การจัดการคลังสินค้าด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง

INVENTORY MANAGEMENT UTILIZING IoT AND MICROSERVICES WITH MACHINE LEARNING MODEL FOR PRODUCT ORDER PREDICTION

ชวินกร ปิยะธนานุกูล

CHAVINKORN PIYATANANUGOON

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2566

KMITL-2023-EN-M-027-077

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INVENTORY MANAGEMENT UTILIZING IoT AND MICROSERVICES WITH MACHINE LEARNING MODEL FOR PRODUCT ORDER PREDICTION

CHAVINKORN PIYATANANUGOON

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2023

KMITL-2023-EN-M-027-077

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2023

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การจัดการคลังสินค้าด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง
นักศึกษา	นาย ชวินกร ปิยะธนานุกูล
รหัสประจำตัว	65016024
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
พ.ศ.	2566
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.อรรถสิทธิ์ หล่าสกุล

บทคัดย่อ

งานวิจัยนี้เสนอการประยุกต์ใช้อินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices เข้ากับการจัดการคลังสินค้า โดยการพัฒนาระบบติดตามการทำงานขนส่งสินค้าบนรถโฟล์คลิฟท์และแสดงผลการทำงานผ่านเว็บแอปพลิเคชันแบบเวลาจริง คุณสมบัติของการใช้อินเทอร์เน็ตในทุกสรรพสิ่งในการจัดการคลังสินค้าทำให้สามารถเชื่อมต่ออุปกรณ์ต่างๆ ในคลังสินค้าเพื่อรวบรวมข้อมูลไปใช้ติดตามและวิเคราะห์ต่อได้ และการใช้ Microservices ในการออกแบบระบบสามารถพัฒนาแบ่งส่วนการทำงานของระบบออกเป็นส่วนย่อย ๆ ที่ทำงานแยกกันและสามารถเชื่อมต่อการทำงานถึงกันได้ ช่วยให้การทำงานยืดหยุ่น ง่ายต่อการขยายและพัฒนาระบบได้ในอนาคต วิทยานิพนธ์ฉบับนี้ยังเสนอการทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง เพื่อนำข้อมูลมาจัดการวางแผนคลังสินค้า รวมถึงการเตรียมการผลิตให้เหมาะสมกับปริมาณของอุปสงค์และอุปทานของตลาด ประสิทธิภาพของแบบจำลองจะวัดด้วยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (MAE) ค่าความคลาดเคลื่อนกำลังสอง (RMSE) และค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (MAPE) โดยแบบจำลองการถดถอยเชิงเส้นมีค่า MAPE เท่ากับร้อยละ 17.51 แบบจำลอง Random Forest Regression มีค่า MAPE เท่ากับร้อยละ 11.91 และแบบจำลอง Extreme Gradient Boosting มีค่า MAPE เท่ากับร้อยละ 11.04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	Inventory Management Utilizing IoT and Microservices with Machine Learning Model for Product Order Prediction
Student	Mr. Chavinkorn Piyatananugoon
Student ID.	65016024
Degree	Master of Engineering
Program	Electrical And Computer Engineering
Year	2023
Thesis Advisor	Assoc. Prof. Dr. Attasit Lasakul

ABSTRACT

This research presents the application of Internet of Things and Microservices in inventory management. The study focuses on developing a system that tracks cargo operations on forklift vehicles and displays real-time tracking results through a web application. By leveraging Internet of Things technology, multiple devices in the warehouse can be connected via the internet, facilitating data collection for tracking and analysis purposes. The utilization of Microservices in system design allows for the division of the system into individual services that work independently but can communicate and transport data between each other. This design approach offers flexibility to add more features and facilitates scalability for future system expansion. In addition, this thesis proposes the use of machine learning models to predict product order volume. This prediction helps in understanding the trends and future volumes of product orders, which in turn assists in warehouse planning and production preparation to meet market demand and supply. The performance of the models is evaluated using the mean absolute error (MAE), the root mean squared error (RMSE) and the Mean Absolute Percentage Error (MAPE). The Linear Regression model achieved a MAPE of 17.51%. The Random Forest Regression model achieved a MAPE of 11.91%. Finally, the Extreme Gradient Boosting model achieved a MAPE of 11.04%.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาและความอนุเคราะห์จากอาจารย์ที่ปรึกษา รศ.ดร. อรรถสิทธิ์ หล่าสกุล ที่ให้คำปรึกษาและคำแนะนำชี้แนะต่างๆตลอดจนให้ความรู้ที่ดีแก่ข้าพเจ้า ๆ ขอขอบพระคุณอย่างยิ่ง

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่ให้การอุปการะอบรมเลี้ยงดู รวมถึงให้คำปรึกษาแนะนำต่างๆ ตลอดจนส่งเสริมให้โอกาสทางการศึกษา และให้กำลังใจแก่ข้าพเจ้าเป็นอย่างดี

สำหรับประโยชน์อันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง และบูรพาจารย์ที่เคารพทุกท่าน รวมทั้งผู้มีพระคุณทุกท่าน

ชวินกร ปิยะธนานุกูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VII
สารบัญรูป.....VIII
บทที่ 1 บทนำ..... 1
1.1 ที่มาและความสำคัญของปัญหา..... 1
1.2 จุดมุ่งหมายและวัตถุประสงค์..... 2
1.3 แนวคิดและทฤษฎีที่ใช้ในการวิจัย..... 2
1.4 ขอบเขตงานวิจัย..... 2
1.5 ขั้นตอนการศึกษา..... 3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 4
2.1 เครื่องมือที่เกี่ยวข้อง..... 4
2.1.1 Python..... 4
2.1.2 OpenCV..... 5
2.1.3 pyzbar..... 5
2.1.4 JavaScript Object Notation (JSON)..... 5
2.1.5 RESTful API..... 6
2.1.6 MQTT..... 6
2.1.7 TypeScript..... 7
2.1.8 Nest JS..... 7
2.1.9 Mongo DB..... 8
2.1.10 AWS EC2..... 8
2.1.11 React JS..... 9
2.1.12 Socket.io..... 9
2.1.13 Google Colab..... 10
2.1.14 Pandas..... 10
2.1.15 Matplotlib..... 11
2.1.16 Seaborn..... 12
2.1.17 Scikit-learn..... 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

2.2 ทฤษฎีการวิเคราะห์การถดถอยเชิงเส้น 12
2.3 ทฤษฎี extreme gradient boosting regression (XGBoost) 14
2.4 ทฤษฎี random forest regression 16
2.5 การวัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง 17
2.6 ทบทวนวรรณกรรม 19
บทที่ 3 วิธีดำเนินการทดลอง 21
3.1 บทนำ 21
3.2 การติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าด้วยอินเทอร์เน็ตในทุกสิ่ง (internet of things) และ microservices 23
3.2.1 การออกแบบอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) และสถาปัตยกรรม Microservices 23
3.2.2 การพัฒนา อินเทอร์เน็ตในทุกสิ่ง (Internet of Things) ร่วมกับโพรโทคอล MQTT 25
3.2.3 การพัฒนาสถาปัตยกรรม Microservices 29
3.2.4 กระบวนการทำงานกับโครงสร้างภายในคลังสินค้า 35
3.3 การทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง 36
3.3.1 การเตรียมข้อมูลสำหรับการเรียนรู้ของเครื่อง 36
3.3.2 การเรียนรู้ด้วยแบบจำลองการเรียนรู้ของเครื่อง 41
3.3.3 การทำนายผลและวัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง 42
บทที่ 4 ผลการทดลอง 44
4.1 บทนำ 44
4.2 ผลการทำงานระบบติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้า 44
4.3 ผลลัพธ์ของการทำนายปริมาณสินค้าของชุดข้อมูล forecasts for product demand ด้วยแบบจำลองการเรียนรู้ของเครื่อง 51
4.3.1 ผลลัพธ์การเตรียมและแบ่งชุดข้อมูล 51
4.3.2 ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น 53
4.3.3 ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลอง Extreme Gradient Boosting 54
4.3.4 ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลอง Random Forest Regression 56
4.3.5 ผลลัพธ์การเปรียบเทียบประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องทั้ง 3 แบบจำลอง 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่ 5 สรุปผลการวิจัย 60
5.1 สรุปผลและวิจารณ์การทดลอง 60
5.2 ข้อเสนอแนะ 61
บรรณานุกรม 62
ภาคผนวก ก. 66
ประวัติผู้เขียน 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า
3.1 อธิบายค่าต่างๆของข้อมูลจาก QR Code รูปแบบ JSON 26
3.2 องค์ประกอบ API ของเซอร์วิสคลังสินค้า 30
3.3 โครงสร้างข้อมูล JSON สำหรับสื่อสารผ่าน Socket.IO กับเซอร์วิสเว็บแอปพลิเคชัน 31
3.4 องค์ประกอบ API ของเซอร์วิสการยืนยันตัวตน 32
3.5 คุณสมบัติของชุดข้อมูล Forecasts for Product Demand 37
4.1 ผลลัพธ์การเรียก API กรณีไม่พบข้อมูลพนักงาน 44
4.2 ผลลัพธ์การเรียก API ตรวจสอบสถานการณ์ใช้รถโฟล์คลิฟท์ 45
4.3 ผลลัพธ์การเข้าสู่ระบบของ API เซอร์วิสยืนยันตัวตน 46
4.4 ผลลัพธ์การเรียก API เซอร์วิสคลังสินค้าเพื่อปรับปรุงข้อมูลการใช้รถโฟล์คลิฟท์ 50
4.5 ผลลัพธ์การออกจากระบบของ API เซอร์วิสยืนยันตัวตน 50
4.6 ผลลัพธ์การวัดประสิทธิภาพแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) 54
4.7 ผลลัพธ์การวัดประสิทธิภาพแบบจำลอง Extreme Gradient Boosting 56
4.8 ผลลัพธ์การวัดประสิทธิภาพแบบจำลอง Random Forest Regression 57
4.9 ผลลัพธ์การเปรียบเทียบความคลาดเคลื่อน MAE RMSE และ MAPE 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 ภาษาไพทอน 4
2.2 ไลบรารี OpenCV 5
2.3 JavaScript Object Notation 6
2.4 โพรโทคอล MQTT 7
2.5 เฟรมเวิร์ค Nest JS 8
2.6 ฐานข้อมูล MongoDB 8
2.7 Amazon Elastic Compute 9
2.8 กระบวนการทำงานของ React JS 9
2.9 ไลบรารี Socket.io 10
2.10 Google Colab 10
2.11 ไลบรารี pandas 11
2.12 ตัวอย่างกราฟของ Matplotlib 11
2.13 ตัวอย่างกราฟของ Seaborn 12
2.14 ตัวอย่างความสัมพันธ์กราฟเส้นตรงของการวิเคราะห์การถดถอยเชิงเส้น 13
2.15 ตัวอย่างกระบวนการทำงานของ Extreme Gradient Boosting 15
2.16 ตัวอย่างกระบวนการทำงานของ Random Forest 17
3.1 กระบวนการติดตามการทำงานรถโฟล์คลิฟท์ภายในคลังสินค้า 21
3.2 กระบวนการทำงานของแบบจำลองการทำนายปริมาณสินค้าด้วยการเรียนรู้ของเครื่อง 22
3.3 การออกแบบโครงสร้างอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) และ สถาปัตยกรรม Microservices 24
3.4 ขั้นตอนการทำงานอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) บนรถโฟล์คลิฟท์ 25
3.5 ตัวอย่างข้อมูล JSON จากการแปลงภาพ QR Code 25
3.6 การตั้งค่า EC2 Instance 26
3.7 การตั้งค่าการเชื่อมต่อกับ EC2 Instance 27
3.8 การทดสอบเชื่อมต่อกับ Console EC2 27
3.9 ผลการติดตั้ง Eclipse Mosquitto Broker 28
3.10 การตั้งค่า Eclipse Mosquitto Broker 28
3.11 การทดสอบเชื่อมต่อกับ Eclipse Mosquitto Broker (ก) การเชื่อมต่อ 28
3.12 โครงสร้างสถาปัตยกรรม Microservices 29
3.13 แผนผังลำดับการทำงานของเซอร์วิสคลังสินค้า 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

3.14 แผนผังลำดับการทำงานของเซอร์วิสการยืนยันตัวตน 33
3.15 แผนผังลำดับการทำงานของเซอร์วิสเว็บแอปพลิเคชัน 34
3.16 แบบจำลองพื้นที่ภายในคลังสินค้า 35
3.17 ขั้นตอนการทำงานภายในคลังสินค้า 36
3.18 ตัวอย่างชุดข้อมูล DataFrame 37
3.19 การตรวจสอบข้อมูลสูญหาย (ก) ผลลัพธ์การตรวจสอบข้อมูลสูญหาย (ข) ผลลัพธ์การหลังการกำจัดข้อมูลสูญหาย 38
3.20 ข้อมูล DataFrame ตามกลุ่มประเภทสินค้าและจำนวนสินค้าต่อปี 38
3.21 กราฟข้อมูลตามกลุ่มประเภทสินค้าและจำนวนสินค้าต่อปี 39
3.22 ข้อมูล DataFrame ปริมาณสินค้าโดยรวมต่อปี 39
3.23 กราฟข้อมูลปริมาณสินค้าโดยรวมต่อปี 40
3.24 กราฟข้อมูลปริมาณสินค้าโดยรวมต่อเดือน 40
3.25 กราฟข้อมูลปริมาณสินค้าโดยรวมตั้งแต่ปี 2012 ถึงปี 2016 41
3.26 กราฟการแบ่งข้อมูลชุดฝึกและชุดทดสอบแบบจุดกระจัดกระจาย 41
4.1 ตัวอย่างการค้นหาข้อมูลพนักงานพบในฐานข้อมูลยืนยันตัวตนผ่านโปรแกรม Robo 3T 45
4.2 ตัวอย่างการค้นหาข้อมูลการใช้รถโฟล์คลิฟท์ผ่านโปรแกรม Robo 3T (ก) ข้อมูลในฐานข้อมูลรถมีชื่อพนักงานอยู่ (ข) ข้อมูลในฐานข้อมูลกรณีรถไม่มีชื่อพนักงานอยู่ 45
4.3 ผลลัพธ์การบันทึกข้อมูลพนักงานลง local storage 46
4.4 ผลลัพธ์ข้อมูลรถผ่านพื้นที่ผ่านได้บน MQTT Broker 47
4.5 ผลลัพธ์ข้อมูลรถผ่านพื้นที่ห้ามผ่านบน MQTT Broker 47
4.6 ผลลัพธ์การค้นหาข้อมูลการใช้รถด้วย vehicleId ที่ฐานข้อมูลคลังสินค้า 48
4.7 ผลลัพธ์การบันทึกข้อมูลประวัติเส้นทางการทำงานรถโฟล์คลิฟท์ลงฐานข้อมูลคลังสินค้า 48
4.8 ผลลัพธ์การแสดงข้อมูลรถโฟล์คลิฟท์ผ่านพื้นที่อนุญาตให้ผ่านได้บนเว็บแอปพลิเคชัน 49
4.9 ผลลัพธ์การแสดงข้อมูลรถโฟล์คลิฟท์ผ่านพื้นที่ห้ามผ่านบนเว็บแอปพลิเคชัน 49
4.10 ผลลัพธ์การปรับปรุงข้อมูลสถานะการใช้รถโฟล์คลิฟท์ที่ฐานข้อมูลคลังสินค้า collection vehicle-data 50
4.11 ผลลัพธ์การลบข้อมูลรหัสพนักงานและชื่อพนักงานจาก local storage 51
4.12 ผลลัพธ์กราฟกระจัดกระจายปริมาณสินค้าต่อเดือนของปี 2011-2017 52
4.13 ผลลัพธ์กราฟกระจัดกระจายปริมาณสินค้าต่อเดือนของปี 2012-2016 52
4.14 ผลลัพธ์กราฟเส้นข้อมูลชุดฝึกและชุดทดสอบ 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

4.15 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าเปรียบเทียบกับกราฟกระจัดกระจายข้อมูลทดสอบ ปี2016ด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น 53
4.16 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น เปรียบเทียบกับกราฟเส้นชุดข้อมูลจริง 54
4.17 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าเปรียบเทียบกับกราฟกระจัดกระจายข้อมูลทดสอบ ปี2016ด้วยแบบจำลอง Extreme Gradient Boosting 55
4.18 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าด้วยแบบจำลอง Extreme Gradient Boosting เปรียบเทียบกับกราฟเส้นชุดข้อมูลจริง 55
4.19 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าเปรียบเทียบกับกราฟกระจัดกระจายข้อมูลทดสอบ ปี2016ด้วยแบบจำลอง Random Forest Regression 56
4.20 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าด้วยแบบจำลอง Random Forest Regression เปรียบเทียบกับกราฟเส้นชุดข้อมูลจริง 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การบริหารคลังสินค้าด้วยรูปแบบดั้งเดิมไม่มีการนำเทคโนโลยีมาใช้ การทำงานต่างๆจึงอยู่ในรูปแบบไม่อัตโนมัติ กระบวนการต่างๆจึงจำเป็นต้องพึ่งพาทรัพยากรมนุษย์เพียงอย่างเดียวในการเก็บข้อมูลต่างๆเพื่อนำมาใช้วิเคราะห์หรือวางแผนงาน ซึ่งส่วนมากจะอยู่ในรูปแบบของการจดในกระดาษหรือเก็บตามไฟล์แบบไม่เป็นระบบ รวมถึงการเก็บข้อมูลอาจเกิดข้อผิดพลาดได้จากความผิดพลาดของมนุษย์ (Human error) ส่งผลให้ข้อมูลที่ได้รับมีไม่เพียงพอที่ใช้ประกอบการตัดสินใจสำหรับฝ่ายบริหารงานในการวางแผนหรือประเมินผลการทำงาน ข้อมูลมีความผิดพลาดไม่สามารถนำมาใช้ทำนายผลล่วงหน้าได้ ข้อมูลจะไม่ได้รับแบบเวลาจริง (Lack of Real-time information) ซึ่งส่งผลให้ยากแก่การวางแผนอุปสงค์และอุปทานของการบริหารสินค้า ซึ่งปัญหาเหล่านี้อาจส่งผลกระทบต่อภาพรวมของการดำเนินธุรกิจได้ เนื่องจากการแข่งขันในหลากหลายองค์กรมีการเปลี่ยนแปลงที่รวดเร็วตามกระแสผู้บริโภค การทำงานต่างๆจึงต้องมีประสิทธิภาพมากที่สุดเพื่อให้ตอบสนองต่อความพึงพอใจของผู้บริโภค

หลากหลายองค์กรในปัจจุบันจึงมีการใช้เทคโนโลยีเพื่อการปรับเปลี่ยนการทำงานในองค์กร (Digital transformation) ให้มีประสิทธิภาพมากขึ้น โดยในงานวิจัยนี้จึงได้ประยุกต์ใช้อินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ในการทำระบบติดตามการทำงานของการขนส่งสินค้าภายในคลังสินค้าด้วยรถโฟล์คลิฟท์ให้ทราบถึงภาระงานของพนักงานในแต่ละวันรวมถึงความสามารถในการทำงานของพนักงานแต่ละคน เพื่อฝ่ายบริหารสามารถนำข้อมูลไปวิเคราะห์และจัดสรรวางแผนทรัพยากรแรงงานให้มีประสิทธิภาพมากขึ้น นอกจากนี้งานวิจัยนี้ยังใช้แบบจำลองการเรียนรู้ของเครื่องเพื่อทำนายปริมาณสินค้าล่วงหน้าเพื่อใช้ในการวางแผนอุปสงค์และอุปทานของการจัดการสินค้าให้ได้ประสิทธิภาพสูงขึ้น

การจัดการคลังสินค้าด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำนายปริมาณสินค้าล่วงหน้าด้วยแบบจำลองของเครื่องนี้เป็นเพียงการตัวอย่างประยุกต์ใช้งานในอุตสาหกรรมหรือธุรกิจที่ใช้คลังสินค้า โดยธุรกิจอื่นๆสามารถนำการออกแบบเทคโนโลยีนี้ไปประยุกต์ใช้ได้ เช่นภาคธุรกิจการเกษตร ธุรกิจด้านการแพทย์ ธุรกิจด้านพลังงาน การออกแบบเมือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัจฉริยะ เป็นต้นเพื่อเพิ่มประสิทธิภาพการทำงานให้เป็นระบบและทำงานได้อย่างอัตโนมัติมีประสิทธิภาพมากยิ่งขึ้น

1.2 จุดมุ่งหมายและวัตถุประสงค์

1. เพื่อพัฒนาระบบติดตามการทำงานของพนักงานและรถโฟล์คลิฟท์ด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices
2. เพื่อพัฒนาแบบจำลองการเรียนรู้ของเครื่องเพื่อทำนายปริมาณสินค้า

