันทึก จำนวนครั้ง ของ T4 ไม่เป็นค่า NULL

T2 ตัวแปร integer บันทึก จำนวนครั้ง ของ T2 ไม่เป็นค่า NULL

T5 ตัวแปร integer บันทึก จำนวนครั้ง ของ T5 ไม่เป็นค่า NULL

T6 ตัวแปร integer บันทึก จำนวนครั้ง ของ T6 ไม่เป็นค่า NULL

T8 ตัวแปร integer บันทึก จำนวนครั้ง ของ T8 ไม่เป็นค่า NULL

T3 ตัวแปร integer บันทึก จำนวนครั้ง ของ T3 ไม่เป็นค่า NULL

T7 ตัวแปร integer บันทึก จำนวนครั้ง ของ T7 ไม่เป็นค่า NULL

Date ตัวแปร date บันทึก วันที่ ไม่เป็นค่า NULL

T4 มีหน้าที่เก็บข้อมูล unit T4 เกี่ยวกับจำนวนการเกิด feature drift เพื่อใช้ในการทำสรุปรายอาทิตย์

T2 มีหน้าที่เก็บข้อมูล unit T2 เกี่ยวกับจำนวนการเกิด feature drift เพื่อใช้ในการทำสรุปรายอาทิตย์

T5 มีหน้าที่เก็บข้อมูล unit T5 เกี่ยวกับจำนวนการเกิด feature drift เพื่อใช้ในการทำสรุปรายอาทิตย์

T6 มีหน้าที่เก็บข้อมูล unit T6 เกี่ยวกับจำนวนการเกิด feature drift เพื่อใช้ในการทำสรุปรายอาทิตย์

T8 มีหน้าที่เก็บข้อมูล unit T8 เกี่ยวกับจำนวนการเกิด feature drift เพื่อใช้ในการทำสรุปรายอาทิตย์

T3 มีหน้าที่เก็บข้อมูล unit T3 เกี่ยวกับจำนวนการเกิด feature drift เพื่อใช้ในการทำสรุปรายอาทิตย์

T7 มีหน้าที่เก็บข้อมูล unit T7 เกี่ยวกับจำนวนการเกิด feature drift เพื่อใช้ในการทำสรุปรายอาทิตย์

Date มีหน้าที่เก็บข้อมูล วันที่ และเวลา การส่งข้อมูลเข้ามาในระบบฐานข้อมูลสำหรับอ้างอิงเวลาเรียกดู

3.4.5. report2

ตารางที่ 3-5 report2

Attribute Name	description	data type	length	key type	nullable
featurename_id	AUTO_INCREMENT Featurename id	integer	32	FK	NOT NULL
timeoccu	จำนวนครั้งที่เกิด	integer	32		NULL
plant_id	ลำดับ plant	integer	32	FK	NULL
date	วันที่	date	3		NOT NULL

ตารางที่ 3-5 ประกอบด้วย 4 Attribute ซึ่งจะประกอบด้วย

featurename_id ตัวแปร integer บันทึก AUTO_INCREMENT เป็น PRIMARY key ไม่เป็นค่า NULL

timeoccu ตัวแปร integer บันทึกจำนวนครั้งที่เกิด เป็นค่า NULL ได้

plant_id ตัวแปร integer บันทึกลำดับ plant เป็นค่า NULL ได้

date ตัวแปร date บันทึก วันที่ ไม่เป็นค่า NULL

featurename_id มีหน้าที่ระบุ id ของแต่ละ featurename เพื่อใช้เป็น FOREIGN key ในการเชื่อม ด้วย ตาราง featurename

timeoccu มีหน้าที่เก็บจำนวนครั้งการเกิด opportunity loss ของ plant นั้นๆ

plant_id มีหน้าที่ระบุ id ของ plant ที่เกิด opportunity loss เพื่อใช้เป็น FOREIGN key ในการเชื่อม ด้วย ตาราง plant

date มีหน้าที่เก็บข้อมูล วันที่ และเวลา การส่งข้อมูลเข้ามาในระบบฐานข้อมูลสำหรับอ้างอิงเวลา เรียกว่า

3.4.6. tag

ตารางที่ 3-6 tag

Attribute Name	description	data type	length	key type	nullable
tag_name	ชื่อ tag	character varying	225		NOT NULL
tag_id	AUTO_INCREMENT Tag id	integer	32	PK	NOT NULL

ตารางที่ 3-6 ประกอบด้วย 2 Attribute ซึ่งจะประกอบด้วย

tag_name ตัวแปร character varying บันทึก ชื่อ tag ไม่เป็นค่า NULL

tag_id ตัวแปร integer บันทึก AUTO_INCREMENT เป็น primary key ไม่เป็นค่า NULL

tag_name เก็บข้อมูล name ของแต่ละ tag เพื่อใช้ในการระบุชื่อ tag ใน Data Frame

tag_id เก็บข้อมูล id ของแต่ละ tag เพื่อใช้เป็น PRIMARY key

3.4.7. feature

ตารางที่ 3-7 feature

Attribute Name	description	data type	length	key type	nullable
feature_max	ย่าน max	numeric	225		NOT NULL
feature_min	ย่าน min	numeric	225		NOT NULL
feature_id	AUTO_INCREMENT Feature id	integer	32	PK	NOT NULL
unit	หน่วย	character varying	225		NULL

ตารางที่ 3-7 ประกอบด้วย 4 Attribute ซึ่งจะประกอบด้วย

feature_max ตัวแปร numeric บันทึก ย่าน max ที่เกิด ไม่เป็นค่า NULL

feature_min ตัวแปร numeric บันทึก ย่าน min ที่เกิด ไม่เป็นค่า NULL

feature_id ตัวแปร integer บันทึก AUTO_INCREMENT เป็น primary key ไม่เป็นค่า NULL

unit ตัวแปร character varying บันทึก หน่วย เป็นค่า NULL ได้

feature_max มีหน้าที่เก็บข้อมูล max ของแต่ละ feature เพื่อใช้เป็นเกณฑ์ในการตัดสินใจว่า เป็น feature drift

feature_min มีหน้าที่เก็บข้อมูล min ของแต่ละ feature เพื่อใช้เป็นเกณฑ์ในการตัดสินใจว่า เป็น feature drift

feature_id มีหน้าที่เก็บข้อมูล id ของแต่ละ min-max เพื่อใช้เป็น PRIMARY key สำหรับอ้างอิง เวลาเรียกดู

unit มีหน้าที่ มีหน้าที่เก็บข้อมูล id ของแต่ละ feature

หลังจากอ้างอิงทฤษฎีแล้ว จะต้องสามารถสรุปขั้นตอนและวิธีการดำเนินการได้ ไม่ว่าจะเป็นการอธิบายภาพรวมของระบบที่จะกล่าวถึง โครงสร้างพื้นฐานของตัวระบบ องค์ประกอบ และผู้ใช้งาน อันเป็นรากฐานสำคัญ การสรุปขั้นตอนการทำให้เปรียบเทียบเหมือนการสรุป algorithm ของระบบว่าจะควรทำงานอย่างไร อีกทั้งการสรุปความสัมพันธ์ และหน้าที่โดยหลังด้วย ER Diagram ที่ช่วยในการวางแผนการออกแบบฐานข้อมูลที่จะเป็น และ อธิบายรายละเอียดต่างๆในฐานข้อมูล ด้วย Data Dictionary หลังจากนั้นจะเป็นการลงมือปฏิบัติในการพัฒนาระบบเพื่อให้ได้ผลลัพธ์ที่ต้องการ

บทที่ 4

การพัฒนาระบบ

ในบทนี้จะกล่าวถึงการออกแบบและเครื่องมือต่างๆ ที่ใช้ในการพัฒนาระบบ การพัฒนาระบบของโครงการที่กล่าวถึงตัวโค้ดที่สำคัญของทุกส่วนในระบบ และ ตารางการปฏิบัติงานตลอด 6 เดือนในช่วงการทำสหกิจศึกษา โดยจะลงรายละเอียดที่สำคัญดังนี้

4.1. การออกแบบและเครื่องมือต่างๆ

การพัฒนาระบบ สามารถนำข้อมูลที่ได้มา ประยุกต์ใช้และออกแบบการทำงานของระบบเพื่อใช้เป็นแนวทางในการพัฒนาระบบให้เป็นไปอย่างมีประสิทธิภาพ ในบทนี้จะแสดงรายละเอียดของการออกแบบและเครื่องมือต่างๆ ที่ใช้ในการพัฒนาระบบดังนี้

4.1.1. Software

เป็นส่วนของซอฟต์แวร์ที่ใช้ในการพัฒนาระบบนี้

1) Visual Studio (Editor) เป็นโปรแกรมหลักที่ใช้ในการพัฒนาระบบ

2) PostgreSQL (Chaudhuri, 2022) เป็นโปรแกรมที่ใช้ในการดูแลจัดการฐานข้อมูลและยังสามารถใช้ทดสอบระบบฐานข้อมูล

4.1.2. Programming Language

เป็นส่วนของภาษาที่ใช้ในการพัฒนาระบบ

1) Python Version 3.7 ภาษาหลักในการพัฒนาระบบ

2) SQL ภาษาที่ใช้ในการพัฒนาระบบฐานข้อมูล

4.1.3. Tools

เป็นส่วนของ Module หรือ library ที่ใช้ในการพัฒนาระบบ

- 1) Pandas 1.3.5 เป็น module มีหน้าที่ใช้ในการจัดการอ่านค่า csv และ การรับค่าจาก pivot แบบ Dataframe
- 2) NumPy 1.21.6 ใช้ในการแปลงค่า Dataframe เป็น Numpy เพื่อใช้ในการทำเกี่ยวกับ Machine Learning
- 3) scikit-learn 1.0.2 ใช้ในการสร้าง model สำหรับ Machine learning
- 4) requests[security] 2.28.1 ใช้ในการติดต่อ server ต่างๆ แต่จะเพิ่ม security module ที่จำเป็น
- 5) requests-ntlm 1.1.0 ใช้ในการติดต่อ server ด้วยการ ยืนยันตัวตนในการติดต่อกับ pivot API

4.1.4. Hardware

เนื่องจากนโยบายของบริษัทที่ทำโครงการจึงต้องใช้โน้ตบุ๊กของบริษัทเท่านั้นเพื่อป้องกันข้อมูลที่เป็นความลับของบริษัท

- 1) notebook Brand : dell
- 2) CPU Intel Core i7 processor
- 3) GPU on board
- 4) Memory 8 GB
- 5) Solid State Drive 500 G

4.1.5. service

เป็นส่วนของ service ที่ใช้ในการพัฒนาระบบซึ่งส่วนใหญ่เป็นของMicrosoft

- 1) Azure Kubernetes เป็น cloud sever service ใช้ deploy งานลงระบบ เพื่อให้สามารถทำงาน
- 2) PI view หรือ plant information system เป็น เป็นระบบที่แสดงข้อมูล plant ของโรงงานสามารถ query ข้อมูลได้
- 3) Microsoft team webhook service เป็น service ในการส่ง message ไปยัง Microsoft Team Channel

4.2. การพัฒนาระบบของโครงการ

ในการพัฒนาระบบประกอบไปด้วย 3 ส่วน คือ

4.2.1. ส่วน WEBHOOK

Develop Virtual Assistant (unit3+unit4)

พัฒนาระบบ Monitor gap benefit ให้สามารถส่ง notification รายกะได้ และยังแสดงความถี่ที่เกิดได้ พร้อมมีปุ่ม เข้าดู plant information ได้ โดยเริ่มที่ไฟล์ key_vault.py ที่จะนำมาใช้งานในไฟล์ที่เหลือถัดไป

key_vault.py:

เป็นไฟล์ที่มีหน้าที่ในการ ติดต่อกับ Azure ในการ deploy ระบบลงใน cloud server โดยจะมีการนำเข้า library ที่สำคัญดังตารางที่ 4-1

ตารางที่ 4-1 ส่วน library ที่ใช้ใน key vault

1	<code>from azure.identity import ClientSecretCredential</code>
2	<code>from azure.keyvault.secrets import SecretClient</code>
3	<code>import json</code>

บรรทัดที่ 1 นำเข้าไฟล์ ClientSecretCredential จาก azure.identity

บรรทัดที่ 2 นำเข้าไฟล์ SecretClient จาก azure.keyvault.secrets

บรรทัดที่ 3 นำเข้าไฟล์ json

หลังจากทำการนำเข้าไฟล์ library เรียบร้อยแล้ว จากนั้นทำการกำหนดค่า id url และ secret เป็นสิ่งที่ใช้ในการ authenticate กับ Azure ดังตารางที่ 4-2

ตารางที่ 4-2 ส่วน ค่ากำหนดเริ่มต้นที่ใช้ใน key vault

4	<code>tenant_id = "xxxxxxxxxxxx"</code>
5	<code>vault_url = "https://chemdmkeyvault.vault.azure.net/"</code>
6	<code>client_id = "xxxxxxxxxxxx"</code>
7	<code>client_secret = "xxxxxxxxxxxx"</code>

บรรทัดที่ 4 กำหนดไอดีสำหรับ piwebapi

บรรทัดที่ 5 กำหนด URL server สำหรับ piwebapi

บรรทัดที่ 6 กำหนดไอดีของ client สำหรับ SERVICE

บรรทัดที่ 7 กำหนด secret สำหรับเชื่อมต่อ key vault

ในส่วนต่อไปเป็นการกำหนด function ในการ เรียกค่า secret ขึ้นมาโดยคือค่าเป็นรูปแบบ tag username password ดังตารางที่ 4-3

ตารางที่ 4-3 ส่วน function ในการ เรียก key มาใช้ในการ ติดต่อ piview

8	<code>def get_secret(secret_name: str):</code>
10	<code>token = ClientSecretCredential(tenant_id=tenant_id, client_id=client_id, client_secret=client_secret)</code>
12	<code>client = SecretClient(vault_url=vault_url, credential=token)</code>
15	<code>secret = client.get_secret(secret_name)</code>
19	<code>return secret['usr'], secret['pwd']</code>

บรรทัดที่ 8 ประกาศชื่อฟังก์ชัน get_secret รับ parameter secret_name เป็น str

บรรทัดที่ 10 ทำการยืนยันตัวตน กับ Azure Active Directory โดยใช้ CLIENT ID และ CLIENT CERTIFICATE และเก็บใน token

บรรทัดที่ 12 ทำการยืนยันตัวตน กับ KEY VAULT โดยใช้ Azure Active Directory TOKEN

บรรทัดที่ 15 เรียกค่า KEY VAULT โดยใช้ฟังก์ชัน GET SECRET

บรรทัดที่ 19 คืนค่า SECRET

ในตารางที่ 4-4 เป็นการประกาศ main ของไฟล์นี้เมื่อไฟล์ถูกเรียกทุกบรรทัดจะทำงาน

ตารางที่ 4-4 ส่วน main

20	#CALL MAIN()
21	if __name__ == "__main__":
22	usr, pwd = get_secret('xxxxservice')

บรรทัดที่ 21 เช็คเงื่อนไขหาก __name__ = "__main__"