1.3 แนวคิดและทฤษฎีที่ใช้ในการวิจัย

หลักการของงานวิจัยนี้ประกอบด้วยการใช้อินเทอร์เน็ตในทุกสรรพสิ่งในการพัฒนาระบบติดตามรถโฟล์คลิฟท์ ซึ่งประกอบด้วยกล้องและอุปกรณ์ควบคุมจากนั้นรับส่งข้อมูลด้วยโพรโทคอล MQTT ไปยัง Microservices เพื่อประมวลผลข้อมูลตรรกะทางธุรกิจและจัดเก็บข้อมูลลงฐานข้อมูล จากนั้นส่งข้อมูลไปแสดงผลที่เว็บแอปพลิเคชัน และการใช้แบบจำลองการเรียนรู้ของเครื่องซึ่งประกอบด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) แบบจำลอง Extreme Gradient Boosting (XGBoost) และ แบบจำลอง Random Forest Regression ในการใช้ทำนายผลปริมาณสินค้า โดยก่อนการนำข้อมูลเข้าเรียนรู้กับแบบจำลองจะผ่านการเตรียมข้อมูลเพื่อทำความสะอาดข้อมูลและเลือกคุณสมบัติที่เกี่ยวข้องก่อน จากนั้นทำการวัดผลและทำนายผลของแบบจำลอง

1.4 ขอบเขตงานวิจัย

1. พัฒนาและทดสอบการทำงานของเว็บแอปพลิเคชันรับส่งข้อมูลกับอินเทอร์เน็ตในทุกสรรพสิ่งและ Microservices
2. ทำแบบจำลองการเรียนรู้ของเครื่องทำนายปริมาณสินค้าของชุดข้อมูล Forecasts for Product Demand [38] จำนวน 1,048,575 แถว และ 5 คอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เปรียบเทียบผลการทำนายและวัดประสิทธิภาพ แบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) แบบจำลอง Extreme Gradient Boosting (XGBoost) และแบบจำลอง Random Forest Regression ด้วยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root - Mean Squared Error) และค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute - Percentage Error)

1.5 ขั้นตอนการศึกษา

ขั้นตอนการศึกษาการจัดการคลังสินค้าด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง ประกอบไปด้วย 7 ขั้นตอน

1. สืบค้นและศึกษางานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้อินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ในการทำงานในอุตสาหกรรมต่างๆ
2. สืบค้นและศึกษางานวิจัยที่เกี่ยวข้องกับการใช้แบบจำลองการเรียนรู้ของเครื่องในการทำนายปริมาณสินค้า
3. ออกแบบพัฒนาระบบติดตามการทำงานรถโฟล์คลิฟท์ด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices
4. ประยุกต์ใช้แบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) แบบจำลอง Extreme Gradient Boosting (XGBoost) และ แบบจำลอง Random Forest Regression ในการทำนายปริมาณสินค้า
5. เก็บผลการทดลองการทำงานของระบบติดตามการทำงานรถโฟล์คลิฟท์ด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices
6. เก็บผลการทำนายและเปรียบเทียบประสิทธิภาพแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) แบบจำลอง Extreme Gradient Boosting (XGBoost) และแบบจำลอง Random Forest Regression ด้วยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root - Mean Squared Error) และค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute - Percentage Error)
7. สรุปผลการทดลอง การจัดการคลังสินค้าด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีและเครื่องมือที่เกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้ อันประกอบด้วยทฤษฎีของแบบจำลองการเรียนรู้ของเครื่อง การประเมินผลความแม่นยำ โพรโทคอลการสื่อสารต่างๆ รวมถึงวรรณกรรมที่เกี่ยวข้อง

2.1 เครื่องมือที่เกี่ยวข้อง

2.1.1 Python [1-2]

รูปที่ 2.1 ภาษาไพทอน

ภาษาไพทอนเป็นภาษาโปรแกรมที่พัฒนาโดย Guido Van Rossum โปรแกรมเมอร์ชาวเนเทอร์แลนด์โดยมีการปล่อยเวอร์ชันแรกในปี 1991 ภาษาไพทอนมีการใช้งานในหลากหลายด้าน เช่น การพัฒนาเว็บฝั่งเซิร์ฟเวอร์ที่มีเฟรมเวิร์กต่างๆรองรับ เช่น Django ที่ใช้ในการพัฒนาเว็บแอปพลิเคชันขนาดใหญ่ทั้งฝั่งหน้าบ้านและหลังบ้าน Flask เฟรมเวิร์กสำหรับทำเว็บเซิร์ฟเวอร์ขนาดเล็กมีชุมชนนักพัฒนาที่รองรับ นอกจากนี้ภาษาไพทอนยังสามารถใช้งานในด้านวิทยาศาสตร์ข้อมูล คอมพิวเตอร์วิทัศน์และการทำแบบจำลองการเรียนรู้ของเครื่อง (Machine learning) โดยมีไลบรารีต่างๆรองรับ เช่น Matplotlib Pandas OpenCV Numpy Keras เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 OpenCV [3]

OpenCV ย่อมาจาก Opensource Computer Vision เป็นไลบรารีโอเพ่นซอร์สที่เปิดให้นักพัฒนาทั่วไปสามารถใช้ในการพัฒนางานด้านคอมพิวเตอร์วิทัศน์โดยมีชุดเครื่องมือ 2มิติ และ 3มิติ ในการจัดการภาพและวิดีโอ OpenCV ยังสามารถประยุกต์ใช้งานกับงานที่หลากหลายเช่น การประมวลผลภาพ การจดจำใบหน้า การตรวจจับรูปทรงต่างๆ และ การใช้งานร่วมกับการเรียนรู้เชิงลึก เป็นต้น OpenCV สร้างขึ้นโดย Intel ด้วยภาษา C++ รองรับการใช้งานด้วยภาษา Python Java และ Matlab

รูปที่ 2.2 ไลบรารี OpenCV

2.1.3 pyzbar [4]

ไลบรารีไพทอนสำหรับถอดรหัสภาพ QR Code โดยข้อมูลที่ได้รับจากการถอดรหัสประกอบด้วย ส่วนแรก data หรือข้อมูลหรือข้อความที่บรรจุอยู่ภายใน QR Code ถัดมา type ชนิดของ code ที่อ่านเช่น QR Code Barcode เป็นต้น ถัดมา polygon ตำแหน่งระบุมุมทั้งสี่ของ QR Code ในระนาบพิกัด แกนนอน แกนดิ่ง และสุดท้าย rect ตำแหน่งระบุมุมทั้งสี่ของ QR Code ในระนาบพิกัด แกนนอน แกนดิ่งในรูปแบบ tuple

2.1.4 JavaScript Object Notation (JSON) [5-6]

JSON ย่อมาจาก JavaScript Object Notation คือรูปแบบข้อมูล text ที่ใช้สร้าง object ในการรับส่งข้อมูลระหว่างแอปพลิเคชันหรือการใช้ Applications programing interface (API) โดยรูปแบบจะอยู่ใน key-value ประเภทข้อมูลที่ JSON เก็บได้ประกอบด้วย string, number, object, array, Boolean และ null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 JavaScript Object Notation

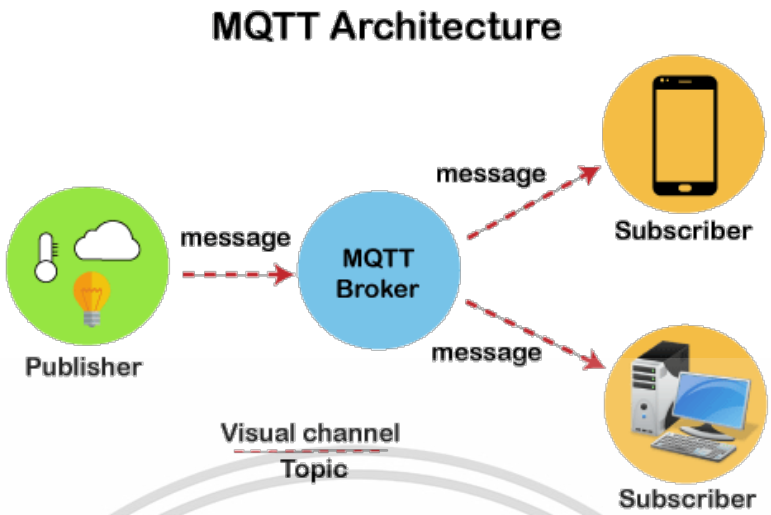
2.1.5 RESTful API [7]

API (Application Programming Interface) หรือส่วนประสานงานโปรแกรมประยุกต์ เป็นอินเทอร์เฟซของระบบคอมพิวเตอร์ที่ใช้สื่อสารรับส่งข้อมูลกับแอปพลิเคชันอื่นๆ โดยประกอบด้วยไคลเอ็นต์หรือผู้ใช้งานที่ต้องการเข้าถึงข้อมูลโดยไคลเอ็นต์จะเป็นบุคคลหรือระบบซอฟต์แวร์ และทรัพยากร คือข้อมูลที่แอปพลิเคชันส่งให้กับไคลเอ็นต์ เช่น วิดีโอ ข้อความ ตัวเลข เป็นต้น REST ย่อมาจาก Representational State Transfer เป็นสถาปัตยกรรมซอฟต์แวร์ที่กำหนดเงื่อนไขการทำงานให้แก่ API โดยเกณฑ์การส่งข้อมูลจะผ่านโพรโทคอล HTTP ประกอบด้วยวิธีดังต่อไปนี้ GET เพื่อเข้าถึงทรัพยากร POST เพื่อส่งทรัพยากรไปยังเซิร์ฟเวอร์ PUT เพื่อปรับปรุงทรัพยากรที่มีอยู่บนเซิร์ฟเวอร์ DELETE เพื่อลบทรัพยากรออกจากเซิร์ฟเวอร์โดยการตอบกลับจากเซิร์ฟเวอร์จะมีสถานะ HTTP เช่น 200 การตอบสนองว่าทำงานสำเร็จ 400 คำขอไม่ถูกต้องเซิร์ฟเวอร์ไม่สามารถประมวลผลได้ 404 ไม่พบทรัพยากร เป็นต้น

2.1.6 MQTT [8-9]

โพรโทคอล MQTT ย่อมาจาก Message Queue Telemetry Transport เป็นโพรโทคอลการส่งข้อความที่ใช้สำหรับสื่อสารระหว่างอุปกรณ์ เซ็นเซอร์อัจฉริยะ หรืออุปกรณ์อินเทอร์เน็ตในทุกสรรพสิ่ง (IoT) ซึ่งใช้แบนด์วิดท์จำกัดในการรับส่งข้อมูลสิ้นเปลืองพลังงานน้อยมีประสิทธิภาพสูง MQTT ประกอบด้วย ไคลเอนต์คืออุปกรณ์ตั้งแต่ตัวควบคุมไปจนถึงเซิร์ฟเวอร์ โดยจะมีหน้าที่เผยแพร่ข้อความ (publish message) หรือรับข้อความ (subscribe message) จาก topic ที่กำหนด โบรกเกอร์คือตัวกลางสำหรับประสานงานรับส่งข้อความกับฝั่งไคลเอนต์ระบุเส้นทาง topic เพื่อข้อความส่งถึงไคลเอนต์ที่กำหนด และการเชื่อมต่อ MQTT จะทำผ่าน TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 โพรโทคอล MQTT

2.1.7 TypeScript [10]

ภาษา TypeScript ถูกพัฒนามาจากภาษา JavaScript โดยสามารถทำงานบนสภาพแวดล้อมของ Node.js หรือ เว็บเบราว์เซอร์ที่มีการรองรับ ECMAScript 3 ขึ้นไป ภาษา TypeScript รองรับ Type annotation ประกาศตัวแปรระบุประเภทข้อมูลเช่น number string Compile-time type checking หรือการตรวจสอบประเภทข้อมูลระหว่างคอมไพล์เพื่อลดข้อมูลผิดประเภท Interfaces สำหรับกำหนดโครงสร้างให้กับคลาส

2.1.8 Nest JS [11]

Nest JS เป็นเฟรมเวิร์คสำหรับสร้างแอปพลิเคชันฝั่งเซิร์ฟเวอร์โดยใช้พื้นฐานของ Node.js โดยภาษาที่ใช้จะเป็น TypeScript และใช้แนวคิดของ OOP สามารถทำงานคู่กับไลบรารีอื่นๆ เช่น Express.js หรือ Socket.io เป็นต้น โครงสร้างของ Nest JS ประกอบไปด้วย Modules ที่ทำหน้าที่ครอบคลุมส่วนประกอบหลักต่างๆได้แก่ Controller และ Service Controller สำหรับจัดการรับ Request จากไคลเอ็นต์และทำหน้าที่ส่งข้อมูลไปให้ไคลเอ็นต์ผ่าน Route ต่างๆ Providers/Services ใช้สำหรับเขียนตรรกะทางโปรแกรมต่างๆ สามารถใช้ decorator @Injectable ในการดึงไปใช้งานได้ Middleware เมื่อมีrequestเข้ามาจะทำงานก่อนเข้าไปที่ route เช่นการทำ log ข้อมูล Pipe สำหรับการทำการตรวจสอบ Data transfer object (DTO) และ Guards ทำหน้าที่สำหรับตรวจสอบ Request ก่อนไปถึงroute สามารถระบุสิทธิการเข้าถึงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 เฟรมเวิร์ค Nest JS

2.1.9 Mongo DB [12]

ฐานข้อมูล MongoDB เป็นระบบจัดเก็บข้อมูลแบบไม่มี SQL ออกแบบมาเพื่อจัดเก็บข้อมูลขนาดใหญ่ที่ต้องการความยืดหยุ่นและปรับเปลี่ยนขนาดได้ โดยโมเดลข้อมูลของ MongoDB จะเก็บข้อมูลเอกสารคล้ายกับ JSON และสคีมาที่สามารถแก้ไขได้ MongoDB สามารถทำการแบ่งคลัสเตอร์ replication เพื่อการใช้งานที่ต่อเนื่องและมีประสิทธิภาพ

รูปที่ 2.6 ฐานข้อมูล MongoDB

2.1.10 AWS EC2 [13]

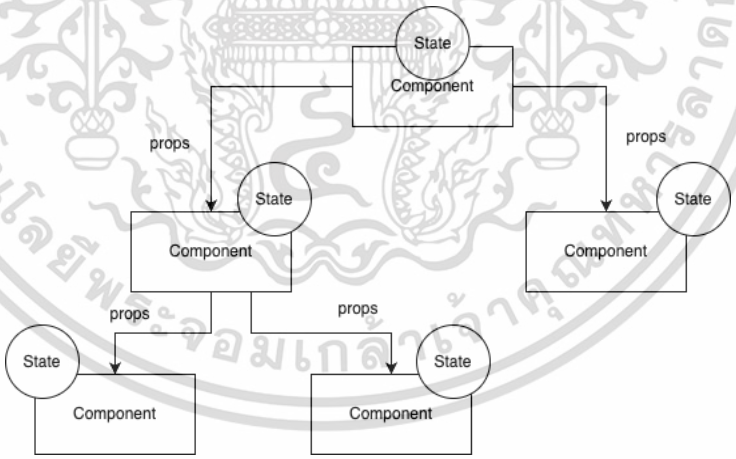
Amazon Elastic Compute Cloud (EC2) เป็นบริการของ Amazon Web Service ที่ให้ผู้ใช้งานสามารถสร้าง Virtual machine บนระบบคลาวด์ได้โดยสามารถปรับแต่ง spec ค่าต่างๆได้ตามความต้องการ เช่น ระบบปฏิบัติการ หน่วยความจำ หน่วยประมวลผลต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 Amazon Elastic Compute

2.1.11 React JS [14]

React เป็นไลบรารี JavaScript ที่พัฒนาโดย Facebook ใช้ในการพัฒนาส่วนของการติดต่อกับผู้ใช้ หรือ User interface โดยหลักการทำงานจะแสดงในรูปที่ 2.8 React จะประกอบด้วย Component หรือหน้าตาส่วนต่างๆในเว็บไซต์ โดยภายใน Component จะมีการเก็บข้อมูลเรียกว่า State จากนั้นเมื่อมีการส่งข้อมูลข้ามไปยัง Component ลูกจะส่งผ่าน Properties (Props)



รูปที่ 2.8 กระบวนการทำงานของ React JS

2.1.12 Socket.io [15]

Socket.io เป็นไลบรารี JavaScript ที่มีความหน่วงต่ำสามารถส่งข้อมูลได้แบบสองทิศทางระหว่างฝั่งไคลเอ็นต์และฝั่งเซิร์ฟเวอร์ โดยการรับส่งข้อมูลจะเป็นไปแบบเวลาจริง (Real-time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานบนพื้นฐานของ WebSocket โดยไลบรารี Socket.io นิยมนำมาใช้กับการสื่อสารที่ต้องการส่งแบบเวลาจริงเช่น ระบบแชท ระบบเกมที่มีผู้เล่นหลายคน ระบบข้อมูล Real-time เป็นต้น

รูปที่ 2.9 ไลบรารี Socket.io

2.1.13 Google Colab [16]

Google Colab เป็นบริการจาก Google ที่ให้ผู้พัฒนาสามารถใช้โปรแกรมเสมือน Jupyter Notebook บนคลาวด์ในการทำงานเกี่ยวกับโปรแกรมไพทอน โดย Google Colab จะติดตั้งสภาพแวดล้อมการทำงานของโปรแกรมมาให้แบบสำเร็จรูป ส่วนมากผู้ใช้งาน Google Colab จะใช้เกี่ยวกับงานพัฒนาวิทยาศาสตร์ข้อมูลและการเรียนรู้ของเครื่องโดยสามารถทำงานร่วมกับไลบรารีต่างๆได้ เช่น pandas, NumPy, matplotlib, scikit-learn เป็นต้น

รูปที่ 2.10 Google Colab

2.1.14 Pandas [17]

Pandas เป็นไลบรารีในภาษาไพทอนที่ใช้ในการจัดการข้อมูลเช่น การนำเข้าข้อมูลจากไฟล์ CSV หรือ Excel จากนั้นแปลงข้อมูลให้อยู่ในรูปแบบตาราง DataFrame แบ่งเป็นข้อมูลแถวและคอลัมน์ pandas สามารถใช้ในการทำความสะอาดข้อมูลได้ เช่น หาค่าข้อมูลสูญหาย (Missing

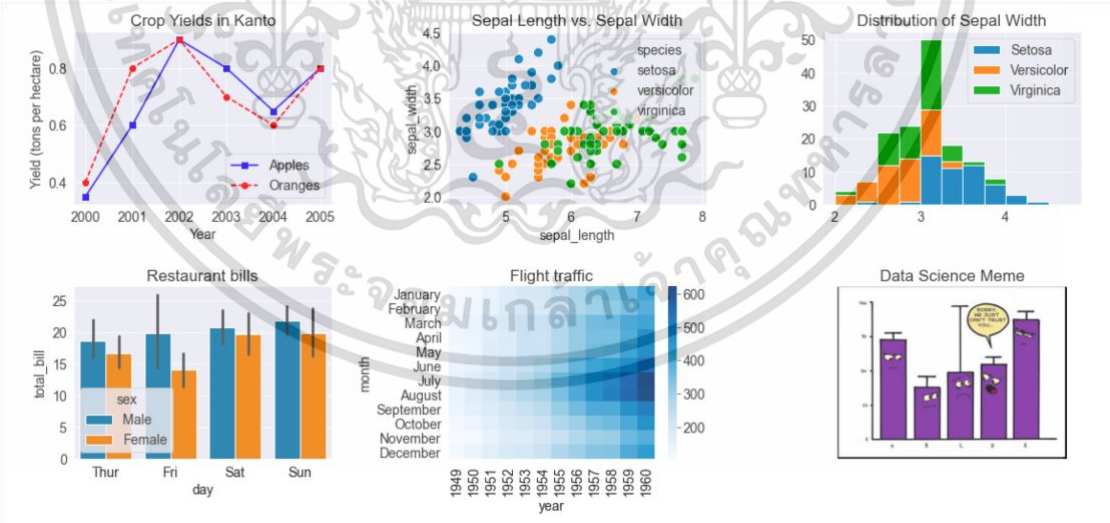
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Value) และทำการลบหรือแทนค่าเหล่านั้นด้วยค่าเฉลี่ย นอกจากนี้ pandas ยังสามารถแปลงรูปแบบประเภทข้อมูลที่ต่างกันให้อยู่ในประเภทเดียวกันได้ pandas สามารถหาค่าความสัมพันธ์ของข้อมูล Correlation ได้

รูปที่ 2.11 ไลบรารี pandas

2.1.15 Matplotlib [18-19]

Matplotlib เป็นไลบรารีสำหรับสร้างกราฟรูปแบบต่างๆได้ เช่น กราฟเส้น กราฟแผนภูมิแท่ง กราฟข้อมูลกระจัดกระจาย กราฟวงกลมและอื่นๆ ดังตัวอย่างกราฟแสดงในรูปที่ 2.12 โดยข้อมูลที่นำเข้ามาจะเป็น List หรือ NumPy Array

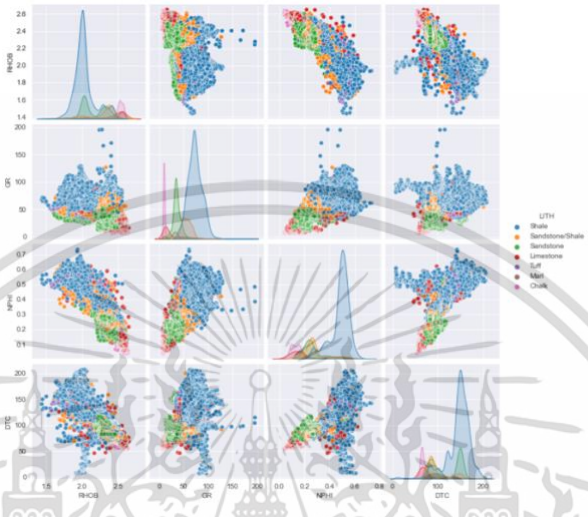


รูปที่ 2.12 ตัวอย่างกราฟของ Matplotlib

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.16 Seaborn [20-21]

Seaborn เป็นไลบรารีสำหรับกราฟรูปแบบต่างๆ เช่น กราฟแผนภูมิแท่ง กราฟ Countplot กราฟ Distribution plot กราฟ Heatmap กราฟ Pairplot เป็นต้นดังแสดงในตัวอย่างรูปที่ 2.13 โดยพัฒนามาบนพื้นฐานของ Matplotlib



รูปที่ 2.13 ตัวอย่างกราฟของ Seaborn

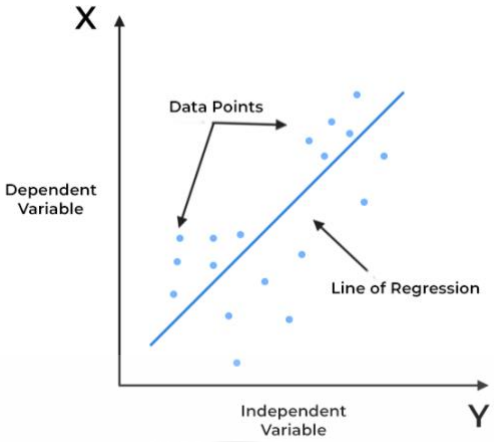
2.1.17 Scikit-learn [22]

Scikit-learn เป็นไลบรารีของภาษาไพทอนสำหรับทำงานด้านวิทยาศาสตร์ข้อมูล เช่นการทำแบบจำลองการเรียนรู้ของเครื่อง ตัวอย่างการทำงานได้แก่ Classification ใช้สำหรับการแยกข้อมูล เช่นการแยกอีเมลว่าเป็นแสปมหรือไม่ Regression ใช้สำหรับการหาความสัมพันธ์ระหว่างข้อมูล เช่น การทำนายราคาหุ้น Clustering การแบ่งกลุ่มข้อมูลเช่น การแบ่งกลุ่มลูกค้า และ Evaluation การวิเคราะห์ประสิทธิภาพความแม่นยำของแบบจำลองการเรียนรู้ของครื่อง

2.2 ทฤษฎีการวิเคราะห์การถดถอยเชิงเส้น [23-25]

การวิเคราะห์การถดถอยเชิงเส้น (Linear Regression) เป็นวิธีการทางสถิติอย่างหนึ่งในการหาความสัมพันธ์ของข้อมูลหรือตัวแปรที่ประกอบไปด้วยตัวแปรต้น (Independent variable) และตัวแปรตาม (Dependent variable) โดยความสัมพันธ์จะออกมาในรูปแบบของการเรียงกันเป็นเส้นตรงเพื่อแสดงผลลัพธ์ข้อมูลการทำนายค่าในอนาคต การวิเคราะห์การถดถอยเชิงเส้นเป็นการเรียนรู้แบบมีผู้สอน (Supervised learning) ที่ใช้ความสัมพันธ์ทางคณิตศาสตร์ระหว่างตัวแปรต่างๆ ในการหาความต่อเนื่องเชิงตัวเลข โดยตัวอย่างการนำไปใช้มีหลากหลาย เช่น การทำนายเงินเดือน การทำนายยอดขาย การทำนายราคาสินค้า ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ตัวอย่างความสัมพันธ์กราฟเส้นตรงของการวิเคราะห์การถดถอยเชิงเส้น

จากรูปที่ 2.14 จะสังเกตได้ว่าแนวแกนนอนจะเป็นค่าของตัวแปรต้น (Independent variable) และแนวตั้งจะเป็นค่าของตัวแปรตาม (Dependent variable) โดยเป้าหมายของและจุดๆ ที่แสดงในภาพจะเป็นจุดของข้อมูล (Data points) เป้าหมายของการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression) นั้นคือการหาเส้นตรง (Line of Regression) เพื่อใช้ทำนายให้ค่าใกล้เคียงกับจุดข้อมูลมากที่สุด การวิเคราะห์การถดถอยเชิงเส้น (Linear Regression) แบ่งได้เป็น 2 ประเภทตามจำนวนตัวแปรอิสระที่จะมากำหนดค่าตัวแปรตามได้แก่

1.) การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression) คือการวิเคราะห์การถดถอยของตัวแปรอิสระ 1 ค่าและตัวแปรตาม 1 ค่าโดยตัวแปรทั้งสองค่าจะมีความสัมพันธ์เชิงเส้นตรงตามกันหรือผกผันกันได้ ตัวอย่างสมการทางคณิตศาสตร์ดังสมการที่ (2.1)

$$Y_i = \beta_0 + \beta_1 X_i \tag{2.1}$$

เมื่อ Y_i คือ ค่าตัวแปรตามลำดับที่ i

β_0 คือ ค่าจุดตัดแกน y

β_1 คือ ค่าความชัน

X_i คือ ค่าคงที่ตัวแปรอิสระในลำดับที่ i

สำหรับตัวอย่างการใช้งานเช่น กำหนดยอดขายสินค้าคือตัวแปร Y และจำนวนประชากรในพื้นที่ด้วยตัวแปร X หากนำข้อมูลทั้ง 2 ค่ามาเขียนสมการเส้นตรงจะตีความได้ว่าปัจจัยของจำนวนประชากรในพื้นที่จะส่งกระทบกับยอดขายสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) การวิเคราะห์การถดถอยเชิงเส้นพหุคูณ (Multiple Linear Regression) คือการวิเคราะห์การถดถอยของตัวแปรอิสระมากกว่า 1 ค่าเพื่อกำหนดค่าตัวแปรตาม ซึ่งหมายความว่ามีหลายปัจจัยที่ส่งผลกระทบต่อตัวแปรตามดังสมการที่ (2.2)

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i \quad (2.2)$$

เมื่อ y คือ ค่าตัวแปรตาม

β_0 คือ ค่าจุดตัดแกน y

β_i คือ ค่าสัมประสิทธิของค่าประมาณการตัวนั้นๆ

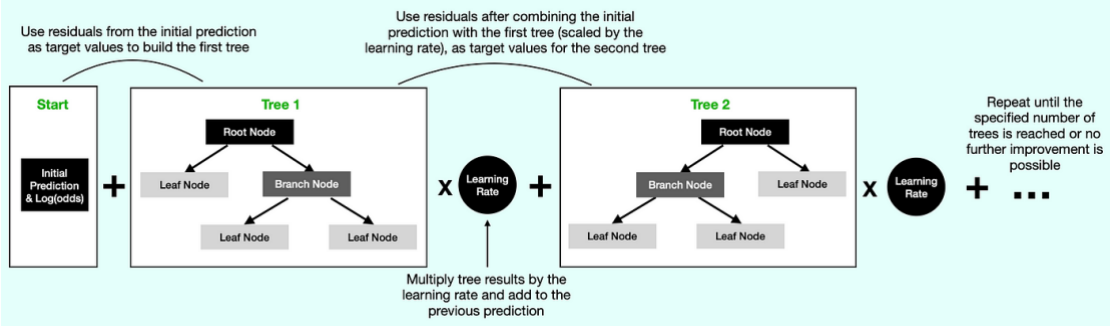
X_i คือ ค่าคงที่ตัวแปรอิสระ

ตัวอย่างการใช้งานเช่น กำหนดให้ตัวแปร Y คืออัตราบริโภคเชื้อเพลิงของรถยนต์ซึ่งปัจจัยที่ส่งผลกระทบต่อความสิ้นเปลืองมีได้หลายค่า เช่น อัตราเร็วของรถยนต์ แรงเสียดทานบนพื้นถนน ความชันบนพื้นถนน เป็นต้นซึ่งปัจจัยต่างๆเหล่านี้จะเป็นตัวแปรอิสระที่จะกำหนดว่าแต่ละปัจจัยจะส่งผลต่ออัตราสิ้นเปลืองเชื้อเพลิงมากน้อยเพียงใด

2.3 ทฤษฎี Extreme Gradient Boosting Regression (XGBoost) [26-27]

Extreme Gradient Boosting เป็นหนึ่งในรูปแบบการเรียนรู้ของเครื่องที่เป็นการเรียนรู้แบบมีผู้สอน (Supervised learning) โดยเป็นวิธีการที่พัฒนามาจาก Gradient Boosting ซึ่งนำต้นไม้ตัดสินใจมาฝึกสอนต่อกันหลาย ๆ ต้น โดยต้นไม้ตัดสินใจแต่ละต้นจะเรียนรู้ค่าความผิดพลาดของต้นไม้ก่อนหน้า ส่งผลให้ความแม่นยำในการทำนายจะมีค่าสูงขึ้นเรื่อย ๆ และเมื่อการเรียนรู้ของต้นไม้ตัดสินใจมีความลึกมาเพียงพอ แบบจำลองจะหยุดเรียนรู้เมื่อไม่มีค่าความผิดพลาดคงเหลือกจากต้นไม้ตัดสินใจให้เรียนรู้อีกต่อไป การใช้งาน Extreme Gradient Boosting สามารถประยุกต์ใช้ได้กับงานจำแนกประเภทหรือ (classification) และ งานด้านปัญหาการถดถอย (Regression)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ตัวอย่างกระบวนการทำงานของ Extreme Gradient Boosting

จากรูปที่ 2.15 จะสังเกตได้ว่าค่าความคลาดเคลื่อน (residual) ของการเรียนรู้ของต้นไม้ตัดสินใจในจังหวะเริ่มต้นจะใช้เป็นค่าเป้าหมายในการสร้างการเรียนรู้ของต้นไม้ตัดสินใจต้นที่หนึ่ง จากนั้นนำผลลัพธ์ค่าความคลาดเคลื่อน (residual) จากต้นไม้ตัดสินใจต้นที่หนึ่งไปคูณกับค่าอัตราการเรียนรู้ (Learning rate) เป็นค่าเป้าหมายของต้นไม้ต้นที่สอง จากนั้นนำค่าต้นไม้ต้นที่สองมารวมกับค่าต้นไม้ต้นที่หนึ่งโดยทำซ้ำไปเรื่อย ๆ จนได้ต้นไม้ที่มีค่าความคลาดเคลื่อนน้อยที่สุดจึงจะเป็นผลลัพธ์การทำนายสุดท้าย โดยหลักการนี้จะเป็นวิธีการ Boosting ของ Ensemble learning เพื่อสร้างตัวเรียนรู้หลายๆตัวให้การเรียนรู้มีประสิทธิภาพมากที่สุด

$$y_i = \beta_0 + \sum(\beta_j \times f_j(x_i)) \tag{2.3}$$

เมื่อ y_i คือ ค่าเป้าหมายที่ทำนายได้ในของลำดับที่ i

β_0 คือ ค่าจุดตัดแกน y

β_j คือ ค่านำหนักที่เกี่ยวข้องกับการเรียนรู้ต้นไม้ลำดับที่ j

$f_j(x_i)$ คือ การทำนายแผนภาพต้นไม้ของลำดับที่ j ด้วยข้อมูลตัวอย่างลำดับที่ i

จากสมการที่ (2.3) แสดงวิธีการทำงานของ Extreme Gradient Boosting Regression โดยการนำค่าของการทำนายต้นไม้ตัดสินใจแต่ละต้นมารวมกันจนได้เป็นค่าเป้าหมายที่ต้องการ

$$target = \sum[Loss(y_i, \hat{y}_i)] + \sum[\Omega(f_j)] \tag{2.4}$$

เมื่อ y_i คือ ค่าเป้าหมาย

$\hat{y}_i$ คือ ค่าทำนายจากแบบจำลอง

$\Omega(f_j)$ คือ ค่าความซับซ้อนของต้นไม้ตัดสินใจ

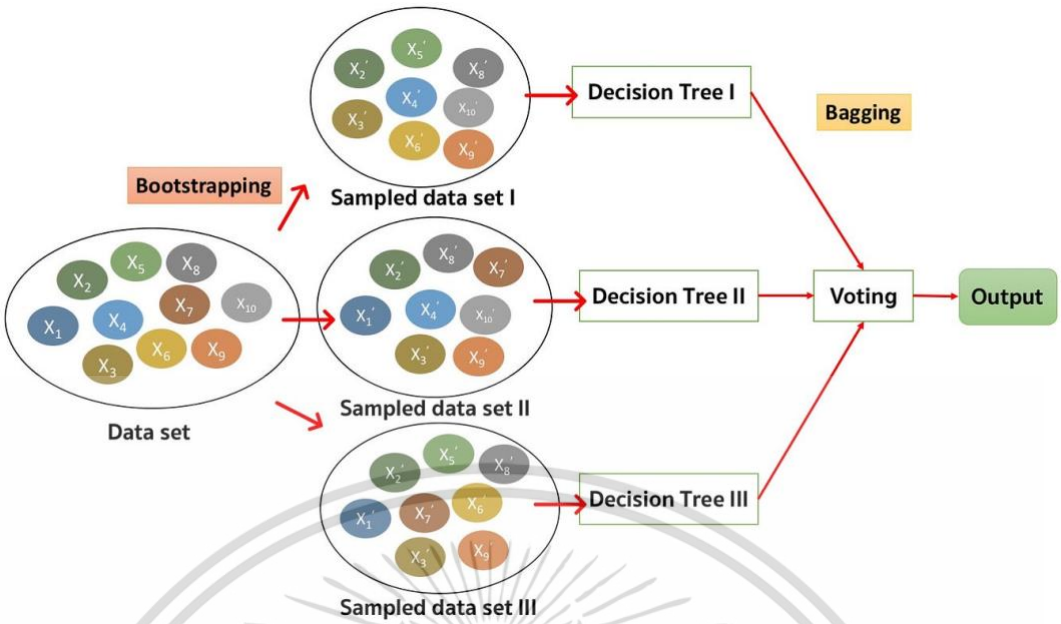
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (2.4) แสดงถึงเป้าหมาย (target) ที่ต้องการปรับปรุงค่าความคลาดเคลื่อนให้มีค่าน้อยลงและควบคุมความซับซ้อนของแบบจำลองเพื่อป้องกันปัญหา overfitting โดย $\sum[Loss(y_i, \hat{y}_i)]$ หมายถึงผลรวมของค่าความคลาดเคลื่อนระหว่างค่าเป้าหมายกับค่าที่ทำนายได้โดยส่วนนี้จะทำการปรับแบบจำลองให้มีความแม่นยำมากขึ้น และ $\sum[\Omega(f_j)]$ ส่วนนี้คือการรวมค่าความซับซ้อนของต้นไม้ตัดสินใจ (Regularization) เพื่อปรับปรุงแบบจำลองไม่ให้เกิด overfitting โดยผลรวมของทั้งสองค่าจะใช้เป็นตัวบ่งชี้ในการฝึกและปรับปรุงแบบจำลองให้มีประสิทธิภาพมากขึ้น

2.4 ทฤษฎี Random Forest Regression [28-29]

Random Forest เป็นหนึ่งในรูปแบบการเรียนรู้ของเครื่องที่เป็นการเรียนรู้แบบมีผู้สอน (Supervised learning) โดยเป็นวิธีที่พัฒนามาจากการใช้ต้นไม้ตัดสินใจ (Decision Tree) สามารถใช้ได้กับงานด้านการจำแนกประเภทหรือ (classification) และงานด้านปัญหาการถดถอย (Regression) จากรูปที่ 2.16 แสดงการทำงานของ Random Forest โดยจะสร้างแบบจำลองทำนายรูปแบบต้นไม้ตัดสินใจหลายๆแบบจำลองได้เป็นแบบจำลองต้นไม้ย่อย ๆ หลาย ๆ แบบจำลองโดยแต่ละแบบจำลองต้นไม้ตัดสินใจจะใช้ข้อมูลจากการสุ่มข้อมูลจากชุดข้อมูลเดิมได้เป็นชุดข้อมูลย่อย โดยแต่ละชุดข้อมูลย่อยที่นำไปใช้ในแบบจำลองต้นไม้ตัดสินใจจะไม่ซ้ำกัน การเรียนรู้ของต้นไม้ตัดสินใจแต่ละต้นจึงทำแบบอิสระ จากนั้นการทำนายผล กรณีแบบจำลองการแยกประเภท (classification) จะใช้การ vote ผลลัพธ์ที่ได้จากการทำนายของแบบจำลองต้นไม้ตัดสินใจแต่ละต้น หากแบบจำลองใดได้ผล vote มากที่สุดจะเป็นผลลัพธ์คำตอบของแบบจำลอง และกรณีการถดถอย (Regression) จะใช้ค่าเฉลี่ยของผลลัพธ์แต่ละแบบจำลองต้นไม้ตัดสินใจเป็นผลลัพธ์การทำนาย ซึ่งการทำงานของ Random Forest เป็นวิธีการของ Bagging หรือ Bootstrap Aggregation ของ Ensemble learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 ตัวอย่างกระบวนการทำงานของ Random Forest

จากสมการที่ (2.5) จะสังเกตได้ว่ารูปแบบสมการจะคล้ายกับสมการของ Extreme Gradient Boosting แต่หลักการทำงานภายในจะแตกต่างกัน โดยหลักการของ Random Forest จะใช้การ Bagging ที่สุ่มข้อมูลและแบ่งแบบจำลองต้นไม้ตัดสินใจหลายๆแบบ และนำผลลัพธ์การทำนายมารวมกันเพื่อหาค่าเฉลี่ยเป็นผลลัพธ์การทำนายของแบบจำลองแทน

$$y_i = \beta_0 + \sum(\beta_j \times f_j(x_i)) \tag{2.5}$$

เมื่อ y_i คือ ค่าเป้าหมายที่ทำนายได้ในของลำดับที่ i

β_0 คือ ค่าจุดตัดแกน y

β_j คือ ค่านำหนักที่เกี่ยวข้องกับการเรียนรู้ต้นไม้ลำดับที่ j

$f_j(x_i)$ คือ การทำนายแผนภาพต้นไม้ของลำดับที่ j ด้วยข้อมูลตัวอย่างลำดับที่ i

2.5 การวัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง [30-31]

การวัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องที่ใช้ในงานด้านการหาค่าถดถอย (Regression) จะใช้ค่าความคลาดเคลื่อน (Loss Function) ในการหาว่าค่าที่ทำนายได้จากแบบจำลองเปรียบเทียบกับค่าข้อมูลจริง หากได้ผลลัพธ์ที่มีค่าน้อยแสดงว่าแบบจำลองสามารถทำนาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลได้มีค่าใกล้เคียงกับความเป็นจริงมาก แบบจึงลองจึงมีความแม่นยำสูง โดยค่าที่ใช้ว่าความคลาดเคลื่อนในงานวิจัยนี้ประกอบไปด้วย 2 ค่าได้แก่

1.) ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) หรือ MAE คือการหาค่าเฉลี่ยของความแตกต่างสัมบูรณ์ระหว่างค่าที่ทำนายจากแบบจำลองและค่าข้อมูลจริง โดยหากค่าค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ยมีค่าน้อยหมายความว่าแบบจำลองทำนายผลได้ใกล้เคียงกับความเป็นจริง ดังแสดงในสมการที่ (2.6)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.6}$$