บรรทัดที่ 22 ทำการเรียกฟังก์ชัน get_secret

เมื่อเสร็จสิ้นในส่วนของ key_vault.py จะต่อด้วย Al_12hr_feature_drift_webhook.py ซึ่งเป็น 1 ใน 2 ส่วนหลักที่สำคัญของ webhook เนื่องจากมีหน้าที่ในการเรียกดูและ ประเมิน ค่าจาก piwebapi

AI_12hr_feature_drift_webhook.py:

เริ่มจากการนำเข้าlibrary ที่ต้องใช้งานทั้งหมดไม่ว่าจะเป็น datetime ในการจัดการด้านเวลา เครื่อง numpy ในการจัดการdata จาก piwebapi pandas ในการจัดการ Dataframe dbreader เป็นmodule ในการติดต่อ database ดังตารางที่ 4-5

ตารางที่ 4-5 ส่วน libery ที่ใช้ใน AI_12hr_feature_drift_webhook

8	<code>import numpy as np</code>
9	<code>import pandas as pd</code>
11	<code>import requests</code>
12	<code>import json</code>
14	<code>import dbreader</code>

บรรทัดที่ 8 นำเข้าไฟล์ numpy โดยใช้ชื่อ np

บรรทัดที่ 9 นำเข้าไฟล์ pandas โดยใช้ชื่อ pd



บรรทัดที่ 11 นำเข้าไฟล์ requests

บรรทัดที่ 12 นำเข้าไฟล์ json

บรรทัดที่ 14 นำเข้าไฟล์ dbreader

จากตารางที่ 4-6 หลังจากนำเข้าไฟล์ต่างๆแล้ว จากนั้นประกาศค่าlistต่างๆที่จะใช้ รวมถึงค่า string firstpart สำหรับ ช่วงต้น message และ finalpart ที่เป็นช่วงท้ายของ message

ตารางที่ 4-6 กำหนดค่าเริ่มต้นสำหรับการทำงาน

27	#define initial value for data
28	outrangevalue=[]#for store out of range feature name
29	outrange=[]#for store value that out of range
30	unit=[]#store unit of out of range feature
31	message="#"#store string
32	#initial first and second part of string
33	firstpart ='For last 12 hour \n\n'
34	finalpart='\n\n **Please consider.   **'

บรรทัดที่ 28 เก็บชื่อfeatureที่ออกนอกขอบเขตในlistชื่อoutrangevalue

บรรทัดที่ 29 เก็บค่าที่ออกนอกขอบเขตในlistชื่อoutrange

บรรทัดที่ 30 เก็บหน่วยออกนอกขอบเขตในlistชื่อunit

บรรทัดที่ 31 ประกาศstringชื่อmessage

บรรทัดที่ 33 ประกาศstringชื่อfirstpart

บรรทัดที่ 34 ประกาศstringชื่อfinalpart

การที่จะใช้ module จะต้องทำการประกาศ objectจาก class module นั้น เพื่อให้สามารถเรียกใช้ในไฟล์นั้นๆได้ ดังตารางที่ 4-7

ตารางที่ 4-7 สร้าง obj จาก class module

44	#create object from class module
45	DB=dbreader.DB()

บรรทัดที่ 45 ประกาศ object จาก class module

โดยหากเรียกฟังก์ชันที่ต้องการสามารถทำได้โดยอาศัย ตัวแปรที่ประกาศก่อนหน้านั้นมาอ้างอิง โดยจากตารางที่ 4-8 ต้องการประกาศค่า tag จึงทำการเรียก ฟังก์ชัน readtag เพื่ออ้างอิง object ในการดึงค่าtag

ตารางที่ 4-8 ประกาศค่าสำหรับ piwebapi

51	tag_list=DB.readtag();
----	------------------------

บรรทัดที่ 51 ทำการอ่านค่าจากdatabase ลง tag_list

ในตารางที่ 4-9 นี้จะเป็นการตั้งค่าที่จำเป็นจะต้องใช้ในการเรียกดูข้อมูลจาก piwebapi เช่นการกำหนดให้ การคำนวณเป็นแบบ เฉลี่ย โดยอาศัย ค่า Timeweighted ในช่วง 12 ชมที่ผ่านมา

ตารางที่ 4-9 กำหนดค่าสำหรับการrequest piwebapi

53	#define value for use to request data piwebapi
54	cal_type = ['Average']
55	cal_basis = 'TimeWeighted'
56	interval='12h'

บรรทัดที่ 54 ประกาศ calculate type เป็น average (ค่าเฉลี่ย)

บรรทัดที่ 55 ประกาศ calculate type เป็น Timeweighted (weighted ด้วยเวลา)

บรรทัดที่ 56 ประกาศ interval เป็น 12 ชม

เมื่อตั้งค่าเสร็จสิ้น จากนั้นจะทำการ request ข้อมูลจาก piwebapi ตามตารางที่ 4-10

ตารางที่ 4-10 ทำการ request piwebapi

62	# Get average value for a list of calendar tags.(request data)
63	df = webapi.getAverageValue(tag_list,start,end,cal_basis,cal_type,interval)

บรรทัดที่ 63 ทำการเรียก ฟังก์ชัน getaveragevalue

จาก ตารางที่ 4-11 ขั้นตอนนี้จะเป็นการคำนวณค่า column ต่างๆที่จำเป็นต้องใช้ และลบ column ที่ไม่จำเป็นอีกแล้วออกไป เพื่อให้ง่ายต่อการนำมาตรวจสอบ ในขั้นตอนต่อไป

ตารางที่ 4-11 ทำการ คำนวณค่าและลบcolumnไม่จำเป็น

70	#calculate value and then drop column that not use
71	#fea11 to T-5 + fea11 to T-5 B
72	df['fea11 to T-5']=df['fea11 to T-5']+df['fea11 to T-5B']
73	df=df.drop('fea11 to T-5B',axis=1)
74	#fea11 to T-6 + fea11 to T-6 B
75	df['fea11 to T-6']=df['fea11 to T-6']+df['fea11 to T-6B']
76	df=df.drop('fea11 to T-6B',axis=1)
77	#fea10 + fea10 B
78	df['fea10']=df['fea10']+df['fea10-B']
79	df=df.drop('fea10-B',axis=1)

บรรทัดที่ 72 ทำการบวกค่า Dataframe column กับ ค่า Dataframe column ลงใน dataframe column

บรรทัดที่ 73 ทำการ drop Dataframe column ทั้งหมด

บรรทัดที่ 75 ทำการบวกค่า Dataframe column กับ ค่า Dataframe column ลงใน dataframe column

บรรทัดที่ 76 ทำการ drop Dataframe column ทั้งหมด

บรรทัดที่ 78 ทำการบวกค่า Dataframe column กับ ค่า Dataframe column ลงใน dataframe columne

บรรทัดที่ 79 ทำการdrop Dataframe column ทั้งหมด

บรรทัดที่ 83 ทำการตัด columnhead ออกเนื่องจากไม่ได้ใช้

ต่อไปจะทำการ แยก feature ที่เป็น input ออกจาก output โดยแบ่งเป็น x และ y และ ทำการสร้าง list คำตอบขึ้นมารองรับ ดังตารางที่ 4-12

ตารางที่ 4-12 loop สำหรับการเพิ่มค่าประเภท X ,Y ลงlist

95	<code>for val in range(len(value)):</code>
97	<code>if(val < (s[i]-Y[i])):</code>
98	<code>xy.append('X')#append x to list XY</code>
99	<code>else:</code>
100	<code>xy.append('Y')#append Y to list XY</code>
101	<code>i=i+1</code>
102	<code>ans.append(0)#test 1 , o default</code>
104	<code>count=0#set count to 0 as start</code>

บรรทัดที่ 95 รูป โดยที่ j โดยมี range เป็นขนาดvalue

บรรทัดที่ 97 เช็คเงื่อนไขว่า val น้อยกว่า (list ของs(size) ที่ index i) - (list ของ ycount ที่ index i)

บรรทัดที่ 98 ทำการ append ค่า 'X' เข้า list XY

บรรทัดที่ 100 ทำการappend ค่า 'Y' เข้า list XY

บรรทัดที่ 101 กำหนดค่า ด้วย i+ 1

บรรทัดที่ 102 ทำการappend ค่า 0 เข้า list ans

บรรทัดที่ 104 กำหนดค่า count = 0

ในขั้นนี้จะเป็นการแยก feature drift จาก ค่าแต่ละตัวโดยการดูจาก min max ที่กำหนดไว้ ซึ่งได้จากการเรียกอ่านจาก DataBase โดยหากตัวใดเกินก็จะทำการบันทึกคำตอบ ขณะเดียวกัน ทำการส่งค่าไปยัง module เพื่อทำการบันทึก ลง Database ดังตารางที่ 4-13

ตารางที่ 4-13 loop สำหรับเช็ค min-max ในและส่งค่าไปยัง DB

105	#for loop to insert data to database by check min max each feature
106	for j in range(len(value)):#for j start at 0 to size of value - 1
107	max=[data for data, in DB.getmax(j+1)]
108	min=[data for data, in DB.getmin(j+1)]
109	max=float(max[0])#convert to max
110	min=float(min[0])#convert to min
111	if(value[j]>max):
112	ans[count]=1
113	DB.inserts2(date.today(),1,(j+1))
114	elif(value[j]<=min):
115	ans[count]=-1#override list of ans at count index to -1 for min
116	DB.inserts2(date.today(),1,(j+1))
117	
118	else:
119	ans[count]=0#override list of ans at count index to 0 for min
120	DB.inserts2(date.today(),0,(j+1))
121	count=count+1#count

บรรทัดที่ 106 ลูป โดยที่ j โดยมี range เป็นขนาดvalue
 บรรทัดที่ 107 อ่านค่า max ของ feature จาก database
 บรรทัดที่ 108 อ่านค่า min ของ feature จาก database

บรรทัดที่ 109	แปลงค่า max เป็น float
บรรทัดที่ 110	แปลงค่า min เป็น float
บรรทัดที่ 111	เช็คเงื่อนไข list ของ value ที่ index j มีค่ามากกว่าค่า max ที่ได้จากการอ่าน
บรรทัดที่ 112	กำหนดค่าให้ ans ตัวที่ count ด้วย 1
บรรทัดที่ 113	ทำการเรียก ฟังก์ชัน inserts2
บรรทัดที่ 114	หากเช็คเงื่อนไข บรรทัดที่ 111 แล้วไม่เป็นจริงให้เช็คเงื่อนไข list ของ value ที่ index j มีค่าน้อยกว่าหรือเท่ากับค่า min ที่ได้จากการอ่าน
บรรทัดที่ 115	กำหนดค่าให้ ans ตัวที่ count ด้วย 1
บรรทัดที่ 116	ทำการเรียก ฟังก์ชัน inserts2
บรรทัดที่ 119	กำหนดค่าให้ ans ตัวที่ count ด้วย 0
บรรทัดที่ 120	ทำการเรียก ฟังก์ชัน inserts2
บรรทัดที่ 121	กำหนดค่า count ด้วยค่า count + 1

หลังจากทำการแยก feature drift ออกจาก x y ในตารางที่ 4-14 เป็นการกำหนดว่ามีค่า feature drift หรือไม่

ตารางที่ 4-14 เช็คเงื่อนไขว่า มี 1 หรือ -1 ใน ans หรือไม่

124	<code>if((1 in ans) or (-1 in ans)):# if ans dont contain 1 or -1 do inside</code>
125	<code>status=True #set true in status</code>

บรรทัดที่ 124 เช็คเงื่อนไข ว่า มี 1 หรือ -1 ใน list ans หรือไม่

บรรทัดที่ 125 กำหนดค่า status ด้วย true

ตารางที่ 4-15 ในส่วนนี้จะเป็นส่วนที่ เป็นการสรุปข้อมูลต่างๆ ออกมาเตรียมไว้สำหรับการเตรียมสร้าง message ไม่ว่าจะเป็น plantname outrangevalue outrangevalueclass getmin max ดังนี้

ตารางที่ 4-15 loopสำหรับจัดเตรียมข้อมูลต่างๆ ในการสร้าง message

126	<code>for index in range(len(ans)):</code>
127	<code>plantindex=DB.findplantindex(index+1)</code>
128	<code>planttuple=[data for data, in DB.getplantname(plantindex)]</code>
129	<code>plantname.append(str(planttuple[0]))</code>
130	<code>if (ans[index] == 1):</code>
131	<code>info.append([data for data, in DB.getmax(index+1)])</code>
132	<code>outrangevalue.append(valuenamename[index])</code>
133	<code>dex=len(str([data for data, in DB.getmax(index+1)][0]).split("."))</code> <code>[1])</code>
134	<code>outrange.append(value[index].round(dex))</code>
135	<code>unit.append([data[0] for data, in DB.getunit(index+1)])</code>
136	<code>if (xy[index]== 'X'):</code>
137	<code>outrangevalue.append('X')#append class to list X</code>
138	<code>elif (xy[index]== 'Y'):#if value of list of xy at index name index is equal Y do inside</code>

บรรทัดที่ 126 ลูป index โดยที่ range = ขนาดของ ans - 1

บรรทัดที่ 127 ประกาศ plantindex โดยเรียกฟังก์ชัน findindex

บรรทัดที่ 128 ประกาศ planttupleโดยกำหนดค่าด้วย dataที่ได้จากฟังก์ชัน getplantname

บรรทัดที่ 129 ทำการ append string ของ plant name ที่มาจาก plant tuple

บรรทัดที่ 130 เช็คเงื่อนไขว่าค่าของ ans ที่ index นั้นมีค่าเท่ากับ 1

บรรทัดที่ 131 ทำการ append info จาก data ที่ได้จากฟังก์ชัน getmax

บรรทัดที่ 132 ทำการ append string ของ outrangevalue ที่ index

บรรทัดที่ 133 ประกาศ dex ที่เก็บ decimal point จากการวัดขนาดของ string ของ (data ที่ได้จากฟังก์ชัน getmax แบ่งโดย point)

บรรทัดที่ 134	ทำการ append value ที่ผ่านการ rounding ด้วย deximal point
บรรทัดที่ 135	ทำการ append data จาก data ที่ได้จากฟังก์ชัน getunit
บรรทัดที่ 136	เช็คเงื่อนไขว่าค่าของ xy ที่ index นั้นมีค่าเท่ากับ X
บรรทัดที่ 137	ทำการ append ค่า X
บรรทัดที่ 138	เช็คเงื่อนไขว่าค่าของ xy ที่ index นั้นมีค่าเท่ากับ

ตารางที่ 4-16 นอกจากการเก็บชื่อ plant แบบเดียวกับตาราง 4-14 แต่มีการ เก็บค่าประเภทเป็น x หรือ y

ตารางที่ 4-16 loopสำหรับจัดเตรียมข้อมูลต่างๆ ในการสร้าง message (ต่อ)

139	outrangeclass.append('Y')#append class to list Y
140	elif (ans[index] == -1):#else if ans at index name index is equal -1
141	info.append([data for data, in DB.getmin(index+1)])
142	outrangevalue.append(valuename[index])
143	dex=len(str([data for data, in DB.getmin(index+1)][0]).split(".")) [1])
144	outrange.append(value[index].round(dex))
145	unit.append([data for data, in DB.getunit(index+1)])
146	if (xy[index]== 'X'):
147	outrangeclass.append('X')#append class to list X
148	elif (xy[index]== 'Y'):
149	outrangeclass.append('Y')#append class to list Y
150	else:
151	outrangevalue.append(valuename[index
152	outrangeclass.append('Z')#append Z as normal (other is ab normal)