เมื่อ y_i คือ ค่าข้อมูลจริงที่ลำดับ i

$\hat{y}_i$ คือ ค่าผลการทำนายที่ลำดับ i

n คือ จำนวนตัวอย่างข้อมูลทั้งหมด

2.) ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) หรือ RMSE คือการหาค่าความแตกต่างระหว่างค่าข้อมูลจริงและค่าที่ได้จากการทำนายจากนั้นนำมาหาค่ารากที่สอง โดยหากแบบจำลองมีผลลัพธ์ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสองน้อยหมายความว่าแบบจำลองทำนายผลได้ใกล้เคียงกับความเป็นจริง ดังแสดงในสมการที่ (2.7)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.7}$$

เมื่อ y_i คือ ค่าข้อมูลจริงที่ลำดับ i

$\hat{y}_i$ คือ ค่าผลการทำนายที่ลำดับ i

n คือ จำนวนตัวอย่างข้อมูลทั้งหมด

3.) ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) หรือ MAPE คือการคำนวนหาเปอร์เซ็นต์ของค่าความผิดผลาดที่ได้จากค่าเฉลี่ยของผลรวมค่าความแตกต่างระหว่างค่าที่ได้จากการทำนายและค่าข้อมูลจริง โดยหากแบบจำลองมีผลลัพธ์ค่าเฉลี่ยร้อยละความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผิดพลาดสัมบูรณ์น้อยหมายความว่าแบบจำลองทำนายผลได้ใกล้เคียงกับความเป็นจริง ดังแสดงในสมการที่ (2.8)

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (2.8)$$

เมื่อ A_t คือ ค่าข้อมูลจริงที่ลำดับ t

F_t คือ ค่าผลการทำนายที่ลำดับ t

n คือ จำนวนตัวอย่างข้อมูลทั้งหมด

2.6 ทบทวนวรรณกรรม

วรรณกรรมของการจัดการคลังสินค้าด้วยอินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่องจะแบ่งการศึกษาได้เป็นสองประเภท ได้แก่ การศึกษาการใช้อินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำงานในอุตสาหกรรม และ การศึกษาการใช้แบบจำลองการเรียนรู้ของเครื่องในการทำนายผลปริมาณสินค้า

การศึกษาการใช้อินเทอร์เน็ตในทุกสรรพสิ่ง และ Microservices ร่วมกับการทำงานในอุตสาหกรรม งานวิจัยของ Mohamed Dhouioui และคณะ [32] เสนอระบบจัดการคลังสินค้าอัจฉริยะโดยการออกแบบกระบวนการทำงาน ให้ง่ายและมีประสิทธิภาพมากขึ้นลดเวลาการขนส่งให้รวดเร็วยิ่งขึ้น ระบบประกอบด้วยอินเทอร์เน็ตในทุกสรรพสิ่งและการประมวลผลบนคลาวด์ เว็บแอปพลิเคชันสำหรับผู้ใช้งาน และ หุ่นยนต์ส่งสินค้า Fucheng Men และคณะ [33] พัฒนาระบบการขนส่งภายในคลังสินค้าโดยใช้อินเทอร์เน็ตในทุกสรรพสิ่งร่วมกับการจัดการเส้นทางด้วยวิธีอาณานิคมมด (Ant colony algorithm) เพื่อติดตามตำแหน่งรถจัดการเส้นทางและควบคุมความเร็วรถ Dimitris Gkoulis และคณะ [34] เสนอแพลตฟอร์มที่ใช้การออกแบบด้วยสถาปัตยกรรม Microservices ร่วมกับการสื่อสารแบบ event-based เพื่อเพิ่มประสิทธิภาพของระบบโดยประยุกต์ใช้งานกับเกษตรอัจฉริยะประกอบด้วยอุปกรณ์วัดความชื้นที่ทำงานโดยอัตโนมัติผ่านการสื่อสารอินเทอร์เน็ตในทุกสรรพสิ่งเพิ่มประสิทธิภาพในการทำงาน

อีกกลุ่มการศึกษาการใช้แบบจำลองการเรียนรู้ของเครื่องในการทำนายผลปริมาณสินค้า Tanisha G. Patil และคณะ [35] เสนอแบบจำลองการเรียนรู้ของเครื่องในการทำนายยอดขายสินค้าตามกลุ่มลูกค้าในวัน Black Friday โดยแบบจำลองประกอบไปด้วยการแยกกลุ่มลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(classification) และการถดถอย (Regression) สำหรับทนายยอดขายสินค้า โดยผลลัพธ์แบบจำลองสามารถเพิ่มประสิทธิภาพในการจัดเตรียมวางแผนสินค้าก่อนวัน Black Friday ได้ดียิ่งขึ้น K. L. Keung และคณะ [36] เสนอเทคโนโลยีในยุค 4.0 ที่ช่วยเพิ่มความพึงพอใจและความจงรักภักดีต่อแบรนด์สินค้ารวมถึงปรับปรุงประสิทธิภาพการทำงาน โดยงานวิจัยกล่าวถึงปัญหาการขนส่งที่ช้าและการทำนายยอดขายสินค้าจากการทำความเข้าใจกับคุณสมบัติต่าง ๆ ที่ส่งผลต่อผลลัพธ์การทำนาย โดยผลลัพธ์พบว่าตำแหน่งที่ตั้งและประเภทสินค้าจะมีผลต่อยอดขายและการขนส่งที่ล่าช้าได้ Puneet และคณะ [37] เสนอระบบจัดการคลังสินค้าอัจฉริยะด้วยแบบจำลองการเรียนรู้ของเครื่อง โดยการใช้แบบจำลองการเรียนรู้ของเครื่องในการทำนายอุปสงค์ของสินค้าล่วงหน้าจะช่วยให้การจัดการคลังสินค้ามีประสิทธิภาพมากขึ้น สามารถผลิตสินค้าได้ตรงตามปริมาณความต้องการของผู้บริโภค ลดปัญหาสินค้าล้นคลังสินค้า หรือสินค้าไม่เพียงพอต่อการจำหน่าย ยกระดับประสบการณ์ที่ดีให้แก่ผู้บริโภค

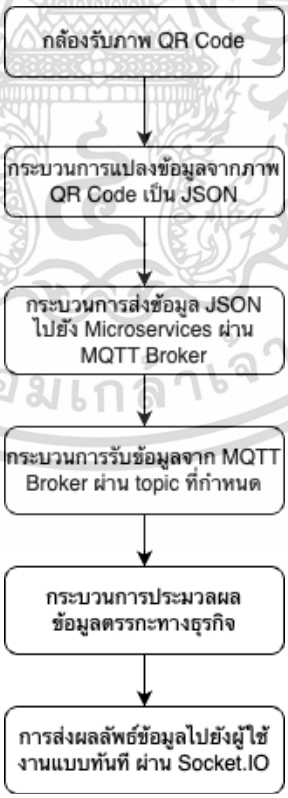
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินการทดลอง

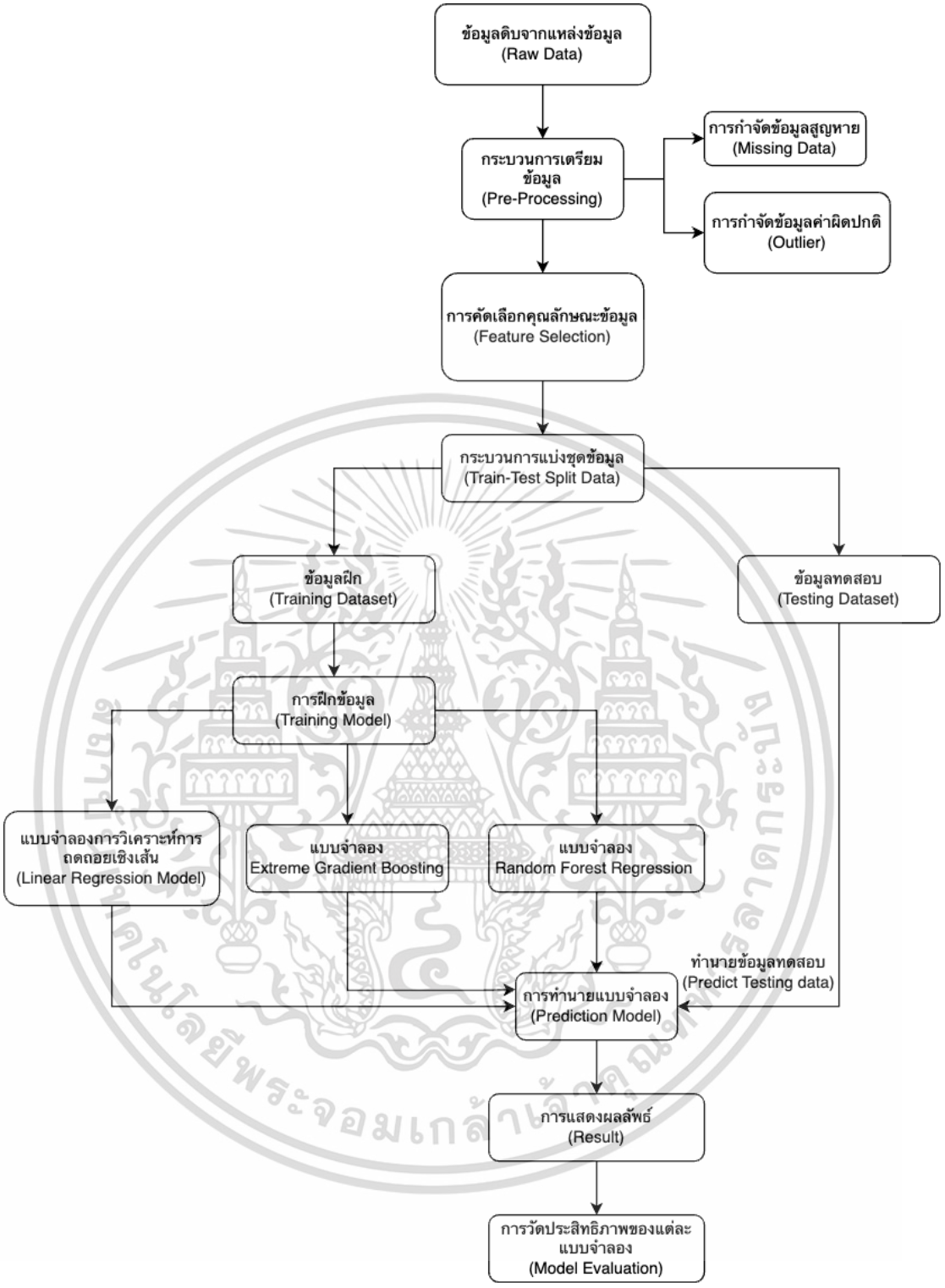
3.1 บทนำ

องค์ประกอบของงานวิจัยนี้ ประกอบไปด้วยการวิจัย 2 ส่วน ได้แก่ การติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าด้วยอินเทอร์เน็ตในทุกสิ่ง และ Microservices พร้อมทำการแสดงผลข้อมูลผ่านหน้าจอแบบเวลาจริง (Real-time monitoring) การทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง (Machine learning model) โดยการวิจัยลำดับแรกจะเริ่มจากรูปที่ 3.1 แสดงกระบวนการทำงานของการติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าโดยเริ่มจาก กล้องทำการรับภาพ QR Code เข้ามาจากนั้นตัวควบคุม (Controller) ส่งข้อมูลภาพไปแปลงรูปแบบข้อมูลเป็น JSON และส่งข้อมูลโดยใช้ MQTT Broker ไปที่ Microservices ผ่าน topic ที่กำหนดไว้เพื่อประมวลผลข้อมูลทางธุรกิจ (Business logic processing) จากนั้นส่งผลลัพธ์ข้อมูลผ่าน Socket.IO ไปแสดงผลที่เว็บแอปพลิเคชันแบบเวลาจริง (Real time web application monitoring)



รูปที่ 3.1 กระบวนการติดตามการทำงานรถโฟล์คลิฟท์ภายในคลังสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 กระบวนการทำงานของแบบจำลองการทำนายปริมาณสินค้าด้วยการเรียนรู้ของเครื่อง

กระบวนการทำงานส่วนที่ 2 ดังแสดงในรูปที่ 3.2 แสดงกระบวนการทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง (Machine learning model) โดยเริ่มจาก การรับข้อมูลดิบจากแหล่งข้อมูลและนำข้อมูลเข้ากระบวนการเตรียมข้อมูล (Pre-Processing) เพื่อเตรียมข้อมูลที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับนำไปใช้ทำแบบจำลองการเรียนรู้ของเครื่อง ซึ่งประกอบไปด้วย การกำจัดข้อมูลสูญหายหรือค่าว่าง (Missing Data) และ การกำจัดข้อมูลที่มีค่าผิดปกติ (Outlier) เช่น ค่าที่แตกต่างมากๆ ไปจากกลุ่มข้อมูลส่วนใหญ่ จากนั้นเข้าสู่การคัดเลือกคุณลักษณะข้อมูล (Feature Selection) ที่จะนำมาใช้สำหรับการทำแบบจำลองการเรียนรู้ของเครื่อง ต่อมานำข้อมูลที่ได้ไปเข้ากระบวนการแบ่งชุดข้อมูลเป็น ข้อมูลฝึก (Training Dataset) เพื่อนำไปใช้ฝึกแบบจำลองการเรียนรู้ของเครื่อง และ ข้อมูลทดสอบ (Testing Dataset) เพื่อทดสอบทำนายผลของแต่ละแบบจำลองการเรียนรู้ของเครื่อง

ในกระบวนการฝึกข้อมูล (Training model) จะประกอบไปด้วย 3 แบบจำลอง ได้แก่ แบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) แบบจำลอง Extreme Gradient Boosting (XGBoost) และ แบบจำลอง Random Forest Regression โดยหลังจากกระบวนการฝึกข้อมูลของทั้ง 3 แบบจำลองได้เสร็จสิ้นแล้วจะนำชุดข้อมูลทดสอบ (Testing Dataset) ไปทำนายของแต่ละแบบจำลอง (Model Prediction) เพื่อนำผลลัพธ์การทำนาย (Result) มาเปรียบเทียบกับข้อมูลจริงเพื่อวัดผลประสิทธิภาพความแม่นยำของแต่ละแบบจำลอง (Model evaluation)

3.2 การติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าด้วยอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) และ Microservices

การวิจัยติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าเพื่อติดตามการทำงานของพนักงานขับรถ และ ตำแหน่งการทำงานของรถโฟล์คลิฟท์นั้นประกอบไปด้วย 4 ลำดับ ได้แก่ การออกแบบอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) และ สถาปัตยกรรม Microservices การพัฒนาอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) ร่วมกับโพรโทคอล MQTT การพัฒนาสถาปัตยกรรม Microservices และกระบวนการทำงานกับโครงสร้างภายในคลังสินค้า

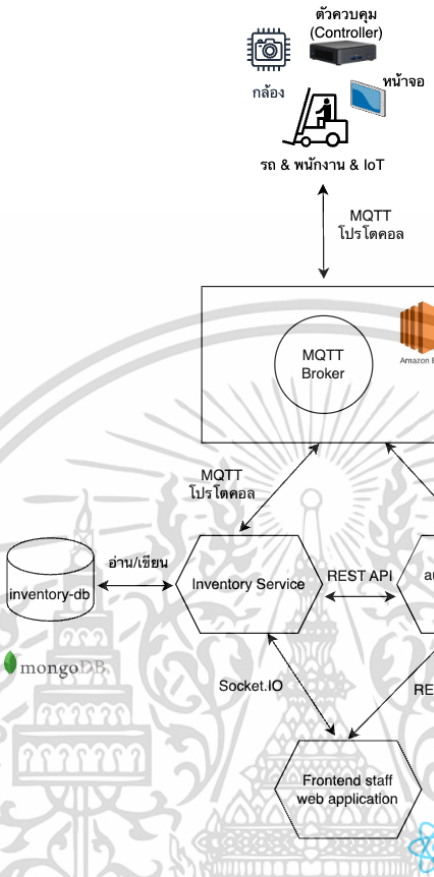
3.2.1 การออกแบบอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) และสถาปัตยกรรม Microservices

การออกแบบระบบติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าดังแสดงในรูปที่ 3.3 ประกอบไปด้วย 2 ส่วน ได้แก่

1.) การออกแบบอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) บนรถโฟล์คลิฟท์จะติดตั้งอุปกรณ์กล้อง web camera สำหรับรับภาพข้อมูล QR Code กล่องควบคุม (nux Intel Core i5 RAM 4 กิกะไบต์) สำหรับควบคุมประมวลผลและเชื่อมต่อกับอินเทอร์เน็ต หน้าจออินเตอร์เฟสสำหรับติดต่อแสดงผลกับผู้ใช้งาน โดยข้อมูลภาพ QR Code ที่ได้จากกล้อง จะแปลงเป็นรูปแบบ JSON และส่งไปยัง Topic ที่กำหนดไว้บน MQTT Broker ที่ติดตั้งอยู่บนคลาวด์ของ AWS EC2 ผ่าน MQTT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโทคอลเพื่อสถาปัตยกรรม Microservices จะนำข้อมูลไปประมวลผลทางตรรกะธุรกิจต่อไป (Business logic process)



รูปที่ 3.3 การออกแบบโครงสร้างอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) และ สถาปัตยกรรม Microservices

2.) การออกแบบสถาปัตยกรรม Microservices นั้น หากสังเกตในรูปที่ 3.3 เมื่อข้อมูลจากอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) บนรถโฟล์คลิฟท์นำส่งออก (publish message) มาที่ topic ที่กำหนดบน MQTT Broker แล้วฝั่งสถาปัตยกรรม Microservices จะทำการรับข้อมูลเข้ามา (subscribe message) โดย เซอร์วิสคลังสินค้า (Inventory Service) จะทำหน้าที่ประมวลผลข้อมูลที่เกี่ยวกับการจัดการของคลังสินค้ารวมถึงติดต่อกับเซอร์วิสการยืนยันตัวตน (Authentication Service) ที่ทำหน้าที่จัดการข้อมูลพนักงานและรถโฟล์คลิฟท์ ซึ่งแต่ละเซอร์วิสจะมีฐานข้อมูลจัดเก็บเป็นของตัวเอง นอกจากนี้การสื่อสารระหว่างเซอร์วิสจะใช้แบบ Representational state transfer (REST API) และสุดท้ายเซอร์วิสเว็บแอปพลิเคชันที่ใช้ติดต่อกับพนักงานขับรถโฟล์คลิฟท์ โดยจะสื่อสารแบบสองทิศทางรับส่งข้อมูลกับเซอร์วิสคลังสินค้าแบบเวลาจริงผ่านไลบรารี Socket.IO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การพัฒนา อินเทอร์เน็ตในทุกสิ่ง (Internet of Things) ร่วมกับโพรโทคอล MQTT

การพัฒนาอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) บนรถโฟล์คลิฟท์มีขั้นตอนการทำงานดังรูปที่ 3.4 โดยจะใช้ภาษาไพทอน3 (python 3) ในการพัฒนาโดยกล้องที่ติดอยู่หน้ารถโฟล์คลิฟท์ทำหน้าที่รับภาพ QR Code เข้ามาจากนั้นไลบรารีที่ใช้อ่านข้อมูลภาพ QR Code ที่ติดไว้ตามตำแหน่งต่างๆภายในคลังสินค้าคือ OpenCV ต่อมาไลบรารี pyzbar จะทำการถอดรหัสข้อมูลภาพที่รับเข้ามา โดยข้อมูลที่ถอดรหัสแล้วจะนำมาแปลงเป็นรูปแบบข้อมูล JSON โดยใช้แพ็กเกจ json ของไพทอน ได้ผลลัพธ์ดังตัวอย่างรูปที่ 3.5 ซึ่งอธิบายค่าต่างๆในตารางที่ 3.1 ถัดมานำข้อมูล JSON ส่งไปยัง MQTT Broker โดยใช้ไลบรารี paho-mqtt เพื่อส่งไปประมวลผลข้อมูลที่ Microservice ต่อไป

รูปที่ 3.4 ขั้นตอนการทำงานอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) บนรถโฟล์คลิฟท์

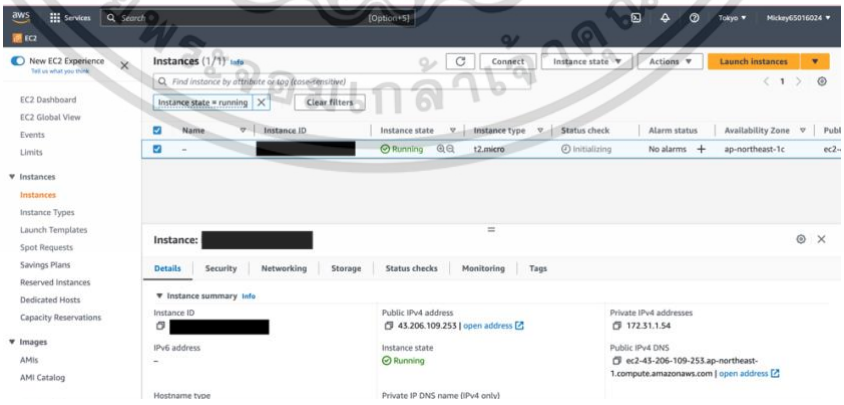
รูปที่ 3.5 ตัวอย่างข้อมูล JSON จากการแปลงภาพ QR Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 อธิบายค่าต่างๆของข้อมูลจาก QR Code รูปแบบ JSON

หัวข้อ	**คำอธิบาย**	**ตัวอย่างข้อมูล**	**รูปแบบข้อมูล**
event	การบ่งบอกเหตุการณ์ เช่น information สำหรับการแจ้งข้อมูลเมื่อรถวิ่งผ่านพื้นที่ปกติ และ alert สำหรับ แจ้งเตือนเมื่อรถวิ่งผ่านบริเวณห้ามผ่าน	Information, alert	String
zone	พื้นที่หลักในคลังสินค้า	A1	String
subZone	พื้นที่ย่อยในคลังสินค้า	02	String
vehicleId	รหัสประจำรถโฟล์คลิฟท์	64075c41ecf2f40e56cfd11d	String
timeStamp	เวลาที่รถวิ่งผ่าน QR Code	2023-05-22 15:16:31.446622	String

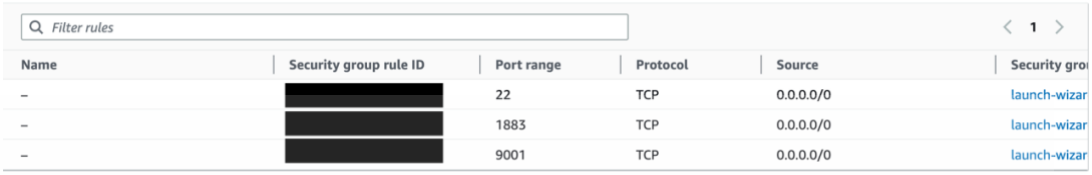
ถัดมาการพัฒนา MQTT Broker จะใช้ Eclipse Mosquitto ซึ่งเป็นเครื่องมือ open-source สำหรับสร้าง MQTT message broker บนพื้นฐานของโพรโทคอล MQTT โดยจะทำการติดตั้งไว้ที่บริการคลาวด์ EC2 ของ Amazon web service ซึ่งทำหน้าที่เป็น Virtual machine หรือเซิร์ฟเวอร์ที่มีระบบปฏิบัติการสามารถทำหน้าที่ประมวลผลต่างๆได้ โดยเริ่มต้นจากสร้าง EC2 Instance ประเภท t2.micro ระบบปฏิบัติการ Ubuntu เมื่อสร้างสำเร็จแล้วจะได้ผลลัพธ์ดังรูปที่ 3.6 มีสถานะ Instance State เป็น Running



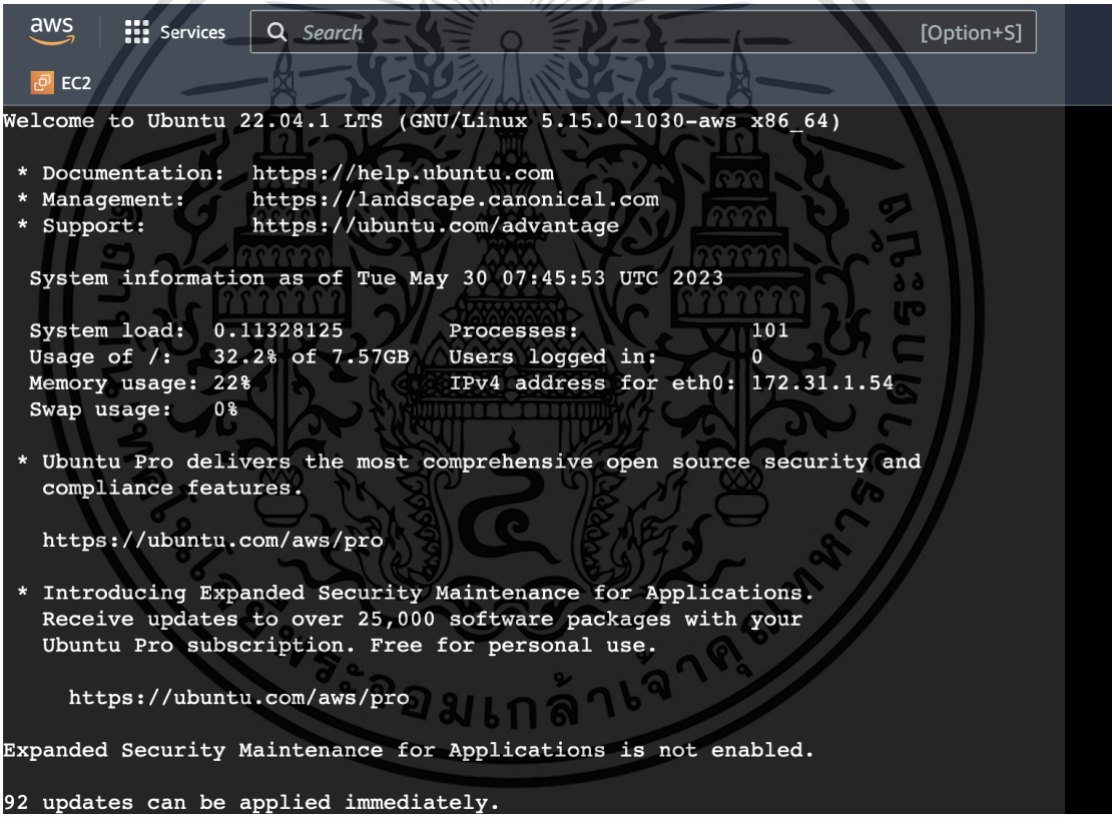
รูปที่ 3.6 การตั้งค่า EC2 Instance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาทำการตั้งค่า outbound rule ของ EC2 ให้รองรับการเชื่อมต่อจากภายนอก โดยตั้งค่าเป็น TCP Port 1883 และ traffic 0.0.0.0/0 โดย Port 1883 จะใช้เชื่อมต่อกับ Eclipse Mosquitto Broker ดังรูปที่ 3.7 จากนั้นทดลองเชื่อมต่อกับ Console EC2 โดยผลลัพธ์ที่ได้จะแสดงข้อมูลเวอร์ชันของระบบปฏิบัติการ Ubuntu ที่ได้ติดตั้งไว้ดังรูปที่ 3.8



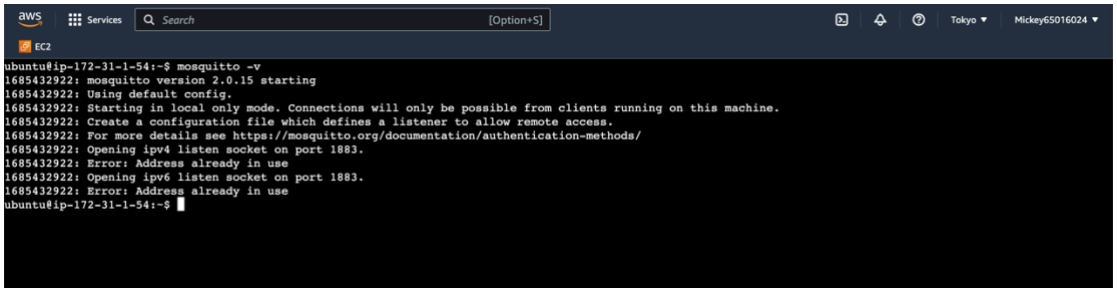
รูปที่ 3.7 การตั้งค่าการเชื่อมต่อกับ EC2 Instance



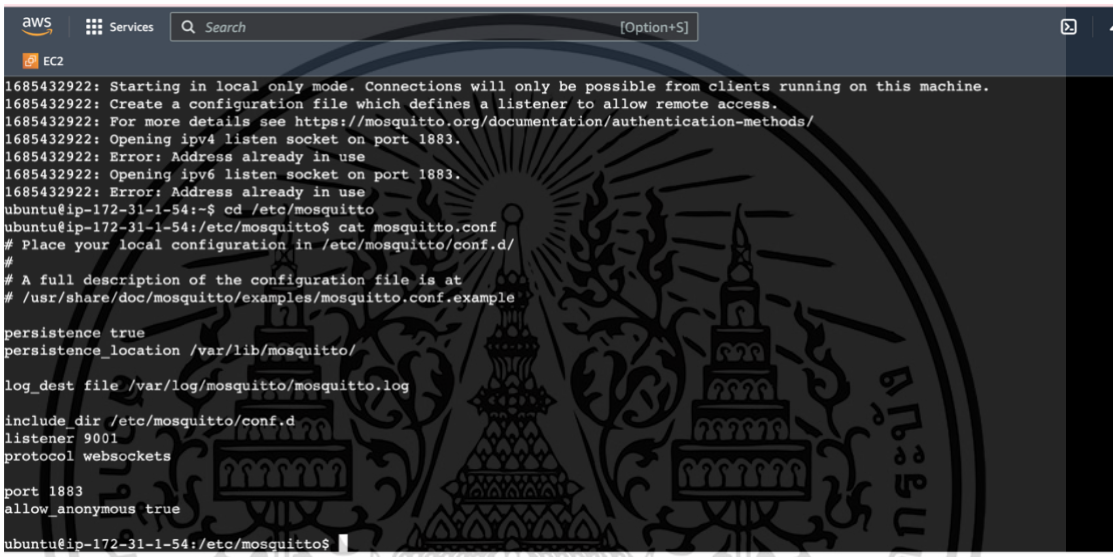
รูปที่ 3.8 การทดสอบเชื่อมต่อกับ Console EC2

จากนั้นทำการติดตั้ง Eclipse Mosquitto Broker ผ่าน Console EC2 โดยผลลัพธ์เมื่อติดตั้งสมบูรณ์จะสามารถตรวจสอบข้อมูลเวอร์ชันได้ดังแสดงในรูปที่ 3.9 ต่อมาทำการตั้งค่าในไฟล์ etc/mosquitto/mosquitto.conf ให้สามารถเชื่อมต่อได้ผ่านไอพีสาธารณะ Port 1883 และ allow_anonymous เป็น true ให้สามารถใช้งานโดยไม่ต้องลงทะเบียนผู้ใช้งานก่อนดังรูปที่ 3.10

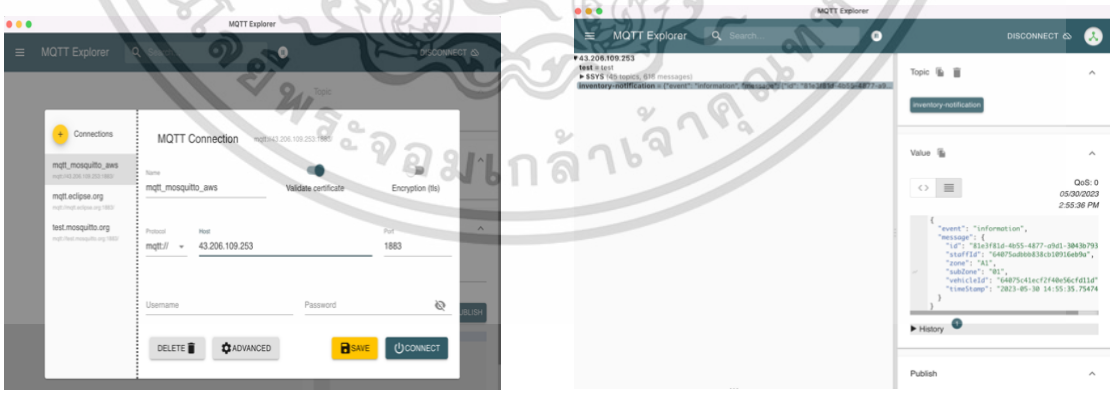
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ผลการติดตั้ง Eclipse Mosquitto Broker



รูปที่ 3.10 การตั้งค่า Eclipse Mosquitto Broker



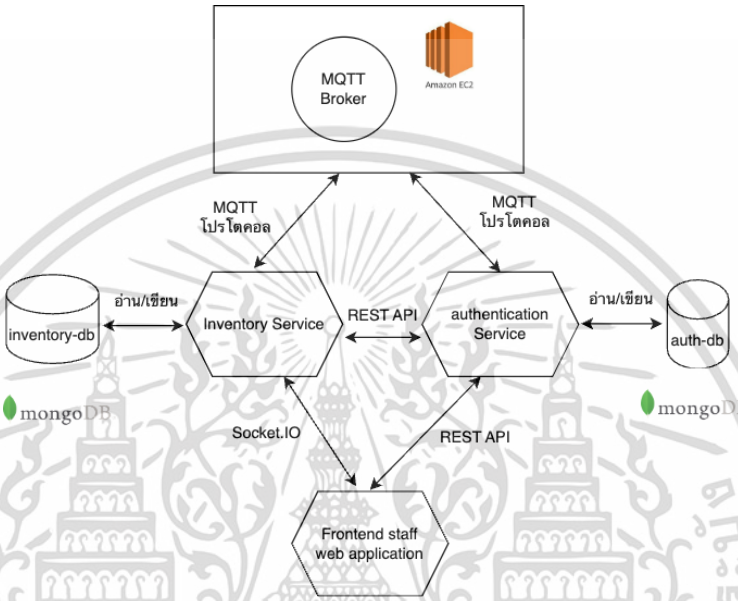
(ก) (ข)

รูปที่ 3.11 การทดสอบเชื่อมต่อกับ Eclipse Mosquitto Broker (ก) การเชื่อมต่อ Eclipse Mosquitto Broker บน AWS EC2 (ข) ผลการรับข้อมูลผ่าน topic ทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.11 (ก) ทำการทดสอบเชื่อมต่อกับ Eclipse Mosquitto Broker ที่ติดตั้งบน AWS EC2 ผ่านโปรแกรม MQTT Explorer โดยผลลัพธ์การเชื่อมต่อและทดลองส่งข้อมูลผ่าน topic ทดสอบดังแสดงผลในรูปที่ 3.11 (ข) จึงเป็นการเสร็จสิ้นการพัฒนาและติดตั้ง MQTT Broker บน EC2 Amazon web service

3.2.3 การพัฒนาสถาปัตยกรรม Microservices



รูปที่ 3.12 โครงสร้างสถาปัตยกรรม Microservices

สถาปัตยกรรม Microservices จะทำหน้าที่ประมวลผลรับส่งข้อมูลกับอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) บนรถโฟล์คลิฟท์ รวมถึงบันทึกข้อมูลลงฐานข้อมูลและส่งข้อมูลแสดงผลไปยังเว็บแอปพลิเคชัน โดยการพัฒนาเซอร์วิสคลังสินค้า (Inventory Service) และเซอร์วิสการยืนยันตัวตน (Authentication Service) จะพัฒนาด้วยเฟรมเวิร์ก NestJS ซึ่งทำงานบนพื้นฐานของภาษา TypeScript และ Node.js ในแต่ละเซอร์วิสจะเชื่อมต่อกับฐานข้อมูลของตัวเองโดยใช้ฐานข้อมูลของ Mongo DB ซึ่งเป็นฐานข้อมูลแบบ NoSQL เก็บข้อมูลเป็น JSON document ขณะที่เซอร์วิสเว็บ-แอปพลิเคชันจะใช้ React ซึ่งเป็นจาวาสคริปต์ไลบรารีสำหรับสร้างส่วนติดต่อกับผู้ใช้งาน (User-interface) ในการติดต่อกับเซอร์วิสอื่นจะใช้ REST API กับ Socket.IO โดยการพัฒนาสถาปัตยกรรม Microservices ประกอบไปด้วย 3 องค์ประกอบ

1.) เซอร์วิสคลังสินค้า (Inventory Service) มีหน้าที่จัดการประมวลผลข้อมูลภายในคลังสินค้า กำหนดว่าพื้นที่ไหนไม่สามารถผ่านได้หรือผ่านได้ จัดเก็บข้อมูลเวลาการทำงานของรถโฟล์คลิฟท์และตำแหน่งของรถที่เคลื่อนที่ผ่าน เซอร์วิสคลังสินค้าประกอบด้วย API สำหรับสื่อสารกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซอร์วิสการยืนยันตัวตน ดังแสดงในตารางที่ 3.2 และ โครงสร้างข้อมูล JSON สำหรับสื่อสารผ่าน Socket.IO กับเซอร์วิสเว็บแอปพลิเคชัน ดังแสดงในตารางที่ 3.3

ตารางที่ 3.2 องค์ประกอบ API ของเซอร์วิสคลังสินค้า

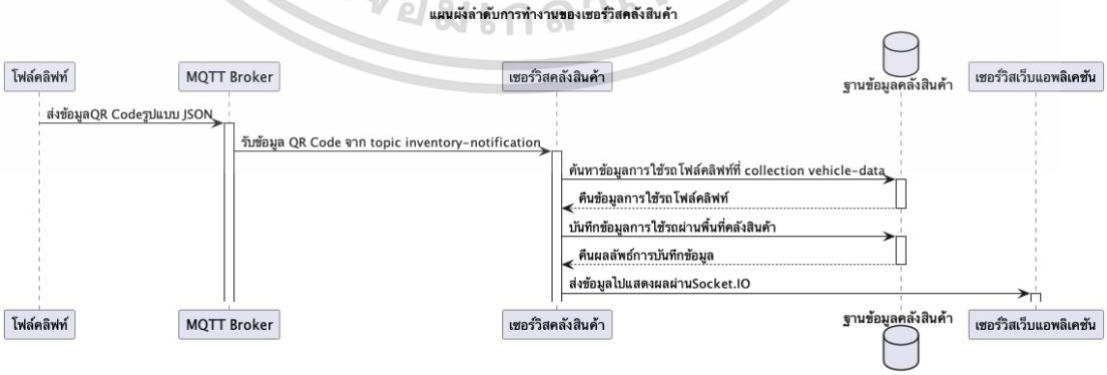
Endpoint	Method	คำอธิบาย	Request Body	Response Body
/vehicle/:id	PATCH	ทำการปรับปรุงข้อมูลรถโฟล์คลิฟท์ว่าคันไหนมีพนักงานกำลังใช้งาน	ข้อมูลรูปแบบออปเจ็กต์ ประกอบด้วย Name (ชื่อพนักงาน), driverId (รหัสพนักงาน), และ isVacancy (รถว่างหรือไม่),	ข้อมูลรูปแบบออปเจ็กต์ ประกอบด้วย Message (ข้อความบอกสถานะการปรับปรุงข้อมูล) และ HTTP status code
/vehicle/vacancy/:id	GET	ทำการเรียกดูข้อมูลรถโฟล์คลิฟท์ว่ามีใครกำลังใช้งาน	ไม่มี	ข้อมูลรูปแบบออปเจ็กต์ ประกอบด้วย Message (ข้อความบอกสถานะการใช้รถ) และ HTTP status code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 โครงสร้างข้อมูล JSON สำหรับสื่อสารผ่าน Socket.IO กับเซอร์วิสเว็บแอปพลิเคชัน

หัวข้อ	**คำอธิบาย**	**ตัวอย่างข้อมูล**	**รูปแบบข้อมูล**
zone	พื้นที่หลักในคลังสินค้า	A1	String
subZone	พื้นที่ย่อยในคลังสินค้า	02	String
vehicleId	รหัสประจำรถโฟล์คลิฟท์	64075c41ecf2f40e56cfd11d	String
timeStamp	เวลาที่รถวิ่งผ่าน QR Code	2023-05-22 15:16:31.446622	String
id	รหัสสุ่มตามtransaction	714dfe95-6be4-4067-ba43-07ca4c1197c9	String
staffId	รหัสพนักงานขับรถโฟล์คลิฟท์	64075adbbb838cb10916eb9a	String
event	เหตุการณ์แจ้งสถานะ ข้อมูล หรือการแจ้งเตือน	information	String

ลำดับการทำงานดังแสดงในรูปที่ 3.13 เริ่มจากรถโฟล์คลิฟท์เคลื่อนที่ผ่านตำแหน่งที่มี QR Code จากนั้นข้อมูลภาพ QR Code จะส่งผ่านระบบอินเทอร์เน็ตในทุกสิ่ง (Internet of Things) มายัง MQTT Broker ที่ topic inventory-notification จากนั้นเซอร์วิสคลังสินค้าจะทำหน้าที่รับข้อมูลเข้ามา (Subscribe message) ผ่านแพ็กเกจ mqtt ของ NestJS จากนั้นนำข้อมูลไปค้นหาในฐานข้อมูล Mongo DB คลังสินค้า (inventory-db) ที่ collection vehicle-data เพื่อรับข้อมูลการใช้รถโฟล์คลิฟท์ จากนั้นทำการบันทึกข้อมูลการใช้รถโฟล์คลิฟท์ผ่านพื้นที่คลังสินค้าที่ collection route-histories และสุดท้ายนำข้อมูลที่ได้ส่งผ่าน Socket.IO ไปแสดงผลที่เซอร์วิสเว็บแอปพลิเคชัน