- บรรทัดที่ 139 ทำการ append ค่า Y
- บรรทัดที่ 140 เช็คเงื่อนไขค่าของ ans ที่ index นั้นมีค่าเท่ากับ -1
- บรรทัดที่ 141 ทำการ append info จาก data ที่ได้จากฟังก์ชัน getmin
- บรรทัดที่ 142 ทำการ append string ของ outrangevalue ที่ index
- บรรทัดที่ 143 ประกาศ dex ที่เก็บ deximal point จากการวัดขนาดของ string
ของ (data ที่ได้จากฟังก์ชัน getmax แบ่งโดย point)
- บรรทัดที่ 144 ทำการ append value ที่ผ่านการ rounding ด้วย deximal point
- บรรทัดที่ 145 ทำการ append data จาก data ที่ได้จากฟังก์ชัน getunit
- บรรทัดที่ 146 เช็คเงื่อนไขว่าค่าของ xy ที่ index นั้นมีค่าเท่ากับ X
- บรรทัดที่ 147 ทำการ append outrangeclass ด้วย X
- บรรทัดที่ 148 เช็คเงื่อนไขว่าค่าของ xy ที่ index นั้นมีค่าเท่ากับ Y
- บรรทัดที่ 149 ทำการ append outrangeclass ด้วย Y
- บรรทัดที่ 151 ทำการ append string ของ outrangevalue ที่ index
- บรรทัดที่ 152 ทำการ append outrangeclass ด้วย Z

หลังข้อมูลทุกที่จำเป็นต้องครบแล้ว ต่อไปจะเป็นการสร้าง message ซึ่งต้องใช้การสร้าง json object โดยมี องค์ประกอบดังตารางที่ 4-17

ตารางที่ 4-17 loop สำหรับการสร้าง message

169	<code>elif(plantindex not in tempplant):</code>
170	<code>message =message+'- '+plantname[j]+' \r'</code>
171	<code>tempplant.append(plantindex)#after add plant index add plant index to temp plant</code>
172	<code>if(outrange[j]!='X' or outrange[j]!='Y'):</code>
173	<code>no=no+1</code>
174	<code>if(plantindex not in xtemp and outrange[j]!='X'):</code>
176	<code>sentent=sentent+"\n\n X: \r'</code>
177	<code>xtemp.append(plantindex)# add plant index to Xtemp</code>
178	<code>elif(plantindex not in ytemp and outrange[j]!='Y'):</code>
180	<code>sentent=sentent+"\n\n Y: \r"</code>
181	<code>ytemp.append(plantindex)</code>
182	<code>if(ans[j]==1):# if ans at index j is equal 1</code>
183	<code>temp=chr('r '+str(no)+' '+outrangevalue[j])+' is = '+str('{:,.}').format(outrange[j]))+' '+unit[j][0]+' [max='+str(float(info [j][0]))+' '+unit[j][0]+'']</code>
184	<code>sentent=sentent+temp</code>
185	<code>elif(ans[j]==-1):# if ans at index j is equal -1</code>
186	<code>temp=chr('r '+str(no)+' '+outrangevalue[j])+ ' is = '+str('{:,.}').format(outrange[j]))+' '+unit[j][0] ' + [min='+str(float(info[j][0]))+' '+unit[j][0]+'']</code>
189	<code>if(status==False):# if status is equal false do inside</code>
190	<code>message = 'For last 12 hour,there is no value out of range \n\n **Good Job and have a Great Day!!! 😊👤 **'</code>

บรรทัดที่ 169	ทำการเช็คเงื่อนไข index ของ plant อยู่ในค่า temp plant หรือไม่
บรรทัดที่ 170	ทำการ joint string message ด้วย plant name ที่ index j
บรรทัดที่ 171	ทำการ append plant index ไปยัง temp plant
บรรทัดที่ 172	ทำการเช็คเงื่อนไขว่าเป็น class x หรือ class y หรือไม่
บรรทัดที่ 173	ทำการ no+1
บรรทัดที่ 174	ทำการเช็คเงื่อนไขว่า plant index ไม่อยู่ใน temp plant หรือไม่
บรรทัดที่ 176	ทำการ joint string message ด้วย X
บรรทัดที่ 177	ทำการ append plant index ไปยัง Xtemp
บรรทัดที่ 178	ทำการเช็คเงื่อนไขว่า plant index ไม่อยู่ใน temp plant หรือไม่
บรรทัดที่ 180	ทำการ joint string message ด้วย Y
บรรทัดที่ 181	ทำการ append plant index ไปยัง Ytemp
บรรทัดที่ 182	ทำการเช็คเงื่อนไขว่า ans ที่ index j เท่ากับ 1
บรรทัดที่ 184	ทำการ joint string temp กับ sentent
บรรทัดที่ 185	ทำการเช็คเงื่อนไขว่า ans ที่ index j เท่ากับ -1
บรรทัดที่ 189	ทำการเช็คเงื่อนไขว่า status เป็น false หรือไม่

หลังจากที่ได้ message แล้ว เราจะนำมาใส่ในส่วนของ text ในบรรทัดที่ 208 ของ body โดยกำหนด content type adaptive card ซึ่งทั้งหมดจะถูกกำหนดเป็นโครงสร้างใน payload card ดังตารางที่ 4-18

ตารางที่ 4-18 ประกาศโครงสร้างของ payload card โดยใส่ message เข้าไป

195	payload_card={
196	"type": "message",
197	"attachments": [{
198	"contentType": "application/vnd.microsoft.card.adaptive",
	⋮
200	"content": {
201	"\$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
202	"type": "AdaptiveCard",
	⋮
205	"body": [
206	{
207	"type": "TextBlock",
208	"text": message,
209	"wrap": True
212	}#,

เมื่อได้ โครงสร้าง json ที่มี text เป็น message แล้ว ต่อไปจะทำการนำไปเป็น parameter ในการทำ request ไปยัง income webhook service ดังตารางที่ 4-19

ตารางที่ 4-19 ทำการส่ง messagecard ไปยัง income service

225	# Sends a card to team channel
226	adaptive = json.dumps(payload_card)
227	msg =json.loads(adaptive)
228	card = requests.post('https://www.secret.scg.com/#/webhookb2',json =msg)

บรรทัดที่ 226 ทำการ dump เป็น json

บรรทัดที่ 227 ทำการ load json

บรรทัดที่ 228 ทำการส่งmessagecardไปยัง income service

Develop Virtual Assistant (ML Feature Drift)

พัฒนาระบบ Monitor AI feature driftให้สามารถส่ง notification รายกะได้ และยังแสดงย่านที่เกินได้พร้อมมีปุ่ม เข้าดู plant information ได้

AI_12hr_gap_benefit_webhook.py:

โดยในไฟล์นี้จะมีการนำเอาไฟล์ที่คล้ายกับ AI_12hr_feature_drift_webhook.py ตารางที่ 4-5 เช่น pandas request dbreader โดยมีลักษณะดังตารางที่ 4-20

ตารางที่ 4-20 ส่วน libery ที่ใช้ใน AI_12hr_gap_benefit_webhook

8	<code>import pandas as pd</code>
9	<code>import requests</code>
10	<code>import urllib3</code>
11	<code>from key_vault import get_secret</code>
12	<code>from piwebapi import PIWebAPI</code>
13	<code>#module</code>
14	<code>import dbreader</code>

บรรทัดที่ 8 นำเข้าไฟล์ pandas

บรรทัดที่ 9 นำเข้าไฟล์ requests

บรรทัดที่ 10 นำเข้าไฟล์ urllib3

บรรทัดที่ 11 นำเข้าไฟล์ get_secret จาก key_vault

บรรทัดที่ 12 นำเข้าไฟล์ piwebapi จาก PIWebAPI

บรรทัดที่ 14 นำเข้าไฟล์ dbreader

เมื่อทำการนำเข้าไฟล์เสร็จต่อไปเป็นการประกาศ tag ที่สำคัญ เพื่อใช้ในการนำเข้าข้อมูล piwebapi ดังตาราง 4-21

ตารางที่ 4-21 ประกาศ list ของ tag filename

27	tag_list=['U3F1.NX03','U3F2.NX13','U3F3.NX03', 'U3F4.NX13','U3F5.NX03','U3F6.NX13','U3F7.NX03']
----	--

บรรทัดที่ 27 ประกาศ list ของ tag ที่จะใช้

โดยนอกจากนี้ในการกำหนดค่าเริ่มต้นให้กับ piwebapi จะคล้ายกับไฟล์ AI_12hr_feature_drift_webhook.py ตารางที่ 4-9 แต่ต่างตรงที่ interval เปลี่ยน เป็น 10 นาที ดังตาราง ที่ 4-22

ตารางที่ 4-22 กำหนดค่าเริ่มต้นสำหรับการทำงาน

32	interval='10minute'
----	---------------------

บรรทัดที่ 32 ประกาศ interval เป็น 10 นาที

ตารางที่ 4-23 นั้นเมื่อกำหนดค่า piwebapi ครบแล้ว จะทำการ request ข้อมูลตาม tag ที่กำหนดไว้มา

ตารางที่ 4-23 ทำการrequest piwebapi 10 นาที

37	#request data 10 minute lost opportunity
38	df2 = webapi.getAverageValue(tag_list,start,end,cal_basis,cal_type,in terval)

บรรทัดที่ 38 ทำการเรียก ฟังก์ชัน get average value

จากนั้น ทำการกำหนดชื่อ column ของ Dataframe และทำการลบ column time ออก เนื่องจากไม่ได้ใช้ ตารางที่ 4-24

ตารางที่ 4-24 ประกาศค่า column และ drop time

39	#define column for pandas Dataframe
40	df2.columns = ['Time','T4 Opt Benefit', T2 Opt Benefitd','T5 Opt Benefit','T6 Opt Benefit', 'T8 Opt Benefit','T3 Opt Benefit','T7 Opt Benefit']
41	value2=df2.drop(['Time'],axis=1).values[:,:].astype(int)
42	value2

บรรทัดที่ 40 ประกาศ list column

บรรทัดที่ 41 ทำการ drop column ชื่อ time และแปลงค่าเป็นเลข

ตารางที่ 4-25 ทำการนำข้อมูลมาเทียบ ตามเกณฑ์ว่าเป็นจริงหรือไม่ จากนั้นจึงนำคำตอบไปบวก 0.166666667 นาที ที่กำหนดไว้เนื่องจากเรารounding ค่ามาเป็น ชม. (1hr = 0.166666667)

ตารางที่ 4-25 loop check ว่า ตัวไหนมีค่าเกิน 100000

52	if(value2[j][i]>100000):
54	ans2[i]=ans2[i]+0.166666667 #(1hr = 0.166666667)
55	count=count+1

บรรทัดที่ 52 เช็คเงื่อนไขว่า value2 row ที่ j และ column ที่ i มีค่ามากกว่า 100000 หรือไม่

บรรทัดที่ 54 ทำการบวกค่า ans2 ที่ i ด้วยค่า 0.166666667 minute

บรรทัดที่ 55 ทำการบวกค่า count ด้วย 1

จากนั้นทำการกำหนดค่าใหม่อีกรอบเพื่อ reset การตั้งค่าให้เป็นไปตามต้องการ โดยประกาศแบบเดียวกับตารางที่ 4-9 ดังตารางที่ 4-26 นี้

ตารางที่ 4-26 กำหนดค่าสำหรับการrequest piwebapi

63	tag_list=['U3F1.NX03','U3F2.NX13','U3F3.NX03', 'U3F4.NX13','U3F5.NX03','U3F6.NX13','U3F7.NX03']
65	df3 = pd.DataFrame()
66	df3['unit']=['T-4','T-2','T-5','T-6','T-8','T-3','T-7']
67	df3['time']=[0,0,0,0,0,0,0]
70	interval='12h'

บรรทัดที่ 63 ประกาศ list ของ tag ที่จะใช้

บรรทัดที่ 65 ประกาศ Dataframe df3

บรรทัดที่ 66 กำหนดค่า column unit ของ Dataframe df3

บรรทัดที่ 67 กำหนดค่า column time ของ Dataframe df3

บรรทัดที่ 70 ประกาศ interval เป็น 12 ชม

ต่อจากนี้จะเป็นการเรียกข้อมูล piwebapi มาอ่านโดยเก็บเป็น Dataframe ที่มีการกำหนด ชื่อ column ขณะเดียวกันมีการกำหนด Value plant โดยใช้ข้อมูลจาก Database ดังตาราง 4-27

ตารางที่ 4-27 ทำการrequest piwebapi 12 hr

75	#call data request 12hr lost
76	df = webapi.getAverageValue(tag_list,start,end,cal_basis,cal_type,interval)
77	df.columns = ['Time','T-4','T-2','T-5','T-6','T-8','T-3','T-7']
78	value=df.values[0][1:].astype(int)#get df type to integer
79	plant=df.columns[1:]#get plant name from columnname skip columnname time

บรรทัดที่ 76 ทำการเรียก ฟังก์ชัน getaveragevalue

บรรทัดที่ 77 ประกาศ list column

บรรทัดที่ 78 ทำการแปลงค่า Dataframe เป็นเลข

บรรทัดที่ 79 ทำการดึงค่า plant name จาก Dataframe

หลังจากได้ข้อมูลมาครบแล้ว จะเป็นขั้นตอนในส่วนนำข้อมูลมาเปรียบเทียบกับเกณฑ์ที่กำหนด หากใช้ทำการกำหนดค่าใน ans พร้อมเก็บจำนวน ชม ดังตาราง 4-27

ตารางที่ 4-28 loop เพื่อกำหนดค่า 0 ให้กลับ list ans

89	<code>if(i>100000):#check in 12 hr if which plant is lost more than 1000</code> <code>00</code>
90	<code>#print(count)</code>
91	<code>ans[count]=1#then add 1 to that plant in ans use index</code>
92	<code>count=count+1</code>
	...
95	<code>for i in ans2 :#put ans2 (hour) to list of hour count</code>
96	<code>if(i>0):</code>
97	<code>hour.append(i)#add hour to data count</code>
98	<code>else:</code>
99	<code>hour.append(-1)</code>

บรรทัดที่ 89 เช็คเงื่อนไขว่า i มีค่ามากกว่า 100000 หรือไม่

บรรทัดที่ 91 กำหนดค่า value ที่ i ด้วย 1

บรรทัดที่ 95 ลูป โดยที่ i โดยขอบเขตเป็นค่าใน ans2

บรรทัดที่ 96 เช็คเงื่อนไขว่า i มีค่ามากกว่า 0 หรือไม่

บรรทัดที่ 97 ทำการappend ค่า i เข้า list hour

บรรทัดที่ 99 ทำการappend ค่า -1 เข้า list hour

ตารางที่ 4-29 ทำการตรวจว่าค่าในทั้งหมดมี ค่าไหนเกิด opportunity lost หรือไม่

ตารางที่ 4-29 การสร้าง message

101	<code>if(1 in ans):#if all value is lost (ans[i]==1)do inside</code>
-----	--

บรรทัดที่ 101 เช็คเงื่อนไขว่า i มีค่าใน ans2 หรือไม่

จากนั้นเมื่อทำการเปรียบเทียบเรียบร้อยแล้ว ต่อไปจะเป็นการทำสรุปออกมาเป็น message โดยมีการรวบรวม lostplant ,lost สร้าง string message ดังตาราง 4-30

ตารางที่ 4-30 loop สำหรับการสร้าง message

102	<code>for index in range(len(ans)):#check for all index in range of 0 to number of item in ans[]</code>
103	<code>if (ans[index] == 1):#if this value is lost (ans[i]==1)do inside</code>
104	<code>lostplant.append(plant[index])#add plant name to lostplant</code>
105	<code>lost.append(value[index])#add value of lost to lost[]</code>
106	<code>else :</code>
107	<code>lostplant.append(0)</code>
108	<code>lost.append(0)</code>
109	<code>firstpart ='For last 12 hour ' #initial first past of message</code>
110	<code>j=1</code>
111	<code>#message generator phase</code>
112	<code>for i in range(len(lostplant)):# loop that start from 0 to the end at lostplant item -1</code>
113	<code>if(lostplant[i]!=0):#if this plant lost do inside (0=normal)</code>
114	<code>temp=str('\r'+str(j)+' '+lostplant[i]+' Opt loss = '+str('{:,.}').format(lost[i]))+' THB/Year **'(lost = '+str('{:,.}').format(round(hour[i],1)))+ ' Hour)**'</code>
115	<code>sentent=sentent+temp #store temp to sentent use +</code>
116	<code>j=j+1</code>
117	<code>finalpart='\n\n **Please take action 🤖.**' #initial last past of message</code>
118	<code>message = firstpart+sentent+finalpart#store each part to message use + (firstpart+sentent+finalpart)</code>

บรรทัดที่ 102	ลูป โดยที่ index โดยมี range เป็นขนาดของ ans
บรรทัดที่ 103	เช็คเงื่อนไขว่า i มีค่าเท่ากับ 1 หรือไม่
บรรทัดที่ 104	ทำการappend ค่า plant ที่ index เข้า list lostplant
บรรทัดที่ 105	ทำการappend ค่า value ที่ index เข้า list lost
บรรทัดที่ 107	ทำการappend ค่า 0 ที่ index เข้า list lostplant
บรรทัดที่ 108	ทำการappend ค่า 0 ที่ index เข้า list lost
บรรทัดที่ 109	ประกาศ string firstpart
บรรทัดที่ 110	กำหนดค่า j ด้วย 1
บรรทัดที่ 112	ลูป โดยที่ i โดยมี range เป็นขนาดของ lostplant
บรรทัดที่ 113	เช็คเงื่อนไขว่าlostplant ที่ i มีค่าไม่เท่ากับ 0 หรือไม่
บรรทัดที่ 115	ทำการ joint string temp กับ sentent
บรรทัดที่ 116	ทำการบวกค่าj ด้วย 1
บรรทัดที่ 118	ทำการ joint string firstpart กับ sentent และ final part

เมื่อสร้าง message เสร็จเรียบร้อยในระหว่างจะมีการส่ง report ไปเก็บใน Database ผ่าน module โดยหากระหว่างนี้ไม่มี opportunity loss จะแทนที่ message ด้วยข้อความว่า 'For last 12 hour,there is no Opportunity loss \n\n **Good Job and have a Great Day!!! 😊**' และ เก็บ report ไปเก็บใน Database ผ่าน module ด้วย ดังตาราง 4-31

ตารางที่ 4-31 ทำการเพิ่มค่าเข้าไปใน DB

19	1	DB.inserts(date.today(),list(ans))#use method from object DB that defi
20	1	nd from module to insert list of ans to report
21	1	else:
22	1	message = 'For last 12 hour,there is no Opportunity loss \n\n **Good
		Job and have a Great Day!!! 😊**'
	1	DB.inserts(date.today(),list(ans))#use method from object DB that defi
		nd from module to insert list of ans(that all 0) to report

บรรทัดที่ 119 ทำการเรียก ฟังก์ชัน inserts โดยการส่ง parameter date.today() , ans

บรรทัดที่ 122 ทำการเรียก ฟังก์ชัน inserts โดยการส่ง parameter date.today() , ans

โดยในส่วนนี้จะมีการสร้าง json object ที่คล้ายกับ AI_12hr_feature_drift_webhook.py ตารางที่ 4-18 แต่จะมีการเพิ่ม action set หรือ ลิงค์ไปยังเว็บ dashboard โดยมีลักษณะดังตาราง 4-32

ตารางที่ 4-32 ประกาศโครงสร้างของ payload card

123	#payload message structure json
	⋮
134	"body": [
135	{
136	"type": "TextBlock",
137	"text": message,
138	"wrap": True
	⋮
141	"type": "ActionSet",
142	"actions": [
143	{
144	"type": "Action.OpenUrl",
145	"title": "PIVIEW",
146	"url": "https://www.secret.scg.com/#/Displays"
149	"\$data": "open web",
150	"separator": True

สุดท้ายนี้ในตาราง 4-33 ทำการส่ง json object ที่ได้จากขั้นตอนที่ผ่านมาผ่าน request ไปยัง income webhook service เช่นเดียวกับ ตารางที่ 4-19

ตารางที่ 4-33 ทำการส่ง messagecard ไปยัง income service

160	card = requests.post('https://www.secret.scg.com/#/webhookb2',json=msg)
-----	---

บรรทัดที่ 160 ทำการส่ง messagecard ไปยัง income service

Develop Weekly Report for Virtual assistance

พัฒนา Virtual assistance ของ AI Supervisory ให้สามารถส่ง notification report รายอาทิตย์ได้

AI_weekly_gap_benefit_webhook.py:

เริ่มต้นโดยการนำเข้าไฟล์ library ที่จำเป็นก่อนโดย มี pandas request dbreader ซึ่งเป็น module นอกในการติดต่อ Database ดังตาราง 4-34

ตารางที่ 4-34 ส่วน library ที่ใช้ใน AI_weekly_feature_drift_webhook

1	#import library
2	import pandas as pd
3	import json
4	import requests
5	import dbreader

บรรทัดที่ 2 นำเข้าไฟล์ pandas

บรรทัดที่ 3 นำเข้าไฟล์ json

บรรทัดที่ 4 นำเข้าไฟล์ requests

บรรทัดที่ 5 นำเข้าไฟล์ dbreaderdbreader

ตาราง 4-35 หลังจากนำเข้าไฟล์ libery เรียบร้อยแล้ว ก่อนการอ่านค่าจะต้องทำการประกาศ list ของแต่ละค่าก่อน นำไปแยก

ตารางที่ 4-35 ประกาศค่าเริ่มต้นสำหรับการอ่านค่า

20	for i in range(len(table)):
21	time.append(list(table[i])[0])
22	for i in range(len(table3)):
23	name.append(list(table3[i])[0])
24	for i in range(len(table2)):
25	plant.append(list(table2[i])[0])
26	size=list(csv2['size'])

บรรทัดที่ 20 ลูป โดยที่ i โดยมี range เป็นขนาดของ table

บรรทัดที่ 21 ทำการ append table ตัวที่ i คอลัมน์ที่ 0 ไปที่ time

บรรทัดที่ 22 ลูป โดยที่ i โดยมี range เป็นขนาดของ table3

บรรทัดที่ 23 ทำการ append table3 ตัวที่ i คอลัมน์ที่ 0 ไปที่ name

บรรทัดที่ 24 ลูป โดยที่ i โดยมี range เป็นขนาดของ table2

บรรทัดที่ 25 ทำการ append table2 ตัวที่ i คอลัมน์ที่ 0 ไปที่ plant

บรรทัดที่ 26 ทำการแปลง size จาก Dataframe เป็น list

ตาราง 4-36 จากนั้นทำการแยกค่า xy และ กำหนด message ช่วงแรก

ตารางที่ 4-36 loopสำหรับการแยก X,Y และ นับยอดทั้งหมด

39	<code>while val in range(len(time)):#for loop that val start from 0 to size of time -1</code>
40	<code>if(val < (s[i]-Y[i])):#if val is less than(s at index i - Y at index i)</code>
41	<code>outrangeclass.append('X')#list of out range class add X at back</code>
42	<code>val=val+1#add 1 to val</code>
43	<code>if(val >= (s[i]-Y[i])):</code>
44	<code>for k in range(Y[i]):#for k start from 0 to (y at index i)-1</code>
45	<code>outrangeclass.append('Y')#list of out range class add Y at back</code>
46	<code>val=val+1#add 1 to val</code>
47	<code>i=i+1#add 1 to i for s and y to move</code>
48	<code>if(not all(v == 0 for v in time)):#check if all item in time is not 0 then do inside</code>
49	<code>firstpart ='For this week \n\n'#set first part of string</code>

บรรทัดที่ 39 ลูป โดยที่ i โดยมี range เป็นขนาดของ time

บรรทัดที่ 40 เช็คเงื่อนไขว่า i มีค่าน้อยกว่า s ตัวที่ i -Y ตัวที่ i หรือไม่

บรรทัดที่ 41 ทำการappend ค่า 'X' ที่ index เข้า list outrangeclass

บรรทัดที่ 42 ทำการบวกค่า val ด้วย 1

บรรทัดที่ 43 เช็คเงื่อนไขว่า i มีค่ามากกว่าหรือเท่ากับ s ตัวที่ i -Y ตัวที่ i หรือไม่

บรรทัดที่ 44 ลูป โดยที่ k โดยมี range เป็น Y ตัวที่ i

บรรทัดที่ 45 ทำการappend ค่า 'Y' ที่ index เข้า list outrangeclass

บรรทัดที่ 46 ทำการบวกค่า val ด้วย 1

บรรทัดที่ 47 ทำการบวกค่า i ด้วย 1

บรรทัดที่ 48 เช็คเงื่อนไขว่า ทุกตัวใน time เป็น 0 ทั้งหมด

ในขั้นนี้ เป็นขั้นตอนในการสร้าง message โดยการอ่านค่า report ที่ได้มา โดยทำการแบ่ง xy และ แทรก ชื่อ plant ที่มี opportunity loss เข้าไประหว่าง plant เมื่อทำครบแล้วจึงนำ message แต่ละ part มาเชื่อมกันด้วย “+” หรือกรณีที่ไม่มี opportunity loss จะแทนที่ข้อความด้วย For this week, there is no value out of range ดังตาราง 4-37

ตารางที่ 4-37 loop สำหรับการสร้าง message

55	<code>if(plantindex not in tempplant):</code>
56	<code>sentent =sentent+'- '+plant[plantindex-1]+' \r'</code>
57	<code>tempplant.append(plantindex)</code>
58	<code>if(time[j]!=0):#check if time at index j is not equal 0</code>
59	<code>temp="#defind temp for each loop</code>
60	<code>if(plantindex not in xtemp and outrange[j]!='X'):</code>
61	<code>sentent=sentent+"\n\n X: \r'</code>
62	<code>xtemp.append(plantindex)</code>
63	<code>elif(plantindex not in ytemp and outrange[j]!='Y'):</code>
64	<code>sentent=sentent+"\n\n Y: \r"</code>
65	<code>ytemp.append(plantindex)</code>
70	<code>temp=str("\r '+str(k+1)+' '+name[j])+ ' has out of range for '+str('{:,}'.format(time[j]))+wordtime+' \n\n '</code>
71	<code>sentent=sentent+temp#store string of temp in to sentent</code>
72	<code>k=k+1</code>
73	<code>sentent=sentent+"\n\n"</code>
74	<code>finalpart='\n\n **Please consider 😊. **'</code>
75	<code>message = firstpart+sentent+finalpart#joint each part to message (first part + sentent + final part)</code>
76	<code>else:</code>
77	<code>message = 'For this week,there is no value out of range\n\n **Good Job and have a Great Day!!! 😊 **'</code>

- บรรทัดที่ 55 ทำการเช็คเงื่อนไข index ของ plant อยู่ในค่า temp plant หรือไม่
- บรรทัดที่ 56 ทำการ joint string sentent ด้วย plant name ที่ indexplant-1
- บรรทัดที่ 57 ทำการ append index plant ไปยัง temp plant
- บรรทัดที่ 58 เช็คเงื่อนไขว่า time ที่ j มีค่าไม่เท่ากับ 0 หรือไม่
- บรรทัดที่ 59 ประกาศ string temp
- บรรทัดที่ 60 ทำการเช็คเงื่อนไขว่า plant index ไม่อยู่ใน xtemp และ outrangeclasse ที่ j เท่ากับ 'x' หรือไม่
- บรรทัดที่ 61 ทำการ joint string sentent ด้วย '\n\n X: \r'
- บรรทัดที่ 62 ทำการ append index plant ไปยัง xtemp
- บรรทัดที่ 63 ทำการเช็คเงื่อนไขว่า plant index ไม่อยู่ใน xtemp และ outrangeclasse ที่ j เท่ากับ 'y' หรือไม่
- บรรทัดที่ 64 ทำการ joint string sentent ด้วย '\n\n Y: \r'
- บรรทัดที่ 65 ทำการ append index plant ไปยัง ytemp
- บรรทัดที่ 71 ทำการ joint string temp กับ sentent
- บรรทัดที่ 72 ทำการบวกค่า k ด้วย 1
- บรรทัดที่ 75 ทำการ joint string first part กับ sentent และ final part

โดยในไฟล์นี้จะมีการสร้าง json object ที่คล้ายกับ AI_12hr_feature_drift_webhook.py ตาราง 4-18 แต่จะมีการเพิ่ม action set หรือ ลิงค์ไปยังเว็บ dashboard โดยมีลักษณะดังตารางที่ 4-38

ตารางที่ 4-38 ประกาศโครงสร้างของ payload card โดยใส่ message เข้าไป

98	"type": "TextBlock",
99	"text": message,
100	"wrap": True

สุดท้ายนี้ในการส่ง json object ที่ได้จากขั้นตอนที่ผ่านมาผ่าน request ไปยัง income webhook service เช่นเดียวกับ ตาราง 4-19

Weekly Monitor AI feature drift report

AI_weekly_feature_drift_webhook.py:

เริ่มต้นโดยการนำเข้าไฟล์ library ที่จำเป็นก่อนโดย มี pandas request dbreader ซึ่งเป็น module นอกในการติดต่อ Database ดังตาราง 4-39

ตารางที่ 4-39 ส่วน library ที่ใช้ใน AI_weekly_gap_benefit_webhook

1	#all library that import
2	import json
3	import requests
4	#import module
5	import dbreader

บรรทัดที่ 2 นำเข้าไฟล์ json

บรรทัดที่ 3 นำเข้าไฟล์ requests

บรรทัดที่ 5 นำเข้าไฟล์ dbreader

จากนั้นดังตาราง 4-40 เราจะทำการ ประกาศค่าเริ่มต้นสำหรับการ เช็คเงื่อนไข และ การสร้าง message โดยเราจะทำการ ประกาศ ชื่อ plant ,time ,lostplant ,sentent

ตารางที่ 4-40 ประกาศค่าเริ่มต้นสำหรับการ เช็คเงื่อนไข และ การสร้าง message

14	plant=['T4 Opt Benefit', T2 Opt Benefitd','T5 Opt Benefit','T6 Opt Benefit' , 'T8 Opt Benefit','T3 Opt Benefit','T7 Opt Benefit'
15	time=[]#for store time
16	lostplant=[]#for plant that lost
17	sentent="#"#for store string use in json