รูปที่ 3.13 แผนผังลำดับการทำงานของเซอร์วิสคลังสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) เซอร์วิสการยืนยันตัวตน (Authentication Service) ทำหน้าที่ประมวลผลข้อมูลการยืนยันตัวตนของพนังงานขับรถโฟล์คลิฟท์และตรวจสอบสถานะการใช้งานรถโฟล์คลิฟท์ว่ามีใครใช้อยู่หรือไม่ เซอร์วิสการยืนยันตัวตน ประกอบด้วย API สำหรับสื่อสารกับเซอร์วิสเว็บแอปพลิเคชัน ดังแสดงในตารางที่ 3.4

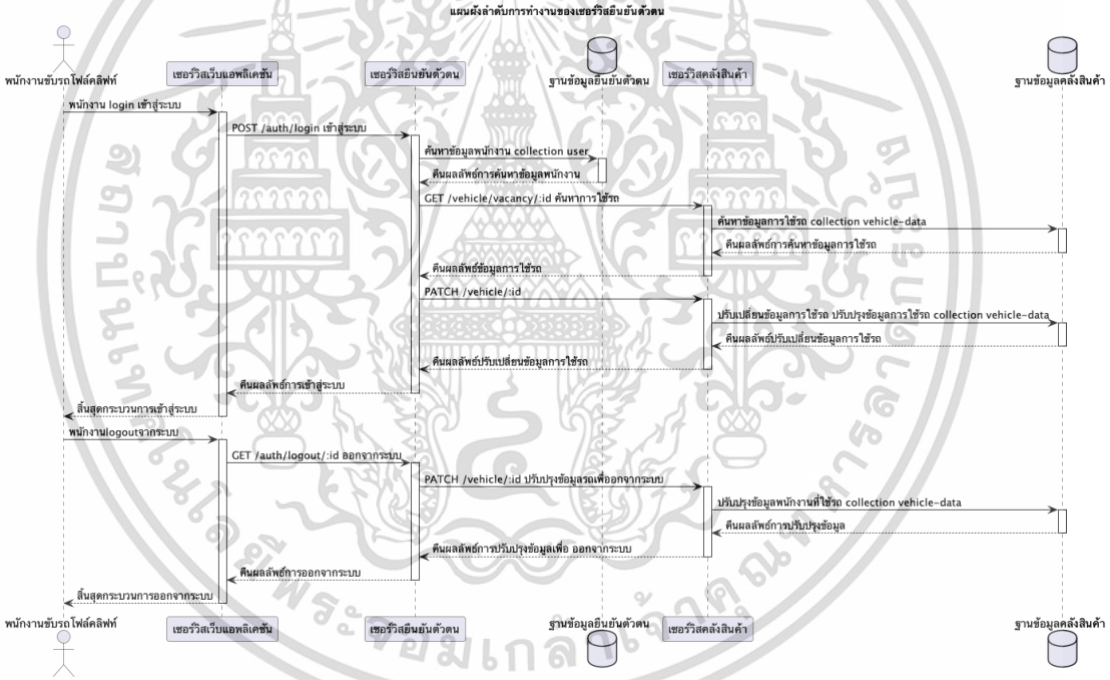
ตารางที่ 3.4 องค์ประกอบ API ของเซอร์วิสการยืนยันตัวตน

Endpoint	Method	คำอธิบาย	Request Body	Response Body
/auth/login	POST	กระบวนการเข้าสู่ระบบยืนยันตัวตนของพนักงานขับรถโฟล์-คลิฟท์และตรวจสอบว่ารถว่างอยู่หรือไม่	ข้อมูลรูปแบบออปเจ็กต์ ประกอบด้วย userId (รหัสพนักงาน), password (รหัสผ่าน), และ vehicleId (รหัสประจำรถ),	ข้อมูลรูปแบบออปเจ็กต์ ประกอบด้วย Message (ข้อความบอกสถานะเข้าสู่ระบบ) และ HTTP status code
/auth/logout/:id	GET	กระบวนการออกจากระบบและปรับปรุงข้อมูลการใช้รถโฟล์-คลิฟท์	ไม่มี	ข้อมูลรูปแบบออปเจ็กต์ ประกอบด้วย Message (ข้อความบอกสถานะการออกจากระบบ) และ HTTP status code

ลำดับการทำงานดังแสดงในรูปที่ 3.14 เริ่มจากพนักงานขับรถเข้าสู่ระบบโดยกรอกข้อมูลผ่านเซอร์วิสเว็บแอปพลิเคชันผ่านหน้าจอบนรถโฟล์คลิฟท์ เซอร์วิสเว็บแอปพลิเคชันเรียก API Method POST ไปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/auth/login ของเซอร์วิสการยืนยันตัวตน โดยส่ง request body ข้อมูล userId password และ vehicleId เพื่อทำการค้นหาข้อมูลพนักงานที่ทำการเข้าสู่ระบบที่ฐานข้อมูลยืนยันตัวตน Collection user เมื่อค้นหาสำเร็จจะทำการคืนข้อมูลพนักงานกลับมาจากนั้นเซอร์วิสการยืนยันตัวตน เรียก API Method GET /vehicle/vacancy/:id ไปที่เซอร์วิสคลังสินค้า โดย id จะเป็นตัวแปรข้อมูลรหัสประจำรถโฟล์คลิฟท์ ซึ่งใช้ค้นหาการใช้รถที่ฐานข้อมูลคลังสินค้า collection vehicle-data ว่ามีใครใช้รถอยู่ หรือลืมออกจากระบบหรือไม่ หลังจากได้รับข้อมูลส่งกลับมาแล้ว เซอร์วิสการยืนยันตัวตนจะทำการเรียก API Method PATCH /vehicle/:id ไปที่เซอร์วิสคลังสินค้า โดย id จะเป็นตัวแปรข้อมูลรหัสประจำรถโฟล์คลิฟท์ ขณะเดียวกันทำการแนบ request body ข้อมูล name driverId และ isVacancy เป็น false เพื่อใช้ในการปรับปรุงข้อมูลการใช้รถว่ามีคนใช้งานอยู่ ที่ฐานข้อมูลคลังสินค้า collection vehicle-data จากนั้นระบบทำการคืนผลลัพธ์การเข้าสู่ระบบกลับมายัง เซอร์วิสเว็บ-แอปพลิเคชัน เพื่อพนักงานสามารถดูข้อมูลระห่างการทำงานภายในคลังสินค้าได้

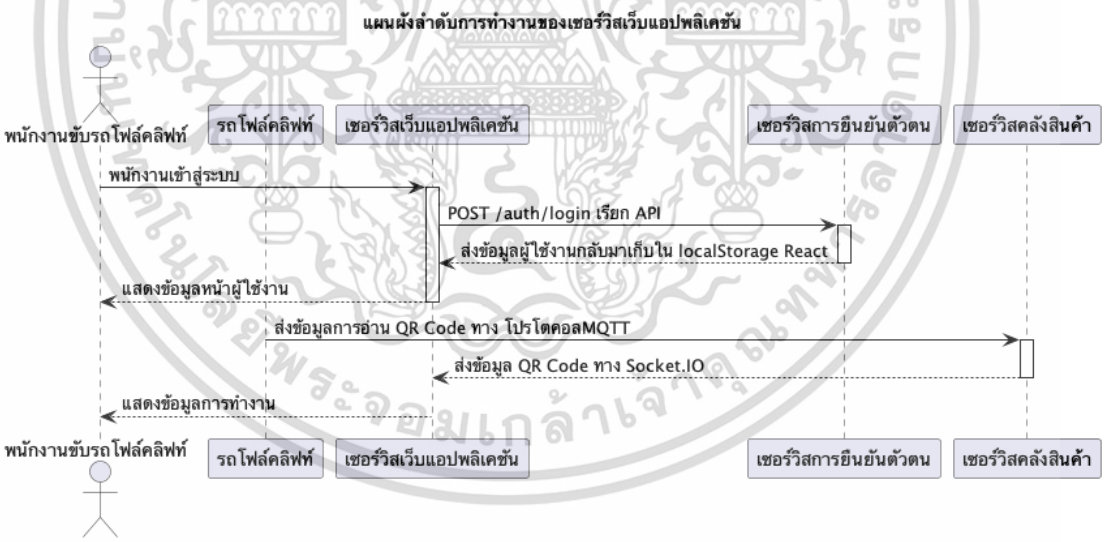


รูปที่ 3.14 แผนผังลำดับการทำงานของเซอร์วิสการยืนยันตัวตน

ในกรณีที่พนักงานออกจากระบบ เซอร์วิสเว็บแอปพลิเคชันทำการเรียก API Method GET ไปที่ /auth/logout/:id ของเซอร์วิสการยืนยันตัวตน โดย id จะเป็นตัวแปรข้อมูลรหัสประจำรถโฟล์คลิฟท์ ที่ใช้ส่งแนบไปพร้อมตัวแปร id ของ API Method PATCH /vehicle/:id ไปที่เซอร์วิสคลังสินค้า เพื่อปรับปรุงข้อมูลการใช้รถให้มีสถานะเป็นว่าง จากนั้นทำการคืนข้อมูลผลลัพธ์การออกจากระบบกลับมาที่เซอร์วิสเว็บแอปพลิเคชัน เป็นการสิ้นสุดกระบวนการออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) เซอร์วิสเว็บแอปพลิเคชัน (Web application service) ทำหน้าที่แสดงผลข้อมูลและเป็นส่วนควบคุมติดต่อกับพนักงานขับรถโฟล์คลิฟท์ (User interface) โดยพนักงานขับรถสามารถใช้งานเว็บแอปพลิเคชันผ่านหน้าจอภายในรถได้ ในรูปที่ 3.15 การทำงานของเซอร์วิสเว็บแอปพลิเคชันแบ่งการทำงานได้ดังต่อไปนี้ การทำงานแรกคือการเข้าสู่ระบบ เมื่อพนักงานเริ่มเข้ากะการทำงานพนักงานจะทำการเข้าสู่ระบบ เซอร์วิสเว็บแอปพลิเคชันจะเรียก API Method POST /auth/login ของเซอร์วิสการยืนยันตัวตน โดยส่ง request body เป็นข้อมูล userId password และ vehicleId เมื่อยืนยันตัวตนสำเร็จแล้วจะทำการส่งคืนข้อมูลกลับมาได้แก่ รหัสพนักงาน รหัสประจำรถโฟล์คลิฟท์ และ สถานะ HTTP Status โดยระบบจะเก็บข้อมูลรหัสพนักงานและรหัสประจำรถโฟล์คลิฟท์ ที่หน่วยความจำ localStorage ของ React เพื่อใช้ในการประมวลผล และ แสดงผลต่อไป ถัดมาการทำงานในส่วนสุดท้ายคือการรับข้อมูลเมื่อขับรถเคลื่อนที่ผ่าน QR Code ตามบริเวณพื้นที่ในคลังสินค้า เมื่อกล้องที่รถโฟล์คลิฟท์ รับภาพ QR Code มาข้อมูลจะถูกส่งไปประมวลผลที่เซอร์วิสคลังสินค้าผ่านโพรโทคอล MQTT จากนั้นเมื่อประมวลผลเสร็จแล้วจะทำการส่งคืนค่ากลับมาในรูปแบบข้อมูล JSON ผ่าน Socket.IO มาที่เซอร์วิสเว็บแอปพลิเคชัน โดย React จะใช้ไลบรารี socket.io-client ในการสื่อสารแบบเวลาจริง (Real time) กับ Socket.IO และแสดงผลข้อมูลการทำงานบนหน้าจอเป็นการสิ้นสุดการทำงาน

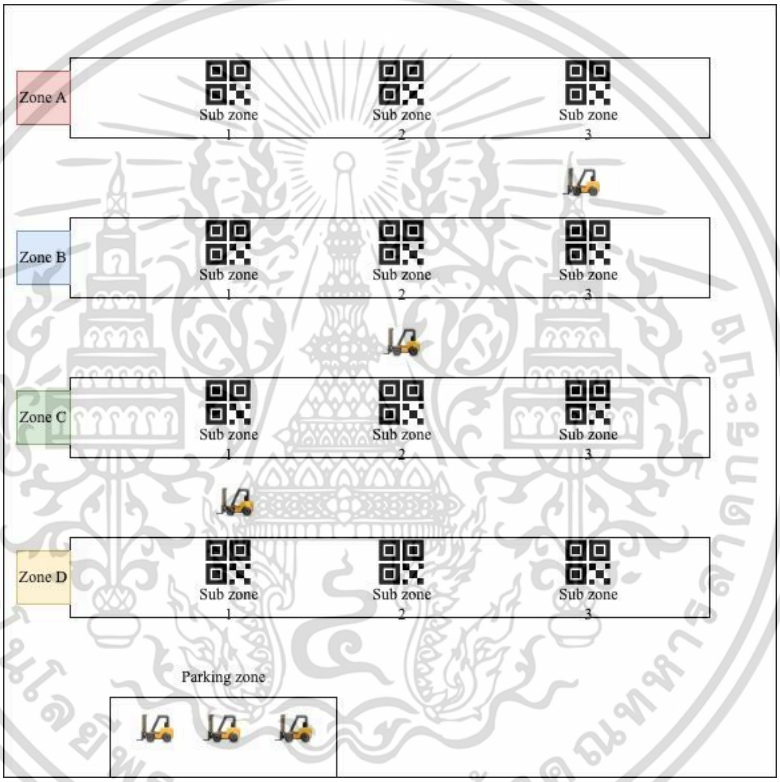


รูปที่ 3.15 แผนผังลำดับการทำงานของเซอร์วิสเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 กระบวนการทำงานกับโครงสร้างภายในคลังสินค้า

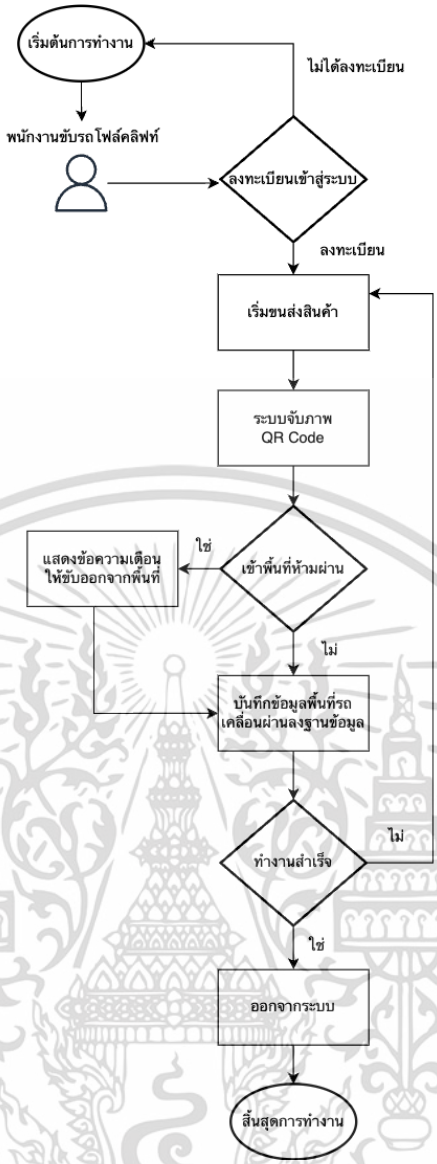
กระบวนการวิจัยการจัดการภายในคลังสินค้าได้ออกแบบการทดลองเป็นการติดตามการทำงานของพนักงานขับรถโฟล์คลิฟท์ขนส่งสินค้าภายในคลังสินค้า โดยจะติดตามตำแหน่งรวมถึงเวลาการเคลื่อนที่ของรถโฟล์คลิฟท์ว่าเคลื่อนที่ผ่านเข้าพื้นที่ใดบ้างและชื่อพนักงานที่ขับรถเพื่อขนส่งสินค้าในแต่ละพื้นที่ ในรูปที่ 3.16 ได้แสดงแบบจำลองพื้นที่ภายในคลังสินค้า โดยแบ่งออกเป็นพื้นที่สำหรับเก็บสินค้า ซึ่งแบ่งออกได้เป็นพื้นที่หลัก ได้แก่ Zone A Zone B Zone C และ Zone D ในแต่ละพื้นที่หลักจะมีแบ่งเป็นพื้นที่ย่อยสำหรับเก็บสินค้าเป็น Sub zone ต่างๆ และ พื้นที่สำหรับจอดรถโฟล์คลิฟท์



รูปที่ 3.16 แบบจำลองพื้นที่ภายในคลังสินค้า

ขั้นตอนการทำงานได้แสดงในรูปที่ 3.17 เริ่มจากพนักงานทำการเข้าสู่ระบบเพื่อยืนยันตัวตนผ่านเว็บแอปพลิเคชันจากนั้นทำการขนส่งสินค้าภายในคลังสินค้าโดยแต่ละพื้นที่จะติด QR Code ไว้ เมื่อรถเคลื่อนที่ผ่านแต่ละพื้นที่ระบบจะทำการเก็บภาพ QR Code และส่งไปประมวลผลเพื่อตรวจสอบว่าพื้นที่การทำงานนั้นสามารถผ่านได้หรือไม่ กรณีเป็นพื้นที่ห้ามผ่านระบบจะทำการแจ้งเตือนข้อความที่หน้าจอบนรถโฟล์คลิฟท์ให้พนักงานขับรถไม่เข้าพื้นที่ดังกล่าว จากนั้นระบบจะทำการบันทึกผลพื้นที่รถเคลื่อนผ่านลงในฐานข้อมูล เมื่อพนักงานทำงานสำเร็จแล้วสามารถนำรถมาคืนที่จุดจอดรถและทำการออกจากระบบเป็นการสิ้นสุดกระบวนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 ขั้นตอนการทำงานภายในคลังสินค้า

3.3 การทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง

การวิจัยในส่วนนี้ศึกษาเพื่อทำนายปริมาณการสั่งซื้อสินค้าล่วงหน้าออกจากคลังสินค้า โดยใช้แบบจำลองการเรียนรู้ของเครื่อง ประกอบไปด้วยกระบวนการ 3 ลำดับ ได้แก่ การเตรียมข้อมูลสำหรับการเรียนรู้ของเครื่อง การเรียนรู้ด้วยแบบจำลองการเรียนรู้ของเครื่อง และ การทำนายผลและวัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง

3.3.1 การเตรียมข้อมูลสำหรับการเรียนรู้ของเครื่อง

การทำนายปริมาณสินค้าล่วงหน้าจำเป็นต้องใช้ข้อมูลดิบในการนำมาเรียนรู้กับแบบจำลองการเรียนรู้ของเครื่อง โดยการวิจัยนี้ได้ใช้ชุดข้อมูลปริมาณสั่งซื้อสินค้าจากแหล่งข้อมูลแบบเปิดของ

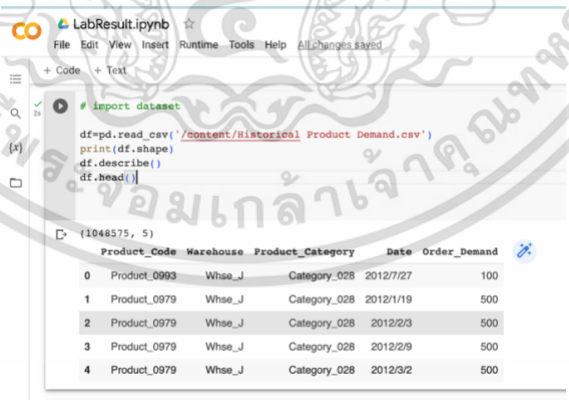
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บไซต์ Kaggle ชุดข้อมูลชื่อ Forecasts for Product Demand [38] มาใช้ในการเรียนรู้แบบจำลองการเรียนรู้ของเครื่อง (Training model) โดยข้อมูลประกอบไปด้วย 1,048,575 แถว และ 5 คอลัมน์ ในตารางที่ 3.5 แสดงรายละเอียดคุณสมบัติต่างๆของชุดข้อมูลที่นำมาใช้เรียนรู้แบบจำลองการเรียนรู้ของเครื่อง

ตารางที่ 3.5 คุณสมบัติของชุดข้อมูล Forecasts for Product Demand [38]