บรรทัดที่ 14 ประกาศ list plant โดยระบุชื่อ plant เป็นสมาชิก list

บรรทัดที่ 15 ประกาศ list time

บรรทัดที่ 16 ประกาศ list lostplant

บรรทัดที่ 17 ประกาศ string sentent

ดังตาราง 4-41 ทำการเช็คเงื่อนไขว่า มี 1 ในansหรือไม่ เพื่อเช็คล่วงหน้า

ตารางที่ 4-41 การสร้าง message

18	เช็คเงื่อนไขว่า มี 1 ในansหรือไม่
19	<code>if(not all(v == 0 for v in ans)):</code>
21	firstpart ='For this week '#set first part of string

บรรทัดที่ 18 เช็คเงื่อนไขว่า ทุกตัวใน time เป็น 0 ทั้งหมด

ตาราง 4-42 เป็นการสร้างข้อความ report ในการประกาศ json object หากข้อความไม่มี feature drift จะแทนที่ message ด้วย For this week,there is no Opportunity loss

ตารางที่ 4-42 loopสำหรับการสร้าง message

22	<code>for i in range(len(plant)):#for i that start from 0 to size of plant - 1</code>
23	<code>if(ans[i]!=0):#if list of ans at index i is not equal 0</code>
24	<code>if(ans[i]>1):#if list of ans at index i is more than 1</code>
25	<code>wordtime=' times'#definid 'times' to wordtime</code>
26	<code>else:</code>
27	<code>wordtime=' time'#definid 'time' to wordtime</code>
28	<code>temp=str("\n\n"+str(j))+'. '+plant[i]+' is Opt loss for ' +str('{:,.}').format(ans[i]))+wordtime</code>
29	<code>sentent=sentent+temp#store string of temp in to sentent</code>
30	<code>j=j+1#add 1 to j</code>
31	<code>finalpart="\n\n **Please consider 🤖.**"#define final part string</code>
32	<code>message = firstpart+sentent+finalpart</code>
33	<code>else:</code>
34	<code>message = 'For this week,there is no Opportunity loss\n\n **Good Job and have a Great Day!!! 😊'</code>

บรรทัดที่ 22 ลูป โดยที่ i โดยมี range เป็นขนาดของ plant

บรรทัดที่ 23 เช็คเงื่อนไขว่า ans ที่ i มีค่าไม่เท่ากับ 0 หรือไม่

บรรทัดที่ 24 เช็คเงื่อนไขว่า ans ที่ i มีค่ามากกว่า 1 หรือไม่

บรรทัดที่ 29 ทำการ joint string sentent ด้วย temp

บรรทัดที่ 30 ทำการบวกค่า j ด้วย 1

บรรทัดที่ 32 ทำการ joint string firstpart กับ sentent และ final part

โดยในส่วนนี้จะมีการสร้าง json object ที่คล้ายกับ AI_12hr_feature_drift_webhook.py ตาราง 4-18 แต่จะมีการเพิ่ม action set หรือ ลิงค์ไปยังเว็บ dashboard โดยมีลักษณะดังตาราง 4-43

ตารางที่ 4-43 ประกาศโครงสร้างของ payload card

47	"body": [
48	{
49	"type": "TextBlock",
50	"text": message,
51	"wrap": True
58	"type": "ActionSet",
59	"actions": [
60	{
61	"type": "Action.OpenUrl",
62	"title": "PIVIEW",
63	"url": "https://www.secret.scg.com/#/Displays"

สุดท้ายนี้ในตาราง 4-44 ทำการส่ง json object ที่ได้จากขั้นตอนที่ผ่านมาผ่าน request ไปยัง income webhook serviceเช่นเดียวกับ ตาราง 4-19

ตารางที่ 4-44 ทำการส่ง messagecard ไปยัง income service

77	card = requests.post('https://www.secret.scg.com/#/webhookb2', json=msg)
----	--

บรรทัดที่ 75 ทำการ dump เป็น json

บรรทัดที่ 76 ทำการ load json

บรรทัดที่ 77 ทำการส่งmessagecardไปยัง income service

4.2.2. ส่วน ANOMALY DETECTION

Develop Anomaly Detection

เป็นการพัฒนา Virtual assistance ของ Anomaly Detection ใช้เทคนิค machine learning ที่เรียกว่า K-means clustering เพื่อให้สามารถส่ง notification report ทุกวันได้ โดยจะแสดงในระหว่างวันนั้นๆ หากเกิด anomaly ณ วันนั้นๆ จะยกเลิกการแจ้งเตือนของวันนั้น

ตัวอย่างการพัฒนา model k-mean clustering :

K_mean clustering UNIT3.ipynb:

โดยเริ่มจาก นำเข้า pandas numpy และ matplotlib สำหรับการ สร้าง model machine learning ดังตารางที่ 45

ตารางที่ 4-45 ส่วนการนำเข้าไฟล์

1	import pandas as pd
2	import numpy as np
3	import matplotlib.pyplot as plt
4	%matplotlib inline
5	df = pd.read_csv('UNIT3 data.csv')
6	df.head(5)

บรรทัดที่ 1 นำเข้าไฟล์ pandas

บรรทัดที่ 2 นำเข้าไฟล์ numpy

บรรทัดที่ 3 นำเข้าไฟล์ matplotlib

บรรทัดที่ 4 นำเข้าไฟล์ csv UNIT3 data.csv

	Time	U3T1.pv	U3T2.pv	U3T3.pv	U3T4.pv
0	11-Oct-12 00:00:00	80.038029	64.958193	24.876602	9.796560
1	11-Oct-12 01:00:00	80.013192	65.281715	24.803479	9.958032
2	11-Oct-12 02:00:00	80.009916	64.618870	26.454210	9.795728
3	11-Oct-12 03:00:00	80.010475	64.506459	24.978429	8.858511
4	11-Oct-12 04:00:00	80.031388	64.366574	25.427279	9.615456

ภาพที่ 4-1 ผลลัพธ์จากคำสั่ง df.head(5)

ในตาราง 4-46 จากนั้นทำการลบ column time ออกเนื่องจากไม่ได้ใช้

ตารางที่ 4-46 ส่วนการจัดการ column

7	df.drop('Time',axis=1,inplace=True)
8	df.dropna(inplace=True)
9	df.describe()

บรรทัดที่ 7 ทำการdrop columntime

บรรทัดที่ 8 ทำการdrop non value

บรรทัดที่ 9 ทำการดูผลลัพธ์

	U3T1.pv	U3T2.pv	U3T3.pv	U3T4.pv
count	87238.000000	87238.000000	87238.000000	87238.000000
mean	96.594762	78.540332	27.812709	12.714860
std	22.124412	19.804581	6.562289	3.318792
min	0.002264	-1.295013	-0.000183	-0.003326
25%	90.174017	73.405584	25.766904	11.296883
50%	105.125084	83.663257	29.413498	13.163377
75%	110.201022	90.085323	31.695826	14.805887
max	115.297380	106.465804	42.399177	23.107758

ภาพที่ 4-2 ผลลัพธ์จากคำสั่ง df.describe

ในตาราง 4-47 ทำการแปลง data frame เป็น numpy เนื่องจากการทำ min max scaler จำเป็นต้องใช้ numpy datatype เท่านั้น

ตารางที่ 4-47 ส่วนการแปลง Dataframe to numpy

10	X = df
11	# Change DATAFRAME to NumPy
12	X_numpy = X.to_numpy()
13	X_numpy

บรรทัดที่ 10 กำหนด X ด้วย Datafram df

บรรทัดที่ 12 แปลง X Dataframe เป็น numpy

บรรทัดที่ 13 ทำการดูผลลัพธ์

จากการแปลงในตารางที่ผ่านมา ทำให้ได้ ข้อมูล x แบบ numpy แต่ เนื่องจาก scale ข้อมูลนั้น ห่างกันเกินไป ซึ่งจะส่งผลในขั้นตอนการสร้าง model ต่อไปจะต้องทำการscale ข้อมูลให้อยู่ในขอบเขต 0 ถึง 1 ดังตาราง 4-48

ตารางที่ 4-48 ส่วนการเตรียม min max scaler

14	import joblib
15	from sklearn.preprocessing import MinMaxScaler
16	scaler=MinMaxScaler()
17	scaler.fit(X_numpy)
18	filename = './UNIT3_minmax.sav'
19	joblib.dump(scaler, filename)
20	X=scaler.transform(X_numpy)

บรรทัดที่ 14 นำเข้าไฟล์ joblib

บรรทัดที่ 15 นำเข้าไฟล์ MinMaxScaler

บรรทัดที่ 16 ทำการประกาศ class MinMaxScaler

บรรทัดที่ 17 ทำfit model

บรรทัดที่ 19 ทำ save model

บรรทัดที่ 20 ทำการ transform data

เมื่อได้ข้อมูลที่ผ่านการ scale แล้วต่อไปจะเป็นการนำข้อมูลทำการแบ่งข้อมูล train test โดยต้องทำการแปลงกลับเป็น Dataframe ก่อนจึงจะทำการแบ่งได้ ดังตาราง 4-49

ตารางที่ 4-49 ส่วนการเตรียมข้อมูล

21	#Prepare data as train-test split
22	from sklearn.model_selection import train_test_split
23	X=pd.DataFrame(X)
24	X

บรรทัดที่ 22 นำเข้าไฟล์ train_test_split

บรรทัดที่ 23 กำหนด X ด้วย Dataframe

	0	1	2	3
0	0.694182	0.614817	0.586725	0.424034
1	0.693966	0.617819	0.585001	0.431021
2	0.693938	0.611668	0.623934	0.423998
3	0.693943	0.610625	0.589127	0.383445
4	0.694124	0.609327	0.599713	0.416198
...
87233	0.868907	0.815730	0.749171	0.565424
87234	0.868368	0.818233	0.745560	0.549982
87235	0.869000	0.822478	0.779165	0.562703
87236	0.868750	0.820630	0.783107	0.573889
87237	0.869383	0.818238	0.775963	0.514744

ภาพที่ 4-3 ผลลัพธ์จากคำสั่ง X

87238 rows × 4 columns

ในตาราง 4-50 เมื่อได้ ข้อมูล train test มาแล้ว เราจะนำข้อมูล train มาทำการ train model Kmean ก่อน โดยกำหนด cluster 2 ตัว เนื่องจากเราจะแบ่งเป็น normal กับ abnormal

ตารางที่ 4-50 ส่วนการทำ kmean clustering

25	from sklearn.cluster import KMeans
26	kmeans = KMeans(n_clusters=2, n_init=20, max_iter=600, random_state=40).fit(X_train)
27	# See clustreing results
28	kmeans.labels_

บรรทัดที่ 25 นำเข้าไฟล์ KMeans

บรรทัดที่ 26 ทำการประกาศ Kmeans และ fit x train

บรรทัดที่ 28 ทำการเรียกผลลัพธ์ Clustering มาดู

```
array([0, 0, 0, ..., 0, 0, 0])
```

ภาพที่ 4-4 ผลลัพธ์จากคำสั่ง kmeans.labels

ในช่วงหลังจากนี้จะทำการตรวจสอบความแม่นยำ model ด้วยการตรวจสอบ cluster_centers ในตาราง 4-51

ตารางที่ 4-51 ส่วนตรวจสอบความแม่นยำ model

29	# Observe K-means cluster center
30	kmeans.cluster_centers_

บรรทัดที่ 30 ทำการเรียกผลลัพธ์ Cluster center มาดู

```
array([[0.89371295, 0.79475808, 0.69882956, 0.58805041],
       [0.45292815, 0.37037127, 0.36107493, 0.29138069]])
```

ภาพที่ 4-5 ผลลัพธ์จากคำสั่ง `kmeans.cluster_centers`

ในตาราง 4-52 เมื่อทำการตรวจสอบความแม่นยำ model เสร็จแล้ว จะเป็นการบันทึก model ด้วย `joblib.dump` เพื่อนำ `.sav` มาใช้ในไฟล์ถัดไป

ตารางที่ 4-52 ส่วนการ save model

31	# save the model to disk
32	import joblib
33	filename = './UNIT3_Kmeans.sav'
34	joblib.dump(kmeans, filename.e)

บรรทัดที่ 32 นำเข้าไฟล์ `joblib`

บรรทัดที่ 34 ทำ save model

Develop Anomaly Detection ของ Unit 1 ,2 ,3 ,4

Unitx_AD.py:

เริ่มด้วยการนำเข้าไฟล์ library ที่จำเป็นต้องใช้ เช่น `numpy pandas filesave joblib request`
ดังตาราง 4-53

ตารางที่ 4-53 ส่วน library ที่ใช้ใน Unitx_AD

8	<code>import numpy as np</code>
9	<code>import pandas as pd</code>
11	<code>import requests</code>
13	<code>import joblib</code>
15	<code>import requests</code>
17	<code>import filesave</code>

บรรทัดที่ 8 นำเข้าไฟล์ numpy
 บรรทัดที่ 9 นำเข้าไฟล์ pandas
 บรรทัดที่ 11 นำเข้าไฟล์ requests
 บรรทัดที่ 13 นำเข้าไฟล์ joblib
 บรรทัดที่ 15 นำเข้าไฟล์ requests
 บรรทัดที่ 17 นำเข้าไฟล์ filesave

จากนั้นในตาราง 4-54 จะทำการกำหนดค่าเริ่มต้นให้กับ piwebapi โดยจะคล้ายกับ ไฟล์ AI_12hr_feature_drift_webhook.py ตาราง 4-18 แต่ต่างตรงที่ interval เปลี่ยน เป็น 1 นาที

ตารางที่ 4-54 กำหนดค่าสำหรับ piwebapi

28	interval='1m'#interval 1 minutes3
----	-----------------------------------

บรรทัดที่ 28 ประกาศ interval เป็น 1 นาที

โดยในตาราง 4-55 จะประกาศ tag file เพื่อใช้ในการเลือก ข้อมูล tag ที่จะดึงข้อมูลมา

ตารางที่ 4-55 ประกาศค่าเริ่มต้นสำหรับการอ่านค่า

33	tag=['UxT1.PV','UxT2.PV','UxT3.PV']
----	-------------------------------------

บรรทัดที่ 33 ประกาศ list ของ tag ที่จะใช้

จากตาราง 4-56 นี้จะเป็นการเรียกข้อมูล ตามที่กำหนดผ่าน get average value โดยลบ column timestamp ออกเนื่องจากไม่ใช้

ตารางที่ 4-56 ทำการ request piwebapi และ drop columnที่ไม่จำเป็น

34	#get data phase
35	df = webapi.getAverageValue(tag,start,end,cal_basis,cal_type,interval)
36	df=df.drop(['timestamp'], axis=1)

บรรทัดที่ 35 ทำการเรียก ฟังก์ชัน get average value

บรรทัดที่ 36 ทำการ drop column ชื่อ time และแปลงค่าเป็นเลข

ตาราง 4-57 เป็นการ เตรียมข้อมูล ด้วยการ นำเข้าไฟล์ min max scaler และทำการ scale ข้อมูลระหว่าง 0,1

ตารางที่ 4-57 ทำการ load model min-max scaler และทำการ prepare data

37	#prepare data phase
38	scaler=joblib.load(path+'Unitx_minmax.sav')
39	X_numpy=scaler.transform(df)# transform by use scaler between 1,0

บรรทัดที่ 38 ทำการเรียก ฟังก์ชัน load ในการดึงข้อมูล model min max scaler

บรรทัดที่ 39 ทำการเรียก ฟังก์ชัน transform ในการ prepare data

เมื่อทำการ scale ข้อมูล เสร็จ ตาราง 4-58 จะเป็นการนำเข้า model kmean ของ unit นั้นๆ แล้ว ทำการ predict ผลลัพธ์ออกมาไว้ใน column class

ตารางที่ 4-58 ทำการ load model kmean และทำการ predict data

40	#predict phase
41	model = joblib.load(path+'Unitx_Kmeans.sav')
42	results = model.predict(X_numpy)#predite ans from x that give
43	result_list = results.tolist()#convert to list
44	df['class'] = result_list#put into Dataframe

บรรทัดที่ 41 ทำการเรียก ฟังก์ชัน load ในการดึงข้อมูล model min max scaler

บรรทัดที่ 42 ทำการเรียก ฟังก์ชัน predict ในการ predict data

บรรทัดที่ 43 ทำการแปลง results เป็น list

บรรทัดที่ 44 ทำการเพิ่ม columnclass ด้วย list result

ตาราง 4-59 ขั้นตอนในการตรวจสอบเกณฑ์ผ่านการclass โดยเริ่มจากการประกาศ class csv

ตารางที่ 4-59 ทำการเอาข้อมูลมาเช็คเงื่อนไข

45	#check condition phase
46	abnormal=False#define abnormal to false
47	filename='Unitx'#define filename
48	csv=filesave.csv(filename)
49	csv.work()#check condition

บรรทัดที่ 46 กำหนดค่า abnormal ด้วย false

บรรทัดที่ 47 กำหนดค่า filename ด้วย 'Unit3'

บรรทัดที่ 48 ทำการเรียก constructor csv

บรรทัดที่ 49 ทำการเรียก ฟังก์ชัน work

ตาราง 4-60 เช็คว่าเป็นข้อมูลabnormalหรือไม่หากเข้าเงื่อนไขจะทำการสร้างmessageและ save status True ลง Database

ตารางที่ 4-60 หากเข้าเงื่อนไขจะทำการสร้าง message และ save status ลง DB

50	if(df['class'].values[0]==1):#check if y that get from model is abnormal (0)
51	print('abnormal')
52	csv.save(True)#save status and dates to csv
53	abnormal=True#set true
54	topic='**unitx process abnormal !!**'#defind topic text
55	message=""#define string
56	i=1#define integral i
57	for inputs in df.iloc[0,:-1].values[0]:
58	message=message+' \n\n'+str(i)+'.'+tag[i]+'=' +str(inputs.round(1))+' T/H'
59	i=i+1#add 1 to i
60	#defind json structure

บรรทัดที่ 50 เช็คเงื่อนไขว่า df column class มีค่าเท่ากับ 1 หรือไม่

บรรทัดที่ 51 ทำการเรียก ฟังก์ชัน save โดย parameter True

บรรทัดที่ 52 กำหนดค่า abnormal ด้วย True

บรรทัดที่ 55 ประกาศ string message

บรรทัดที่ 56 กำหนดค่า i ด้วย 1

บรรทัดที่ 57 ลูป โดยที่ inputs โดยมี ขอบเขตเป็น สมาชิกของ data frame row size-1

บรรทัดที่ 59 ทำการบวกค่า i ด้วย 1

โดยในส่วนนี้จะมีการสร้าง json object ที่คล้ายกับ AI_12hr_feature_drift_webhook.py ตาราง 4-17 แต่จะมีการแยก text อีกตัวเป็น topic โดยมีลักษณะดังตาราง 4-61

ตารางที่ 4-61 โครงสร้างของpayload card

61	payload_card={
62	"type": "message",
66	"content": {
67	"\$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
68	"type": "AdaptiveCard",
71	"body": [
72	{
73	"type": "TextBlock",
74	"text": topic,
75	"wrap": True,
76	"color": "Attention"
79	"type": "TextBlock",
80	"text": message,
81	"wrap": True

ตาราง 4-62 หากข้อมูลที่ได้มาไม่มี abnormal Data จะทำการ ส่ง state false ไปยัง Database

ตารางที่ 4-62 ส่วนไม่เข้าเงื่อนไข

88	else:
89	csv.save(False)#save status and date to csv
90	if(abnormal==True):#if abnormal is true do inside

จะทำการ save status ลง DB

บรรทัดที่ 89 ทำการเรียก ฟังก์ชัน save โดย parameter False

บรรทัดที่ 90 เช็คเงื่อนไขว่า abnormal มีค่าเท่ากับ True หรือไม่

หาก state abnormal เป็น true สุดท้ายนี้ในตาราง 4-63 ทำการส่ง json object ที่ได้จากขั้นตอนที่ผ่านมาผ่าน request ไปยัง income webhook serviceเช่นเดียวกับ ตาราง 4-18

ตารางที่ 4-63 ส่วนที่ทำการส่ง message card ไปยัง income service

93	card = requests.post('https://www.secret.scg.com/#/webhookb2',json=msg)
----	---

บรรทัดที่ 93 ทำการส่งmessage cardไปยัง income service

4.2.3. ส่วนที่เกี่ยวข้องกับระบบฐานข้อมูล

Develop Database พัฒนาระบบการเก็บข้อมูลแบบ database

dbreader.py:

เริ่มด้วยการนำเข้าไฟล์ library ที่จำเป็นต้องใช้ เช่น psycopg2 pandas ดังตาราง 4-64

ตารางที่ 4-64 ส่วน library ที่ใช้ใน dbreader

1	#import use libery
2	import psycopg2
3	import pandas as pd
4	path='/src/'#set path prefix

บรรทัดที่ 1 นำเข้าไฟล์ psycopg2

บรรทัดที่ 2 นำเข้าไฟล์ pandas

บรรทัดที่ 3 กำหนด path prefix ของไฟล์ที่จำเป็นต้องอ่านในการทำงาน

ที่ตาราง 4-65 ทำการประกาศ class DB และ กำหนด constructor เพื่อ config conn สำหรับการติดต่อ database

ตารางที่ 4-65 class DB

5	class DB:#define class name DB
	constructor เพื่อ ตั้งค่า server psycopg2
6	def __init__(self):#define class constructor
7	#define this connection to method psycopg1.connect()
8	self.conn = psycopg2.connect(
12	password="xx#2022",
13	port="xxxx")

บรรทัดที่ 6 ประกาศ class constructer

บรรทัดที่ 7 ประกาศ host config โดยฟังก์ชัน connect

ตาราง 4-66 ทำการประกาศฟังก์ชัน inserts โดยกำหนด คำสั่ง raw sql ในการ insert report ไปยัง report พร้อม คำนวณค่า print respond โดยเก็บไว้ใช้ในการสรุป opportunity loss รายอาทิตย์

ตารางที่ 4-66 class ชื่อ database

14	<code>def inserts(self,dates,input):</code>
15	<code>sql=""" INSERT INTO report("T4","</code> <code>T2","T5","T6","T8","T3","Date","T7")</code>
16	<code>VALUES (%s,%s,%s,%s,%s,%s,%s,%s);"""#raw sql</code>
17	<code>cur=self.conn.cursor()#define cursor of this connection</code>
18	<code>#print(dates)</code>
19	<code>#print(input)</code>
20	<code>#execute command by use rawsql with vars of tuple</code>
21	<code>cur.execute(sql,(input[0],input[1],input[2],input[3],</code> <code>input[4],input[5],dates,input[6]))</code>
22	<code>#commit execute of command</code>
23	<code>self.conn.commit()</code>
24	<code>#close cursor</code>
25	<code>cur.close()</code>
26	<code>return print('DB repond')</code>

บรรทัดที่ 14 ประกาศ ฟังก์ชัน insert รับ argument dates ,input

บรรทัดที่ 15 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 17 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก class conn

บรรทัดที่ 21 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql ข้อมูลที่นำไปกำหนด

บรรทัดที่ 23 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 25 ทำการเรียกฟังก์ชัน close เพื่อสิ้นสุดการทำงานของ cursor

บรรทัดที่ 26 คืนค่ากลับไปด้วยการ print ข้อความ

ตาราง 4-67 ทำการประกาศฟังก์ชัน read โดยกำหนด คำสั่ง raw sql ในการ select report โดย sum ค่าแต่ละตัว พร้อม คืนค่า tuple ในการสรุป opportunity loss รายอาทิตย์

ตารางที่ 4-67 function read

27	<code>def reads(self):#define method name reads to read all data from table name report by date that 1 week before</code>
28	<code>#raw sql</code>
29	<code>sql="" select sum("T4")as T4 ,sum("T2")as T2 ,sum("T5")as T5 ,sum("T6")as T6 ,sum("T8")as T8 ,sum("T3")as T3 ,sum("T7") as T7 FROM report</code>
30	<code>WHERE "Date" BETWEEN date(CURRENT_DATE)-7 AND date(CURRENT_DATE)-0""</code>
35	<code>value=cur.fetchall()</code>
36	<code>return value#return list of tuple name value</code>

ใช้ในการ read ข้อมูลและทำการรวมค่าจาก report ระหว่าง ช่วงเวลาที่กำหนด

บรรทัดที่ 27 ประกาศ ฟังก์ชัน reads

บรรทัดที่ 29 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 31 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก class conn

บรรทัดที่ 33 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql

บรรทัดที่ 34 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 35 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 36 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-68 ทำการประกาศฟังก์ชัน inserts โดยกำหนด คำสั่ง raw sql ในการ insert report/ ไปยัง report/ พร้อม คืนค่า print respond โดยเก็บไว้ใช้ในการสรุป feature drift รายอาทิตย์

ตารางที่ 4-68 function inserts 2

37	<code>def inserts2(self,dates,input,feature_index):#define method name i nsert2 for insert date and input to table report 2</code>
38	<code>sql=""" INSERT INTO report2(featurename_id,timeoccu,plant_id, date)</code>
39	<code>VALUES (%s,%s,%s,%s);"""</code>
41	<code>plant_index=self.findplantindex(feature_index);#define plant in dex by use method findplantindex</code>
44	<code>cur.close()#close cursor</code>

ใช้ในการ insert ข้อมูลเข้า report2

บรรทัดที่ 37 ประกาศ ฟังก์ชัน inserts2

บรรทัดที่ 38 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 41 ทำการเรียกฟังก์ชัน findplantindex เพื่อหา index plant

บรรทัดที่ 44 ทำการเรียกฟังก์ชัน close เพื่อสิ้นสุดการทำงานของ cursor

ตาราง 4-69 ทำการประกาศฟังก์ชัน read โดยกำหนด คำสั่ง raw sql ในการ select report2 โดย sum ค่าแต่ละตัว พร้อม คืนค่า tuple ในการสรุป feature drift รายอาทิตย์

ตารางที่ 4-69 function read 2

45	<code>def reads2(self):#define method name reads2 to read all data from</code>
47	<code>table name report 2 by date that 1 week before</code>
48	<code>sql="""select s1.ans from</code>
49	<code>(</code>
50	<code>select sum(timeoccu) as ans ,featurename_id,plant_id FRO</code>
51	<code>M report2</code>
52	<code>WHERE "date" BETWEEN date(CURRENT_DATE)-</code>
53	<code>7 AND date(CURRENT_DATE)-0</code>
54	<code>GROUP BY featurename_id,plant_id</code>
55	<code>) s1</code>
56	<code>ORDER BY featurename_id</code>
57	<code>"""</code>
58	<code>value=cur.fetchall()#define value that get value by fetch all valu</code>
59	<code>e from commit</code>
60	<code>return value#return list of tuple name value</code>

ใช้ในการ read ข้อมูลและทำการรวมค่าจาก report2 ระหว่าง ช่วงเวลาที่กำหนด

และต้องไม่เป็น 0

บรรทัดที่ 45 ประกาศ ฟังก์ชัน inserts2

บรรทัดที่ 47 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 59 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 60 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-70 ทำการประกาศฟังก์ชัน getallplantname โดยกำหนด คำสั่ง raw sql ในการ อ่าน ชื่อ plant ทั้งหมด

ตารางที่ 4-70 function getallplantname

61	def getallplantname(self):
62	#raw sql
63	sql="" select plant_name FROM plant""
64	cur=self.conn.cursor()#define cursor of this connection
65	# print(input)
	⋮
68	value=cur.fetchall()
69	return value#return list of tuple name value

ใช้ในการ read ข้อมูลplant_name ทั้งหมดจาก plant

บรรทัดที่ 61 ประกาศ ฟังก์ชัน getallplantname

บรรทัดที่ 63 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 64 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก class conn

บรรทัดที่ 68 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 69 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-71 ทำการประกาศฟังก์ชัน readtag โดยกำหนด คำสั่ง raw sql ในการ อ่านชื่อ tag ทั้งหมด

ตารางที่ 4-71 function readtag

70	<code>def readtag(self):</code>
71	<code>#raw sql</code>
72	<code>sql="" select tag_name FROM tag""</code>
73	<code>cur=self.conn.cursor()#define cursor of this connection</code>
74	<code>cur.execute(sql)#execute command by use rawsql</code>
75	<code>self.conn.commit()#commit execute of command</code>
76	<code>lists=[] #defind list for store all tag</code>
77	<code>for i in cur.fetchall():#for each data i that fetch all from com mit</code>
78	<code>lists.append(i[0])#add value of tuple to list at back</code>
79	<code>return lists #reture list of value</code>

ใช้ในการ read ข้อมูล tag_name ทั้งหมดจาก tag

บรรทัดที่ 70 ประกาศ ฟังก์ชัน getallplantname

บรรทัดที่ 72 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 73 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก class conn

บรรทัดที่ 74 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql ข้อมูลที่ใช้

บรรทัดที่ 75 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 76 ลูป ค่า i โดน range ตั้งแต่ 1 ถึง ขนาด ของ size - 1

บรรทัดที่ 77 เช็คเงื่อนไขว่า input มากกว่า หรือ เท่ากับ size ที่ i-1 และ input น้อยกว่า หรือ เท่ากับ size ที่ i หรือไม่

บรรทัดที่ 78 ทำการ append i ไปยัง lists

บรรทัดที่ 79 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-72 ทำการประกาศฟังก์ชัน getmax โดยกำหนด คำสั่ง raw sql ในการ อ่านชื่อ ค่า max จาก index ที่ได้รับ

ตารางที่ 4-72 function getmax

80	<code>def getmax(self,index):</code>
81	<code>#raw sql</code>
82	<code>sql="" select feature_max FROM feature</code>
83	<code>WHERE "feature_id" = (%s)""</code>
	<code>...</code>
87	<code>value=cur.fetchall()</code>
88	<code>return value#return list of tuple name value</code>

ใช้ในการ read ข้อมูล feature_max จาก feature โดยดูจาก feature_id

บรรทัดที่ 80 ประกาศ ฟังก์ชัน getmax

บรรทัดที่ 82 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 87 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 88 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-73 ทำการประกาศฟังก์ชัน getmin โดยกำหนด คำสั่ง raw sql ในการ อ่านชื่อ ค่า min จาก index ที่ได้รับ