คุณสมบัติ	คำอธิบาย	ตัวอย่างข้อมูล
Product_code	รหัสสินค้า	Product_1359
Warehouse	คลังสินค้า	Whse_J
Product_Category	ประเภทสินค้า	Category_001
Date	วันที่	2012/2/9
Order_Demand	จำนวนการสั่งสินค้า	150000

ในการวิจัยได้ใช้เครื่องมือ Google Colab ซึ่งเป็น Software as a Service ในการเขียนโปรแกรมไพทอนเพื่อพัฒนาแบบจำลองการเรียนรู้ของเครื่อง ซึ่งทำงานบนคลาวด์จาก Google โดยจะทำงานบนพื้นฐานของ Jupyter Notebook โดยขั้นตอนแรกจะทำการอ่านชุดข้อมูลไฟล์ .csv Forecasts for Product Demand [38] โดยใช้ไพทอนไลบรารี pandas ซึ่งจะอ่านและแปลงข้อมูลให้อยู่ในรูปแบบ DataFrame หรือตารางข้อมูลสองมิติมีแถวและคอลัมน์ดังรูปที่ 3.18



รูปที่ 3.18 ตัวอย่างชุดข้อมูล DataFrame

ถัดมาเข้าสู่ขั้นตอนทำความสะอาดข้อมูล (Cleaning Data) โดยเริ่มจากการตรวจสอบชุดข้อมูลสูญหาย (Missing Data) ดังแสดงในรูปที่ 3.19 (ก) ซึ่งคุณสมบัติของข้อมูล Date มีข้อมูลสูญหายจำนวน 11,239 แถว จากนั้นทำการกำจัดข้อมูลสูญหายออกได้ผลลัพธ์ดังแสดงในรูปที่ 3.19 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
df.isnull().sum()
```

```
Product_Code          0
Warehouse             0
Product_Category      0
Date              11239
Order_Demand          0
dtype: int64
```

(ก)