ตารางที่ 4-73 function getmin

89	<code>def getmin(self,index):</code>
90	<code>sql="" select feature_min FROM feature</code>
91	<code>WHERE "feature_id" = (%s)"""</code>
92	<code>cur=self.conn.cursor()#define cursor of this connection</code>
93	<code>cur.execute(sql,(index,))</code>
94	<code>self.conn.commit()#commit execute of command</code>
95	<code>value=cur.fetchall()</code>
96	<code>return value#return list of tuple name value</code>

ใช้ในการ read ข้อมูล feature_min จาก feature โดยดูจาก feature_id

บรรทัดที่ 89 ประกาศ ฟังก์ชัน getmin

บรรทัดที่ 90 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 92 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก class conn

บรรทัดที่ 93 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql ข้อมูลที่ใช้

บรรทัดที่ 94 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 95 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 96 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-74 ทำการประกาศฟังก์ชัน getunit โดยกำหนด คำสั่ง raw sql ในการ อ่านชื่อ ค่า unit จาก index ที่ได้รับ

ตารางที่ 4-74 function getunit

97	<code>def getunit(self,index):</code>
98	<code>#raw sql</code>
99	<code>sql="" select unit FROM feature</code>
100	<code>WHERE "feature_id" = (%s)""</code>
101	<code>cur=self.conn.cursor()#define cursor of this connection</code>
102	<code>cur.execute(sql,(index,))</code>
103	<code>self.conn.commit()#commit execute of command</code>
104	<code>value=cur.fetchall()</code>
105	<code>return value#return list of tuple name value</code>

ใช้ในการ read ข้อมูล unit จาก feature โดยดูจาก feature_id

บรรทัดที่ 97 ประกาศ ฟังก์ชัน getunit

บรรทัดที่ 99 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 101 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก class conn

บรรทัดที่ 102 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql ข้อมูลที่ใช้

บรรทัดที่ 103 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 104 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 105 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-75 ทำการประกาศฟังก์ชัน readname โดยกำหนด คำสั่ง raw sql ในการ อ่านชื่อ name ทั้งหมด name คือ feature name ไม่มีการตัดบางตัวออก

ตารางที่ 4-75 function readname

106	<code>def readname(self):</code>
107	<code>#raw sql</code>
108	<code>sql="" select feature_name FROM name ""</code>
	<code>...</code>
112	<code>lists=[]#define list for store all name</code>
113	<code>for i in cur.fetchall():#for each data i that fetch all from com mit</code>
114	<code>lists.append(i[0])#add value of tuple to list at back</code>
115	<code>return lists#return list</code>

ใช้ในการ read ข้อมูล feature_name ทั้งหมดจาก name

บรรทัดที่ 106 ประกาศ ฟังก์ชัน readname

บรรทัดที่ 108 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 109 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก

class conn

บรรทัดที่ 110 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql ข้อมูลที่ใช้

บรรทัดที่ 111 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 112 ประกาศ list lists

บรรทัดที่ 113 ลูป ค่า i โดนขอบเขตเป็นค่าที่ได้จาก ฟังก์ชัน fetchall เพื่อดึงค่าที่

ต้องการออกมา

บรรทัดที่ 114 ทำการ append i ไปยัง lists

บรรทัดที่ 115 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-76 ทำการประกาศฟังก์ชัน getplantname โดยกำหนด คำสั่ง raw sql ในการ อ่านชื่อ plant โดยอ้างอิงจาก index ที่ได้รับ

ตารางที่ 4-76 function getplantname

116	<code>def getplantname(self,index):</code>
117	<code>#raw sql</code>
118	<code>sql="" select plant_name FROM plant</code>
119	<code>WHERE "plant_id" = (%s)""</code>
120	<code>cur=self.conn.cursor()#define cursor of this connection</code>
121	<code>cur.execute(sql,(index,))#execute command by use rawsql with i</code>
122	<code>self.conn.commit()#commit execute of command</code>
123	<code>value=cur.fetchall()#define value that get value by fetch all val</code>
124	<code>return value#return list of tuple name value</code>

ใช้ในการ read ข้อมูล plant_name จาก plant โดยดูจาก plant_id

บรรทัดที่ 116 ประกาศ ฟังก์ชัน getplantname

บรรทัดที่ 118 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 120 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก

class conn

บรรทัดที่ 121 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql ข้อมูลที่ใช้

บรรทัดที่ 122 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 123 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 124 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-77 ทำการประกาศฟังก์ชัน getplantindex โดยกำหนด คำสั่ง raw sql ในการ อ่านชื่อ plant index โดยคำนวณจาก index feature ที่ได้รับ

ตารางที่ 4-77 function findplantindex

125	<code>def findplantindex(self,input):</code>
126	<code> sizereader=pd.read_csv(path+'size.csv')</code>
127	<code> size=sizereader['size']#definid size value to store serie</code>
128	<code> size[0]=1#set value of series name size at index 0 to 1</code>
129	<code> for i in range(1,len(size)):</code>
130	<code> #check if (input is more than or equal to size at index i-1 and input is less than or equal to size at index i)</code>
131	<code> if((input >= size[i-1]) and (input <= size[i])):#then do inside</code>
132	<code> return i#return value</code>

ใช้ในการ หาเลขindex จาก csv ของ size

บรรทัดที่ 125 ประกาศ ฟังก์ชัน findplantindex

บรรทัดที่ 126 ทำการเรียกฟังก์ชัน read_csv จาก pandas

บรรทัดที่ 127 ประกาศ series size

บรรทัดที่ 128 กำหนด size ที่ 0 = 1

บรรทัดที่ 129 ลูป ค่า i โดน range ตั้งแต่ 1 ถึง ขนาด ของ size - 1

บรรทัดที่ 131 เช็คเงื่อนไขว่า input มากกว่า หรือ เท่ากับ size ที่ i-1 และ input น้อยกว่า หรือ เท่ากับ size ที่ i หรือไม่

บรรทัดที่ 132 คืนค่ากลับไปด้วยค่าที่ต้องการ

ตาราง 4-78 ทำการประกาศฟังก์ชัน getallfeaturename โดยกำหนด คำสั่ง raw sql ในการอ่านชื่อ featurename ทั้งหมดโดยผ่านการตัวบางตัวออกเนื่องไม่ได้ใช้อ้างอิง

ตารางที่ 4-78 function findplantindex

133	<code>def getallfeaturename(self):</code>
134	<code>#raw sql</code>
135	<code>sql="" select feature_name FROM featurename ""</code>
136	<code>cur=self.conn.cursor()#define cursor of this connection</code>
137	<code># print(input)</code>
138	<code>cur.execute(sql)#execute command by use rawsql</code>
139	<code>self.conn.commit()#commit execute of command</code>
140	<code>value=cur.fetchall()#define value that get value by fetch all value from commit</code>
141	<code>return value#return list of tuple name value</code>

ใช้ในการ read ข้อมูล feature_name ทั้งหมดจาก featurename โดยที่ไม่มี featurename ที่ไม่ได้ใช้แสดงผล

บรรทัดที่ 133 ประกาศ ฟังก์ชัน getallfeaturename

บรรทัดที่ 135 ประกาศ sql query string ชื่อ sql

บรรทัดที่ 136 ประกาศ cursor สำหรับการเชื่อมต่อ โดย ทำการเรียกฟังก์ชัน cursor จาก

class conn

บรรทัดที่ 138 ทำการเรียกฟังก์ชัน execute เพื่อกำหนด คำสั่ง sql ข้อมูลที่ใช้

บรรทัดที่ 139 ทำการเรียกฟังก์ชัน commit เพื่อเริ่มการทำงาน คำสั่ง

บรรทัดที่ 140 ทำการเรียกฟังก์ชัน fetchall เพื่อดึงค่าที่ต้องการออกมา

บรรทัดที่ 141 คืนค่ากลับไปด้วยค่าที่ต้องการ

4.3. ตารางการปฏิบัติงานระยะเวลา 6 เดือน

การปฏิบัติงานตลอด 6 เดือน มีการจัดบันทึกตลอดเพื่อกำหนดระยะเวลาในการพัฒนาแต่ละส่วน โดยอาศัยการประชุมกับพี่เลี้ยง และ อาจารย์เพื่อให้งานดำเนินไปได้ด้วยดี ดังนั้นผู้จัดทำจึงได้สรุป ระยะเวลาปฏิบัติได้ดังนี้

ตารางที่ 4-79 ตารางการปฏิบัติงาน

การปฏิบัติงาน	Jul	Aug	Sep	Oct	Nov	Dec
สร้าง Virtual assistance ของ UNIT3 AI supervisory						
สร้าง Virtual assistance ของ UNIT4 AI supervisory สร้าง weekly report ของ AI supervisory						
Virtual assistance ของ Anomaly Detection						
พัฒนาระบบ Data base สำหรับ Virtual assistance						
ทำ report ,KM documents และ presentation						

จากตาราง 4-79 ซึ่งแสดงเวลาที่จัดทำโครงการและรวบรวมเอกสารสรุปได้ดังนี้

ในระยะสองเดือนแรก ผู้จัดทำได้เก็บ requirement และ สร้าง Virtual assistance ของ UNIT3 AI supervisory หรือ ที่เรียกว่า opportunity loss ในช่วงเดือนถัดมา ผู้จัดทำได้สร้าง Virtual assistance ของ UNIT4 AI supervisory หรือ feature drift และ สร้าง weekly report ของ AI supervisory 2 unit โดยในระหว่างขั้นนี้ผู้จัดทำได้พบ case study ของ anomaly detection ทำให้ในช่วงเดือนที่สี่ ผู้จัดทำได้พัฒนา Virtual assistance ของ Anomaly Detection เพื่อรับมือ case นี้

ในเดือนที่ห้านี้ ผู้จัดทำได้ปรับปรุงระบบการเก็บข้อมูลเนื่องจากการล่มของAKS จึงเกิดการพัฒนา ระบบ Data base สำหรับ Virtual assistance ขึ้นมา โดยในเดือนสุดท้ายนี้จะเป็นการรวบรวม และ ปรับปรุงข้อบกพร่องต่างๆ ให้มากที่สุด และเริ่มจัดทำเอกสารรายงาน เอกสารความรู้ และ สไลด์ นำเสนอขึ้น โดยในระหว่างแต่ละเดือนมีการจัดทำสไลด์นำเสนอต่อบริษัทในการแสดงความคืบหน้า ของงานที่ได้รับ

จากที่กล่าวมาทั้งหมดนี้แสดงให้เห็นว่าหลังจากการวางแผนพัฒนาแล้วเราจะต้องเลือกเครื่องมือที่เหมาะสมกับการทำงานไม่ว่าจะเป็นSoftware hardware library service เพื่อให้งานสามารถพัฒนา ได้อย่างเหมาะสมและมีประสิทธิภาพ โดยในส่วน การพัฒนาจะต้องมีการแบ่งออกเป็น 3 ส่วนได้แก่ ส่วน feature drift ส่วน Gap benefit ส่วน anomaly detection ส่วน database ดังนั้นในการ พัฒนาสิ่งที่สำคัญที่สุดคือการกำหนดเวลาเพราะหากเราแบ่งเวลาได้ไม่ดี การพัฒนาอาจทำได้ไม่เต็มที่ เนื่องจากกระยะเวลานั้นไม่เหมาะสม

บทที่ 5

ผลการดำเนินงาน

หลังจากการพัฒนาเสร็จสมบูรณ์สิ่งที่ผู้ใช้งานได้รับคือประสบการณ์ ดังนั้นบทนี้จะกล่าวถึงผลการดำเนินงาน ไม่ว่าจะเป็น ส่วนของผู้ใช้งาน หรือ ส่วนฐานข้อมูล ซึ่งจะอธิบายต่อไปนี้

5.1. ส่วนผลลัพธ์ที่ผู้ใช้งานได้รับ

5.1.1. UNIT3 AI supervisory

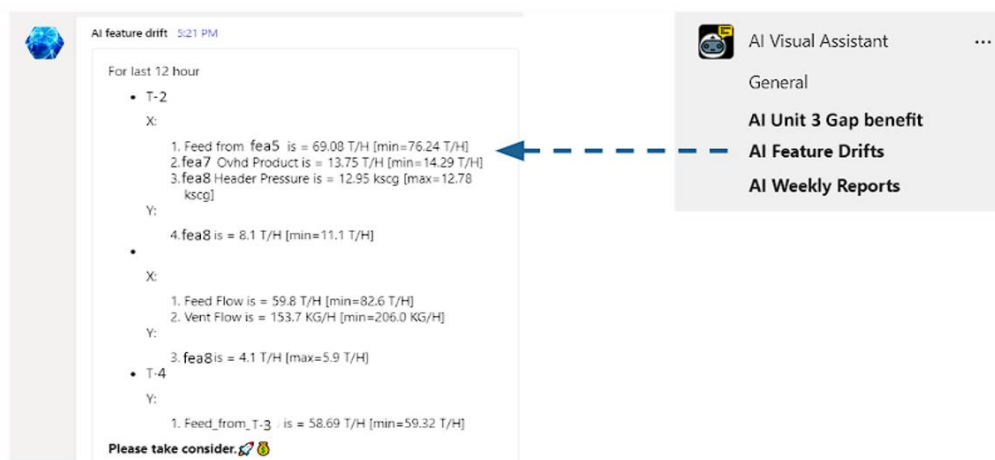


ภาพที่ 5-1 การแสดงผลของ webhook AI Unit 3 Gap benefit

จากรูปภาพ 5-1 โดยเมื่อระบบทำงานในแต่ละวันผู้ใช้งานจะส่งได้รับคือ message ในช่องที่ผู้ใช้งานเข้าร่วม โดยจะมีการแจ้งของระบบ

เมื่อเปิดดู ข้อความจะมีเวลากำกับและมีการจัดบรรทัดแบบรายงาน แสดง unit ที่เกิด Opportunity loss พร้อมเม็ดเงินที่สูญเสียไป รวมถึง ระยะเวลาที่เกิดไปกี่ ชม. โดยไม่เกิน 12 ชม. โดยด้านล่างจะมีปุ่ม PIVIEW สำหรับการเข้าเว็บ dashboard สำหรับรายละเอียดเพิ่มเติม

5.1.2. UNIT4 AI supervisory

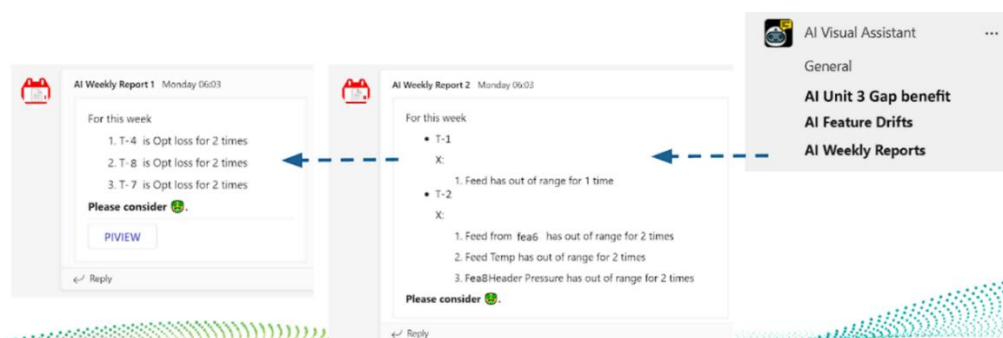


ภาพที่ 5-2 การแสดงผลของ webhook AI Feature Drift

จากรูปภาพ 5-2 โดยเมื่อระบบทำงานในแต่ละวันผู้ใช้งานจะส่งได้รับคือ message ในช่องที่ผู้ใช้งานเข้าร่วม โดยจะมีการแจ้งของระบบ

เมื่อเปิดดู ข้อความจะมีเวลากำกับและมีการจัดบรรทัดในรูปแบบรายงาน แสดง plant ที่เกิด feature drift โดยแบ่งเป็น X,Y พร้อมค่า feature ที่ out of range ไป รวมถึง ค่า min max เท่าไรที่กำหนดไว้

5.1.3. AI Weekly report



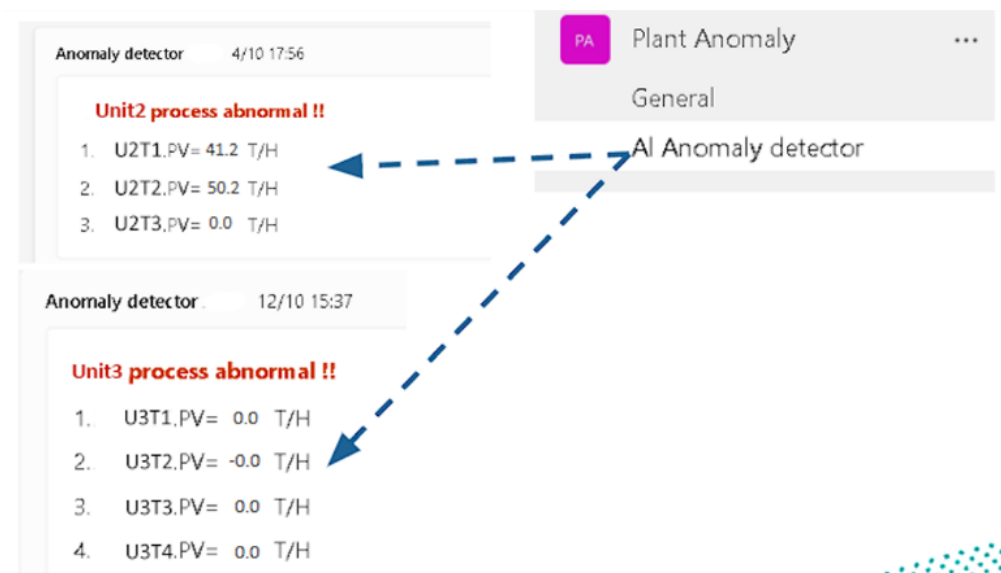
ภาพที่ 5-3 การแสดงผลของ webhook AI Weekly report

จากรูปภาพ 5-3 โดยเมื่อระบบทำงานครบ 7 วันผู้ใช้งานจะส่งได้รับคือ message ในช่องที่ผู้ใช้งานเข้าร่วม โดยจะมีการแจ้งของระบบ

เมื่อเปิดดู ข้อความจะมีเวลากำกับและมีการจัดบรรทัดในรูปแบบรายงาน แสดง plant ที่เกิด Opportunity loss (report1) หรือ feature drift(report2) โดย

report1 จะแสดง unit ที่เกิด Opportunity loss พร้อม จำนวนครั้งที่เกิด โดยด้านล่างจะมีปุ่ม PIVIEW สำหรับการเข้าเว็บ dashboard สำหรับรายละเอียดเพิ่มเติม แต่ report2 จะแสดง plant ที่เกิด feature drift แบ่งเป็น X ,Y พร้อม ชื่อ feature ที่ out of range โดยนับจำนวนที่เกิดไป รวมถึง

5.1.4. AI Anomaly detection

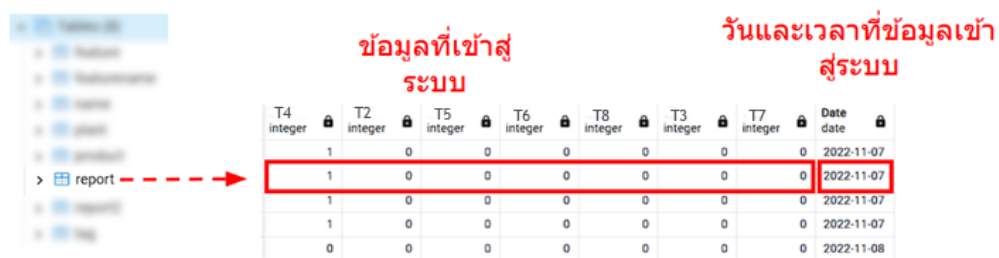


ภาพที่ 5-4 การแสดงผลของ webhook AI Anomaly detector ของ Unit 2 ,3

จากรูปภาพ 5-4 โดยเมื่อระบบทำงานในแต่ละวันผู้ใช้งานจะส่งได้รับคือ message ในช่องที่ผู้ใช้งานเข้าร่วม โดยจะมีการแจ้งของระบบ

เมื่อเกิด anomaly data โดยเมื่อเปิดดู ข้อความจะมีเวลากำกับและมีการจัดบรรทัดในรูปแบบรายงานแสดง unit ที่เกิด anomaly พร้อมค่า feature ที่ เป็น anomaly ไปรวมถึงค่าที่มีปัญหา

5.2. ส่วนของฐานข้อมูล



ข้อมูลที่เข้าสู่ระบบ **วันและเวลาที่ข้อมูลเข้าสู่ระบบ**

T4 integer	T2 integer	T5 integer	T6 integer	T8 integer	T3 integer	T7 integer	Date date
1	0	0	0	0	0	0	2022-11-07
1	0	0	0	0	0	0	2022-11-07
1	0	0	0	0	0	0	2022-11-07
0	0	0	0	0	0	0	2022-11-08

ภาพที่ 5-5 ข้อมูลที่เข้าสู่ ฐานข้อมูล

จากรูปภาพ 5-5 โดยเมื่อระบบทำงานในแต่ละวัน ข้อมูล report 1 และ report2 จะถูกนำเข้ามาบันทึกผ่านระบบฐานข้อมูล โดยมีเวลากำกับไว้ ซึ่งหากต้องการ query ข้อมูลสามารถใช้ระบบ หรือ ทำการ query โดยตรงได้

หลังการพัฒนาเสร็จสิ้น รวมถึงการแก้ไขข้อบกพร่องเรียบร้อยแล้ว ผลลัพธ์ที่ได้มาจะแบ่งออกเป็น ส่วนของผู้ใช้งานระบบคือ ส่วนการแจ้งเตือน พร้อมข้อความของ Microsoft Team channel ที่กำหนดไว้ โดย สามารถดูได้ทันทีและสะดวกสบาย และ ส่วนของฐานข้อมูลที่สามารถนำไปใช้งานต่อในระบบอื่นได้เนื่องจากสามารถใช้อ้างอิงด้วยเวลา และสามารถทำการร้องขอข้อมูลได้หลังจากนี้จะเป็นบทสรุปและแนวทางการพัฒนาต่อ โดยจะกล่าวในบทต่อไป

บทที่ 6

บทสรุปและแนวทางในการพัฒนาต่อ

หลังจากสิ้นสุดการปฏิบัติงานแล้วระบบผู้ช่วยเสมือนจริงสำหรับการกำกับดูแลปัญญาประดิษฐ์ มีผลของการดำเนินงานปัญหาที่พบในขณะพัฒนาระบบ รวมถึงข้อเสนอแนะในการนำเอาระบบไปพัฒนาต่อยอดในอนาคต มีรายละเอียดดังนี้

6.1. สรุปผลการดำเนินงาน

ผลจากการดำเนินงานของระบบ ผู้ช่วยเสมือนจริงสำหรับการกำกับดูแลปัญญาประดิษฐ์ สามารถนำไปเป็นต้นแบบในการพัฒนาต่อและสามารถใช้งานได้จริง โดยสามารถสรุปผลการดำเนินงานได้ดังนี้

- 6.1.1. ผู้ใช้สามารถรับการแจ้งเตือนผ่าน Microsoft Team ราย 12 ชม.ได้
- 6.1.2. ผู้ใช้สามารถรับการแจ้งเตือนผ่าน Microsoft Team ราย 7 วันได้
- 6.1.3. ผู้ใช้สามารถรับการแจ้งเตือนผ่าน Microsoft Team เมื่อเกิดความผิดปกติทันทีได้ภายในวัน 1 ครั้งต่อวัน

6.2. ปัญหาที่พบในการดำเนินงาน

6.2.1. พบ feature drift มาด้วยจำนวนความถี่สูง ที่ UNIT4 เนื่องจาก Unit shut down แล้วเราไม่ทราบ ทำให้ค่าข้อมูลการ run เครื่องจักรในปัจจุบันมีค่าเกินย่านของข้อมูลที่นำมาสอน AI

6.2.2. เดิมทีระบบ AI supervisory virtual assistant ถูกพัฒนาให้เก็บข้อมูลผ่าน csv file ที่ถูก save ไว้ใน Azure Kubernetes service แต่พบปัญหาว่า เมื่อ Kubernetes cluster ล่มข้อมูลที่เก็บอยู่ใน csv file จะหายไป ส่งผลให้ไม่สามารถนำข้อมูลไปทำ weekly report ต่อได้

6.3. แนวทางในการแก้ปัญหา

6.3.1. สร้างระบบในส่วน Anomaly detection เข้าไปเพื่อใช้เป็น ข้อมูล proof ความผิดปกติที่เกิดขึ้นกับ unit

6.3.2. ปรับปรุงระบบการจัดเก็บข้อมูลให้มีประสิทธิภาพมากขึ้นโดยอาศัยระบบฐานข้อมูล

6.4. แนวทางการพัฒนาต่อ

แนวทางการพัฒนาต่อผู้ช่วยเสมือนจริงสำหรับการกำกับดูแลปัญญาประดิษฐ์ให้สามารถ ทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้นโดยการทำให้ระบบสามารถแนะนำการดำเนินการต่อไป เมื่อได้รับการแจ้งเตือนที่สำคัญ รวมไปถึงการแสดงผลในรูปแบบอื่นๆของตัวระบบ

การทั้งกล่าวมาทั้งหมดนี้ล้วนเป็นบทสรุปทั้งหมดในการพัฒนา และเป็นเป้าหมายที่ประสบความสำเร็จด้วยดี นอกจากนี้ปัญหาระหว่างพัฒนาที่พบเจอ จะต้องอาศัย ความรู้ที่มี ประสบการณ์ที่สะสม รวมถึงความเหลือจากอาจารย์ที่ปรึกษา และพี่เลี้ยง ในการหาแนวทางการในแก้ปัญหาจนได้ผลลัพธ์ที่ต้องการออกมาอย่างราบรื่น โดยตัวระบบนั้นยังสามารถพัฒนา ได้อีกหลายหลายวิธี จึงสามารถสรุปออกมาเป็น แนวความคิดที่สามารถต่อยอดหรือพัฒนาต่อได้เป็นอย่างดีที่สุดบรรณานุกรม

บรรณานุกรม

- Achieve.Plus. (30 July 2020). รู้จักใช้ Scikit-learn เหมือนมีโปรในเกม. เข้าถึงได้จาก medium: <https://medium.com/achieve-space/รู้จักใช้-scikit-learn-เหมือนมีโปรในเกม-4cebd3195518>
- Amazon Web Services, Inc. (- - 2023). Python คืออะไร. เข้าถึงได้จาก : https://aws.amazon.com/what-is/python/?nc1=h_ls
- Arnon Puitrakul. (24 september 2020). Parallelise Python แบบปloom ๆ ด้วย Joblib. เข้าถึงได้จาก arondora: <https://arondora.in.th/parallelise-python-using-joblib/>
- BorntoDev Co., Ltd. (16 April 2020). พื้นฐานการใช้ NumPy. เข้าถึงได้จาก borntodev: <https://www.borntodev.com/2020/04/16/พื้นฐานการใช้-numpy-ใน-python-3/>
- Federico Di Gregorio. (- - -). psycopg2.PyPi. เข้าถึงได้จาก PyPi: <https://pypi.org/project/psycopg2/>
- Ian Cordasco ,Cory Benfield Ben Toews. (- - -). GitHub - requests-requests-ntlm- NTLM authentication support for Requests-. เข้าถึงได้จาก GitHub: <https://github.com/requests/requests-ntlm>
- Kenneth Reitz. (- - -). Requests: HTTP for Humans™ - Release v2.28.2 documentation. เข้าถึงได้จาก Requests: HTTP for Humans™: <https://requests.readthedocs.io/en/latest/>
- Microsoft Corporation. (- - -). azure-identity.PyPi. เข้าถึงได้จาก pypi: <https://pypi.org/project/azure-identity/>
- Microsoft Corporation. (- - -). azure-keyvault-secrets.PyPi. เข้าถึงได้จาก pypi: <https://pypi.org/project/azure-keyvault-secrets/>
- Panchart Mittrakul. (30 November 2021). รู้จัก pandas – Library อันดับ 1 สำหรับการทำ Data Analysis. เข้าถึงได้จาก <https://blog.skooldio.com/what-is-pandas/>
- Saksit Srimarong. (28 August 2020). 4 ประเภทของการแบ่งกลุ่มข้อมูล (Clustering). (Government Big Data Institute (GBDi)) เรียกใช้เมื่อ 28 april 28 จาก <https://bigdata.go.th/big-data-101/4-types-of-clustering/>
- Saptak Chaudhuri. (29 What is PostgreSQL? 2022). What is PostgreSQL? เข้าถึงได้จาก <https://appmaster.io/blog/what-is-postgresql>

บรรณานุกรม (ต่อ)

- wannaphong. (16 april 2015). *urllib เข้าถึงอินเทอร์เน็ตใน Python 3*. เข้าถึงได้จาก Python 3: <https://python3.wannaphong.com/2015/04/urllib-python-3.html>
- เอิญ สุริยะฉาย. (2021). *Database Design and Applications*. Bangkok: Department of Computer and Information Science CIS kmutnb.
- ภาคภูมิ สารพัฒน์ นนทวิทย์ ชิวเรืองโรจน์ และ ปพจน์ ธรรมเจริญพร. (8 april 2020). *อีกขั้นของ k-means algorithm ที่สามารถแบ่งกลุ่มข้อมูลได้ทุกประเภท*. (Government Big Data Institute (GBDi)) เรียกใช้เมื่อ 28 april 2023 จาก <https://bigdata.go.th/big-data-101/k-means-algorithm-for-clustering-large-data-sets-with-categorical-values/>
- ลือพล พิพานเมฆาภรณ์. (2021). *Principle of Machine Learning*. ใน *040613467 Machine Learning* (หน้า 2-6). Bangkok: Department of Computer and Information Science CIS kmutnb.
- สถิตย์ ประสมพันธ์. (2020). *Clustering*. ใน *Data Science* (หน้า 3). bangkok.

ภาคผนวก ก
วิทยาการคอมพิวเตอร์

[1] เอกสารโค้ดเต็ม : <https://github.com/GTAHITMAN12/coopdoc>