```
[16]  df.isnull().sum()
```

```
Product_Code        0
Warehouse           0
Product_Category    0
Date                0
Order_Demand        0
dtype: int64
```

(ข)

รูปที่ 3.19 การตรวจสอบข้อมูลสูญหาย (ก) ผลลัพธ์การตรวจสอบข้อมูลสูญหาย (ข) ผลลัพธ์การหลังการกำจัดข้อมูลสูญหาย

จากนั้นทำการเลือกคุณสมบัติข้อมูลที่สนใจ ได้แก่ Date Product_Category และ Order_Demand และทำการใช้คำสั่ง groupby ของ pandas เพื่อทำการแบ่งกลุ่มข้อมูลโดยจะแบ่งตามคุณสมบัติ Date และ Product_Category จากนั้นใช้คำสั่ง sum ของ pandas เพื่อหาผลรวมของปริมาณสินค้า Order_Demand โดยผลลัพธ์ได้เป็น DataFrame ประเภทและจำนวนสินค้าที่มีกรอบเวลาเป็นปีดังแสดงในรูปที่ 3.20 และทำการเขียนกราฟเส้นโดยใช้ไพทอนไลบรารี seaborn ที่ทำงานบนพื้นฐานของ Matplotlib ได้ผลลัพธ์ดังรูปที่ 3.21 ข้อมูลแกนนอนคือปีและแกนตั้งคือจำนวนสินค้า โดยจำนวนประเภทสินค้าที่มากที่สุดคือ Category_019

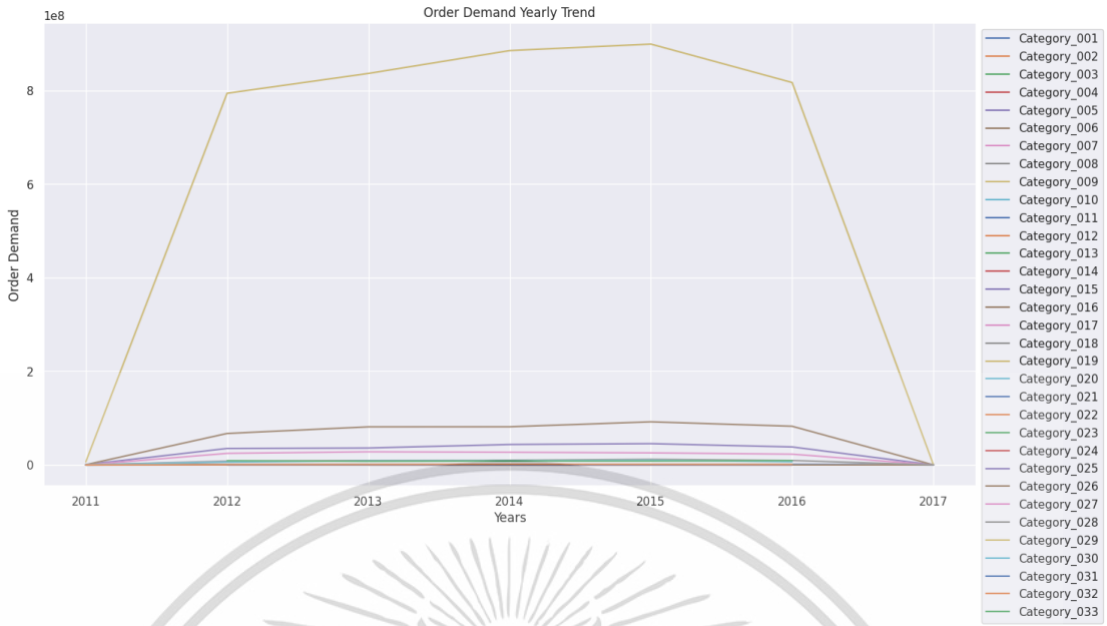
```
[22] product_cate_year
```

	year	Product_Category	Order_Demand
0	2011	Category_001	997
1	2011	Category_003	1708
2	2011	Category_004	2780
3	2011	Category_005	673750
4	2011	Category_006	490509
...	...	...	...
183	2017	Category_019	210800
184	2017	Category_020	200
185	2017	Category_021	1
186	2017	Category_026	15
187	2017	Category_028	250

188 rows × 3 columns

รูปที่ 3.20 ข้อมูล DataFrame ตามกลุ่มประเภทสินค้าและจำนวนสินค้าต่อปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 กราฟข้อมูลตามกลุ่มประเภทสินค้าและจำนวนสินค้าต่อปี

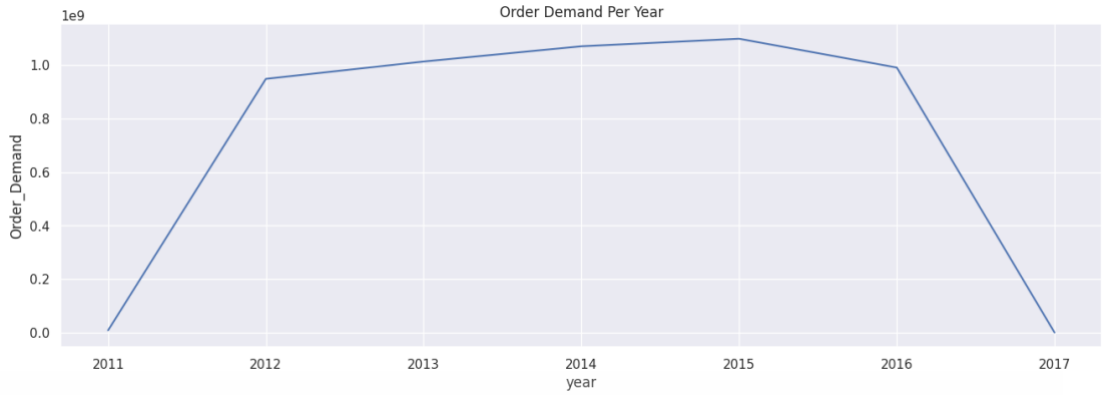
เมื่อสังเกตประเภทข้อมูลจากรูปที่ 3.21 จะเห็นได้ว่ามีจำนวนข้อมูลหลายประเภท แต่ในงานวิจัยนี้มีจุดประสงค์ที่สนใจศึกษาปริมาณสินค้าโดยรวมเท่านั้น จึงได้ทำการเลือกคุณสมบัติข้อมูลใหม่อีกครั้งโดยเลือกเฉพาะ Date และ Order_Demand โดยทำการใช้คำสั่ง groupby ของ pandas ด้วยคุณสมบัติ Date และทำการรวมปริมาณสินค้า Order_Demand ด้วยคำสั่ง sum ของ pandas ได้เป็น DataFrame ดังรูปที่ 3.22 และทำการเขียนกราฟเส้นโดยใช้ไพทอนไลบรารี seaborn ได้ผลลัพธ์เป็นกราฟปริมาณสินค้าต่อปีดังรูปที่ 3.23

product_demand_year

	year	Order_Demand
0	2011	8363894
1	2012	949259991
2	2013	1014087922
3	2014	1071178367
4	2015	1099398391
5	2016	991590399
6	2017	294967

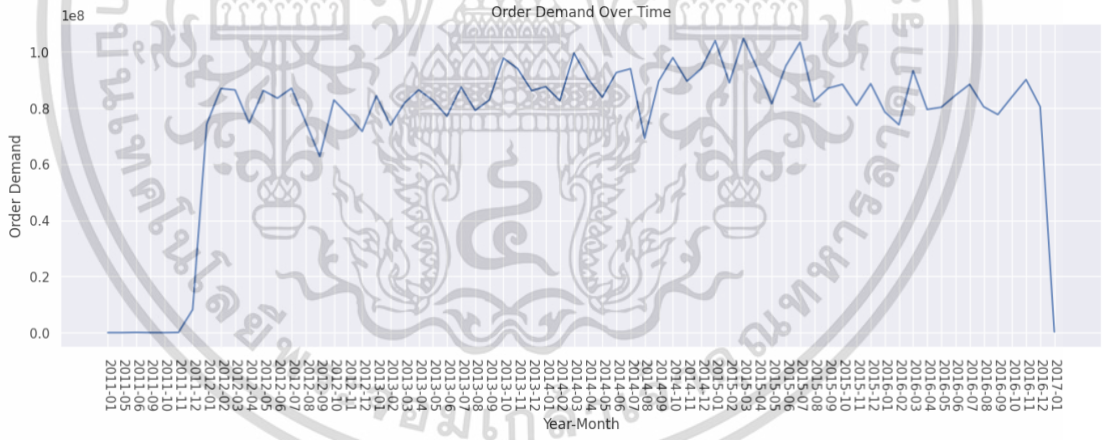
รูปที่ 3.22 ข้อมูล DataFrame ปริมาณสินค้าโดยรวมต่อปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 กราฟข้อมูลปริมาณสินค้าโดยรวมต่อปี

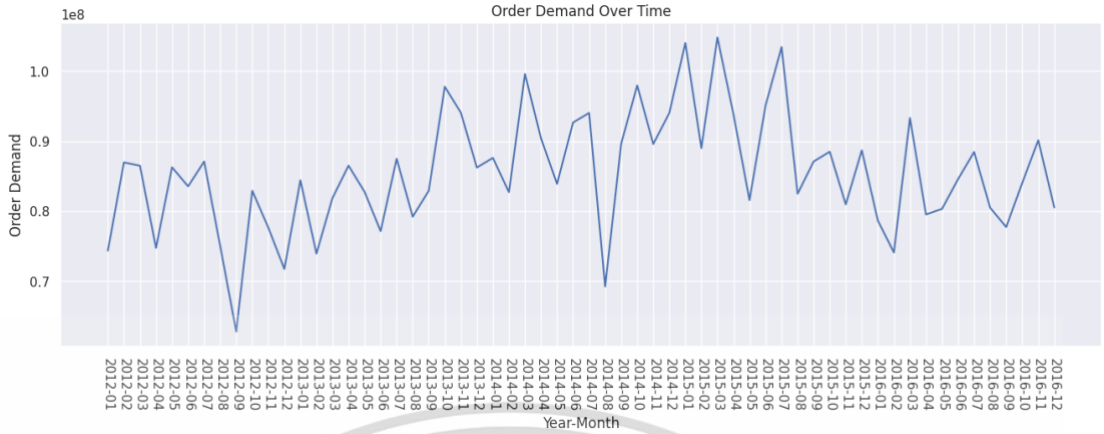
หากสังเกตที่รูปที่ 3.23 กราฟที่ตำแหน่งปี 2011 และปี 2017 มีปริมาณที่ต่ำกว่าชุดข้อมูลปีอื่นมากจึงทำการเขียนกราฟใหม่โดยปรับกรอบเวลาให้เป็นรายเดือนเพื่อดูข้อมูลได้ละเอียดมากขึ้น ได้ผลดังรูปที่ 3.24 โดยข้อมูลในปี 2011 มีปริมาณสินค้าน้อยกว่ากลุ่มข้อมูลปีอื่นๆและมีข้อมูลขาดหายไปบางเดือน และ ข้อมูลปี 2017 มีข้อมูลเพียงเดือนเดียว ด้วยเหตุนี้จึงทำการเลือกข้อมูลช่วงกรอบเวลาเฉพาะปี 2012 ถึงปี 2016 ที่มีข้อมูลครบถ้วนและค่าข้อมูลที่ไม่แตกต่างกันมากเกินไป



รูปที่ 3.24 กราฟข้อมูลปริมาณสินค้าโดยรวมต่อเดือน

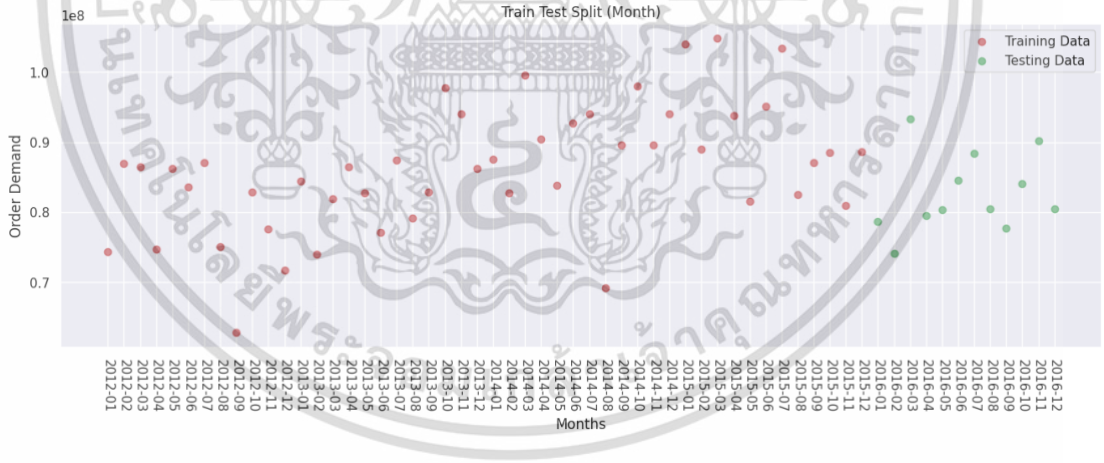
ถัดมาทำการเขียนกราฟเส้นแสดงปริมาณสินค้าต่อเดือนโดยเริ่มจากปี 2012 ถึงปี 2016 ได้ผลลัพธ์ดังรูปที่ 3.25 จากนั้นทำการแบ่งชุดข้อมูลออกเป็นข้อมูลฝึก (Training Dataset) สำหรับนำไปใช้เรียนรู้แบบจำลองการเรียนรู้ของเครื่อง และ ข้อมูลชุดทดสอบ (Testing Dataset) สำหรับทดสอบแบบจำลองการเรียนรู้ของเครื่องโดยใช้ฟังก์ชัน train_test_split ของไพทอนไลบรารี scikit-learn ซึ่งเป็นเครื่องมือในการทำการเรียนรู้ของเครื่องสร้างขึ้นมาจากโมดูลต่างๆ เช่น Numpy Scipy และ Matplotlib

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 กราฟข้อมูลปริมาณสินค้าโดยรวมตั้งแต่ปี 2012 ถึงปี 2016

การแบ่งข้อมูลจะใช้ชุดฝึกข้อมูล (Training Dataset) อยู่ที่ 80% คือตั้งแต่ปี 2012 ถึงปี 2015 และข้อมูลชุดทดสอบ (Testing Dataset) อยู่ที่ 20% คือปี 2016 จากนั้นทำการเขียนกราฟแผนภูมิจุดแบบกระจัดกระจาย (Scatter plot) ได้ดังรูปที่ 3.26 โดยข้อมูลสีแดงจะเป็นข้อมูลฝึกและสีเขียวเป็นข้อมูลชุดทดสอบ



รูปที่ 3.26 กราฟการแบ่งข้อมูลชุดฝึกและชุดทดสอบแบบจุดกระจัดกระจาย

3.3.2 การเรียนรู้ด้วยแบบจำลองการเรียนรู้ของเครื่อง

การสร้างแบบจำลองการเรียนรู้ของเครื่องในงานวิจัยนี้ได้แบ่งเป็น 3 แบบจำลองของเครื่อง ได้แก่

1.) แบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) จะใช้ไพทอนไลบรารี scikit-learn ในการทำ เริ่มจากใช้ฟังก์ชัน sklearn.linear_model นำเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Linear-Regression เข้ามาเพื่อใช้ฟังก์ชัน fit() ในการฝึกแบบจำลอง โดยฟังก์ชันนี้จะรับ2ตัวแปร ตัวแปรแรกจะรับข้อมูลฝึก (Training value) โดยในข้อมูลนี้จะเป็นวันที่ของสินค้า และตัวแปรที่สองคือเป้าหมายข้อมูล (Target value) โดยข้อมูลนี้จะเป็นปริมาณสินค้าของแต่ละวัน จากนั้นทำการสั่งฟังก์ชันทำงานเป็นการสิ้นสุดการฝึกข้อมูล

2.) แบบจำลอง Extreme Gradient- Boosting (XGBoost) จะใช้ไพทอนไลบรารีของ XGBoost ในการทำเริ่มจากนำเข้า xgboost.XGBRegressor จากนั้นใช้ฟังก์ชัน fit() ในการฝึกแบบจำลอง โดยฟังก์ชันนี้จะรับ2ตัวแปร ตัวแปรแรกจะรับข้อมูลฝึก (Training value) โดยในข้อมูลนี้จะเป็นวันที่ของสินค้า และตัวแปรที่สองคือเป้าหมายข้อมูล (Target value) โดยข้อมูลนี้จะเป็นปริมาณสินค้าของแต่ละวัน จากนั้นทำการสั่งฟังก์ชันทำงานเป็นการสิ้นสุดการฝึกข้อมูล

3.) แบบจำลอง Random Forest Regression จะใช้ไพทอนไลบรารี scikit-learn โดยเริ่มจากนำเข้า sklearn.ensemble นำเข้า RandomForestRegressor จากนั้นทำการตั้งค่าตัวแปร n_estimators เพื่อกำหนดจำนวนแผนภาพต้นไม้ในการฝึกแบบจำลอง Random Forest อยู่ที่ 50000 ต่อมาทำการใช้ฟังก์ชัน fit() ในการฝึกแบบจำลอง โดยฟังก์ชันนี้จะรับ2ตัวแปร ตัวแปรแรกจะรับข้อมูลฝึก (Training value) โดยในข้อมูลนี้จะเป็นวันที่ของสินค้า และตัวแปรที่สองคือเป้าหมายข้อมูล (Target value) โดยข้อมูลนี้จะเป็นปริมาณสินค้าของแต่ละวัน จากนั้นทำการสั่งฟังก์ชันทำงานเป็นการสิ้นสุดการฝึกข้อมูล

3.3.3 การทำนายผลและวัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง

การทำนายผลแบบจำลองข้อมูลเพื่อวัดประสิทธิภาพของแต่ละแบบจำลองของเครื่องในงานวิจัยนี้จะใช้ 2 ตัววัดผลได้แก่

1.) ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) หรือ MAE คือการหาค่าเฉลี่ยของความแตกต่างสัมบูรณ์ระหว่างค่าข้อมูลจริงและค่าข้อมูลจากการทำนายผลของแบบจำลองการเรียนรู้ของเครื่อง ถ้าหากค่า MAE มีค่าต่ำหมายความว่าค่าการทำนายนั้นมีความใกล้เคียงกับความเป็นจริง โดยเครื่องมือในการทำ MAE จะใช้ไพทอนไลบรารี scikit-learn โดยการนำเข้า sklearn.metrics และเรียกใช้ฟังก์ชัน mean_absolute_error() โดยฟังก์ชันนี้จะรับค่าตัวแปร 2 ตัวได้แก่ ค่าผลลัพธ์ของข้อมูลจริงโดยที่นี้จะใช้ค่าของปริมาณสินค้าของชุดข้อมูลฝึก และค่าปริมาณสินค้าที่ได้จากการทำนายของแบบจำลองการเรียนรู้ของเครื่อง

2.) ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) หรือ RMSE คือการหาค่าความแตกต่างของค่าข้อมูลจริงและค่าข้อมูลจากการทำนายผลของแบบจำลองการเรียนรู้ของเครื่อง โดยเครื่องมือในการทำ RMSE จะใช้ไพทอนไลบรารี scikit-learn โดยการนำเข้า sklearn.metrics และเรียกใช้ฟังก์ชัน mean_squared_error() โดยฟังก์ชันนี้จะรับค่าตัวแปร 2 ตัวได้แก่ ค่าผลลัพธ์ของข้อมูลจริงโดยที่นี้จะใช้ค่าของปริมาณสินค้าของชุดข้อมูลฝึก และค่าปริมาณสินค้าที่ได้จากการทำนายของแบบจำลองการเรียนรู้ของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) หรือ MAPE คือการคำนวนหาเปอร์เซ็นต์ของค่าความผิดผลาดที่ได้จากค่าเฉลี่ยของผลรวมค่าความแตกต่างระหว่างค่าที่ได้จากการทำนายและค่าข้อมูลจริง โดยเครื่องมือในการทำ MAPE จะใช้ไพทอนไลบรารี scikit-learn โดยการนำเข้า sklearn.metrics และเรียกใช้ฟังก์ชัน mean_absolute_percentage-error โดยฟังก์ชันนี้จะรับค่าตัวแปร 2 ตัวได้แก่ ค่าผลลัพธ์ของข้อมูลจริงโดยที่นี้จะใช้ค่าของปริมาณสินค้าของชุดข้อมูลฝึก และค่าปริมาณสินค้าที่ได้จากการทำนายของแบบจำลองการเรียนรู้ของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 บทนำ

ผลลัพธ์ของการทดลองนี้ประกอบไปด้วย 2 ส่วนได้แก่ ผลลัพธ์ของการการติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าด้วยอินเทอร์เน็ตในทุกสิ่งและMicroservices และผลลัพธ์ของการทำนายปริมาณสินค้าของชุดข้อมูล Forecasts for Product Demand [38] ด้วยแบบจำลองการเรียนรู้ของเครื่อง

4.2 ผลการทำงานระบบติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้า

การวิจัยการติดตามการทำงานของพนักงานขับรถโฟล์คลิฟท์จะเริ่มติดตามผลการทำงานตั้งแต่พนักงานขับรถเริ่มกะการทำงานเข้าสู่พื้นที่คลังสินค้าและลงทะเบียนยืนยันตัวตนพนักงานเข้าสู่ระบบ โดยผลการทดลองจะแบ่งได้เป็น 3 ลำดับดังต่อไปนี้

1.) พนักงานขับรถลงทะเบียนเข้าสู่ระบบผ่านเว็บแอปพลิเคชัน ผลลัพธ์ที่เกิดขึ้นคือเซอร์วิสเว็บแอปพลิเคชันจะทำการเรียก API Method POST /auth/login เพื่อส่งข้อมูล userId password และ vehicleId ไปที่เซอร์วิสยืนยันตัวตนเพื่อค้นหาข้อมูลพนักงานในฐานข้อมูลยืนยันตัวตน กรณีที่หาข้อมูลไม่พบจะได้ผลลัพธ์ API กลับมาที่เว็บแอปพลิเคชันดังตารางที่ 4.1 กรณีที่หาข้อมูลพนักงานพบดังในรูปที่ 4.1 เซอร์วิสยืนยันตัวตนจะทำการเรียก API Method GET /vehicle/vacancy/:id ของเซอร์วิสคลังสินค้าเพื่อตรวจสอบข้อมูลการใช้รถโฟล์คลิฟท์โดย id ที่ส่งไปจะเป็นรหัสประจำตัวรถ vehicleId

ตารางที่ 4.1 ผลลัพธ์การเรียก API กรณีไม่พบข้อมูลพนักงาน

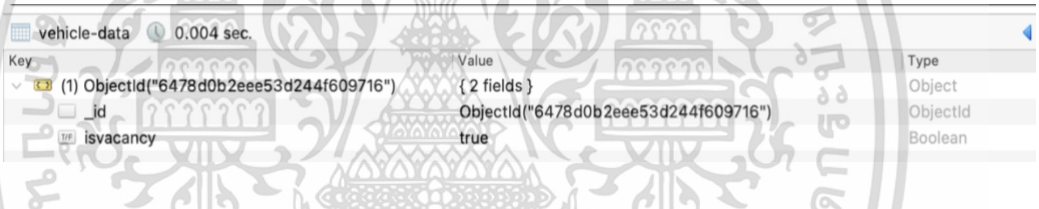
สถานะค้นพบข้อมูลพนักงาน	Response message	HTTP Status
ไม่พบข้อมูล	Data not found	404

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

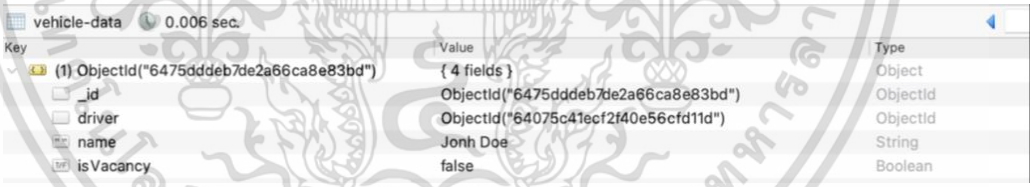
รูปที่ 4.1 ตัวอย่างการค้นหาข้อมูลพนักงานพบในฐานข้อมูลยืนยันตัวตนผ่านโปรแกรม Robo 3T

กรณีที่ผลการค้นหาข้อมูลการใช้รถพบว่ารถมีชื่อพนักงานอยู่จะได้ผลดังรูปที่ 4.2 (ก) และกรณีที่ไม่มีชื่อพนักงานดังรูปที่ 4.2 (ข) และได้ผลลัพธ์ API ดังตารางที่ 4.2 โดยการทำงานของการค้นหาข้อมูลการใช้รถเพื่อนำข้อมูลไปประยุกต์ใช้ตรวจสอบการลืมออกจากระบบของพนักงานหรือการปลอมแปลงการใช้รถได้

(ก)



(ข)



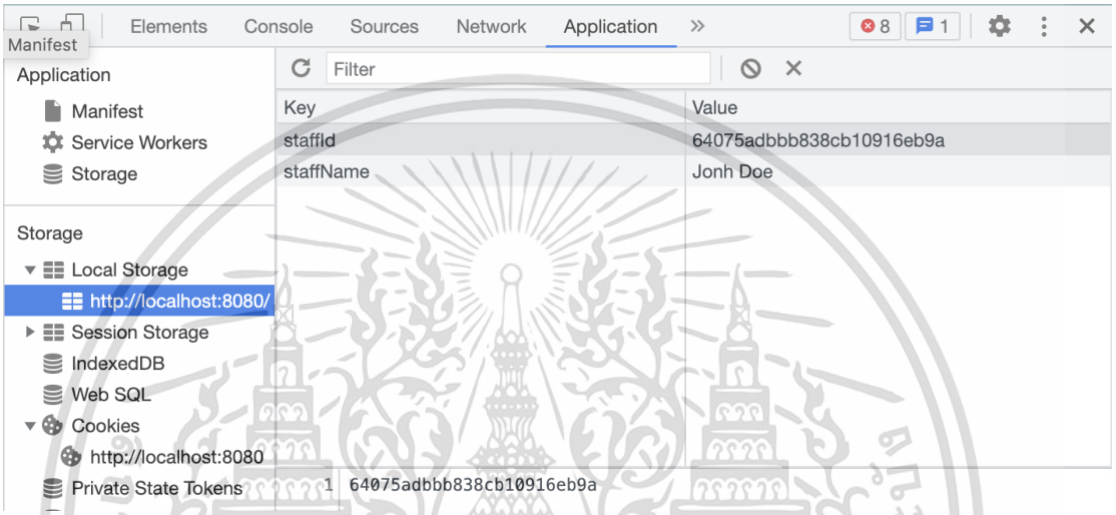
รูปที่ 4.2 ตัวอย่างการค้นหาข้อมูลการใช้รถโฟล์คลิฟท์ผ่านโปรแกรม Robo 3T (ก) ข้อมูลในฐานข้อมูลรถมีชื่อพนักงานอยู่ (ข) ข้อมูลในฐานข้อมูลกรณีรถไม่มีชื่อพนักงานอยู่

ตารางที่ 4.2 ผลลัพธ์การเรียก API ตรวจสอบสถานการณ์ใช้รถโฟล์คลิฟท์

สถานะรถมีชื่อพนักงาน	Response message	HTTP Status
False	“Isvacancy”:true	200
True	“Isvacancy”:false, “staffId”:"64075adbbb838cb10916eb9a"	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถัดมาเซอร์วิสยืนยันตัวตนทำการเรียก API Method PATCH /vehicle/:id ไปที่เซอร์วิสคลังสินค้า โดย id จะเป็นตัวแปรข้อมูลรหัสประจำรถโฟล์คลิฟท์เพื่อปรับปรุงข้อมูลการใช้รถให้เป็นของพนักงานปัจจุบันจากนั้นคืนผลลัพธ์การเข้าสู่ระบบกลับคืนไปยังเว็บแอปพลิเคชันได้ผลดังตารางที่ 4.3 และทำการบันทึกข้อมูล รหัสพนักงาน (staffId) และ ชื่อพนักงาน (staffName) ลงในlocal storage ของ web browser ดังรูปที่ 4.3 เพื่อใช้ในการแสดงข้อมูลและใช้ในการกรองข้อมูลจาก Socket.IO ของเซอร์วิสคลังสินค้า เป็นการสิ้นสุดผลของกระบวนการเข้าสู่ระบบ



รูปที่ 4.3 ผลลัพธ์การบันทึกข้อมูลพนักงานลง local storage

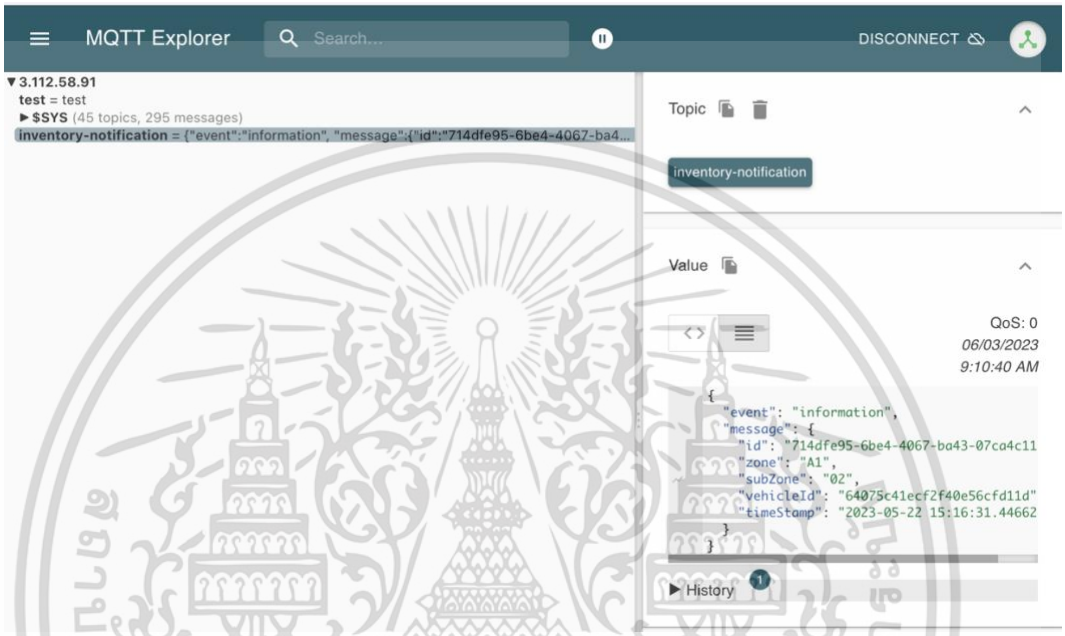
ตารางที่ 4.3 ผลลัพธ์การเข้าสู่ระบบของ API เซอร์วิสยืนยันตัวตน

สถานะการเข้าสู่ระบบ	Response message	HTTP Status
สำเร็จ	"vehicleId":"64075c41ecf2f40e56cfd11d", "staffId":"64075adbbb838cb10916eb9a", “staffName”:”Jonh Doe”	200
ไม่สำเร็จ	“Message”: ”Login failed”	401

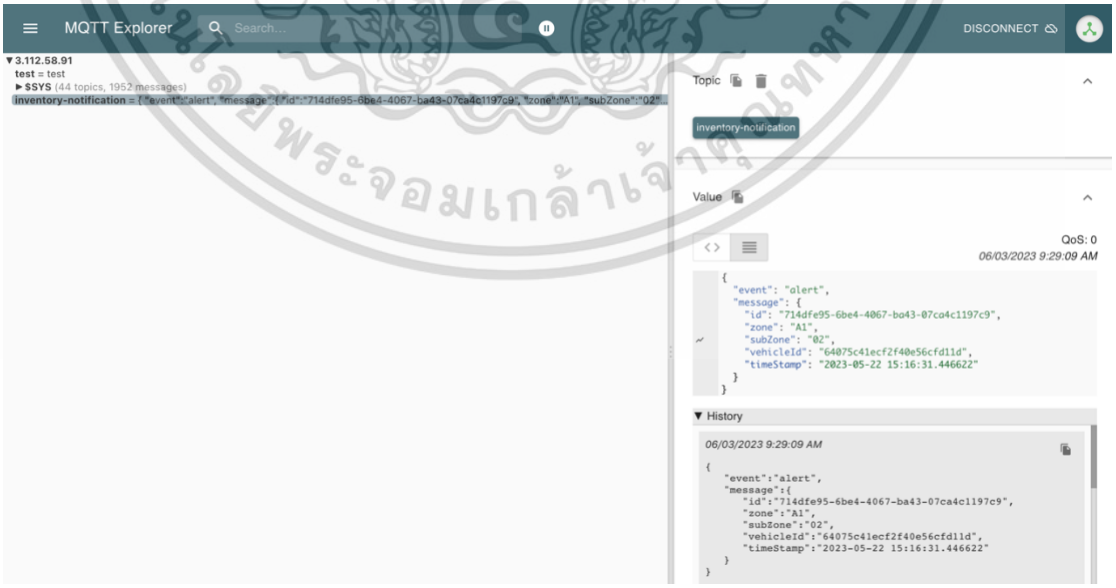
2.) เมื่อพนักงานทำการเข้าสู่ระบบเพื่อยืนยันตัวตนสำเร็จแล้ว พนักงานจะเริ่มทำการขนส่งสินค้าภายในคลังสินค้าตามตารางงานที่ได้รับ ในแต่ละพื้นที่ภายในคลังสินค้าจะมีการติดตั้งแผ่น QR Code เพื่อระบุตำแหน่งพื้นที่หลักและพื้นที่ย่อยต่างๆ เมื่อพนักงานได้ขับรถโฟล์คลิฟท์ผ่านเข้าไปในแต่ละพื้นที่เพื่อขนย้ายสินค้า ระบบกล้องอินเทอร์เน็ตในทุกสิ่งที่ติดตั้งไว้บริเวณด้านหน้ารถจะทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ่านข้อมูลภาพ QR Code และส่งข้อมูลไปยัง MQTT Broker ที่มีชื่อ topic ว่า inventory-notification ผ่านโพรโทคอล MQTT โดยผลลัพธ์ที่ได้จะเป็นข้อมูล JSON แบ่งเป็น 2 กรณีคือกรณีที่รถเคลื่อนที่ผ่านพื้นที่ห้ามผ่านดังตัวอย่างผลลัพธ์ในโปรแกรม MQTT Explorer แสดงในรูปที่ 4.4 และกรณีที่รถเคลื่อนที่ผ่านพื้นที่ผ่านได้ดังแสดงในรูปที่ 4.5 หากสังเกตผลลัพธ์จะมีข้อแตกต่างคือ event ของพื้นที่ผ่านได้จะเป็น information แต่ event ของพื้นที่ห้ามผ่านจะเป็น alert



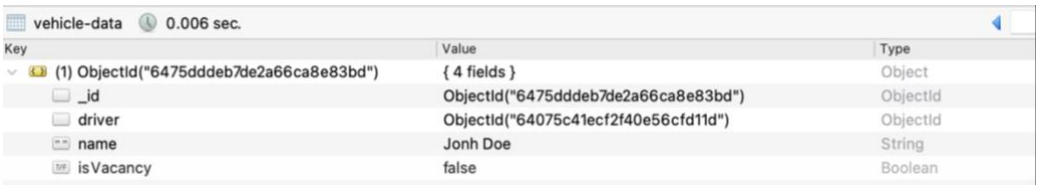
รูปที่ 4.4 ผลลัพธ์ข้อมูลรถผ่านพื้นที่ผ่านได้บน MQTT Broker



รูปที่ 4.5 ผลลัพธ์ข้อมูลรถผ่านพื้นที่ห้ามผ่านบน MQTT Broker

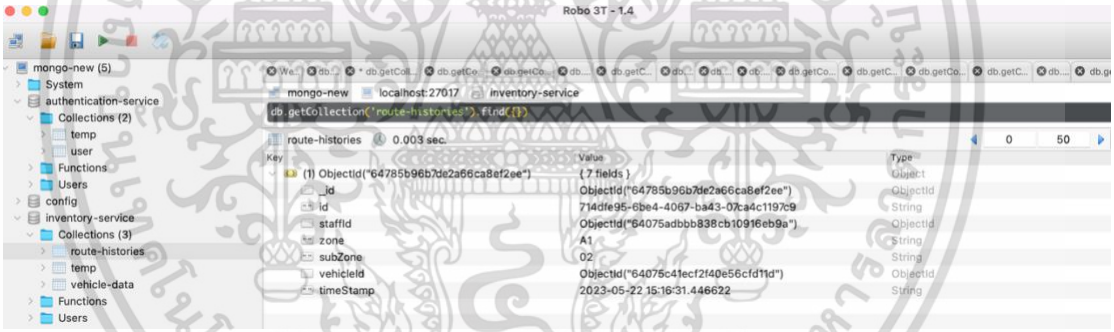
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถัดมาเซอร์วิสคลังสินค้าทำการรับข้อมูลจาก MQTT Broker ผ่าน topic inventory-notification และทำการค้นหาข้อมูลการใช้รถว่าพนักงานคนใดกำลังใช้อยู่โดยใช้ vehicleId ในการค้นหาที่ฐานข้อมูลคลังสินค้า collection vehicle-data ได้ผลลัพธ์ดังรูปที่ 4.6



รูปที่ 4.6 ผลลัพธ์การค้นหาข้อมูลการใช้รถด้วย vehicleId ที่ฐานข้อมูลคลังสินค้า

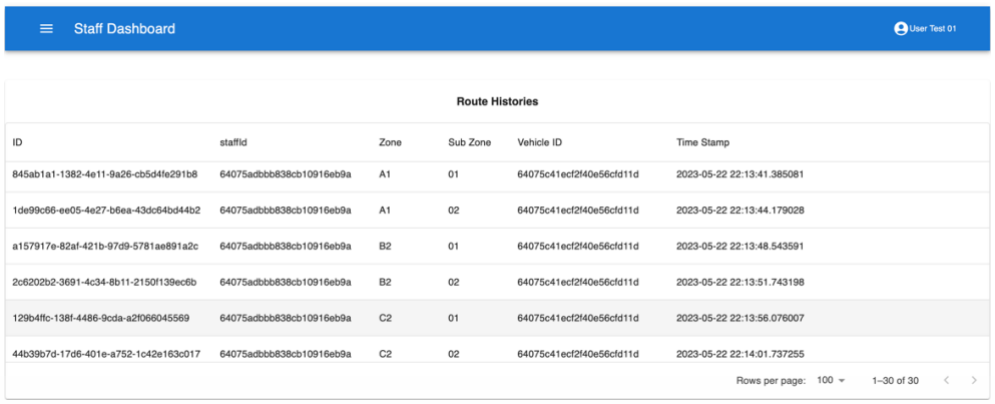
เซอร์วิสคลังสินค้าทำการบันทึกข้อมูลประวัติเส้นทางที่รถโฟล์คลิฟท์ผ่านที่ฐานข้อมูลคลังสินค้า collection route-histories โดยนำข้อมูลผู้ขับรถที่ค้นหาได้จากฐานข้อมูลคลังสินค้า collection vehicle data ในรูปที่ 4.6 มารวมกับข้อมูลจาก MQTT Broker ในรูปที่ 4.5 โดยผลลัพธ์การบันทึกข้อมูลดังรูปที่ 4.7



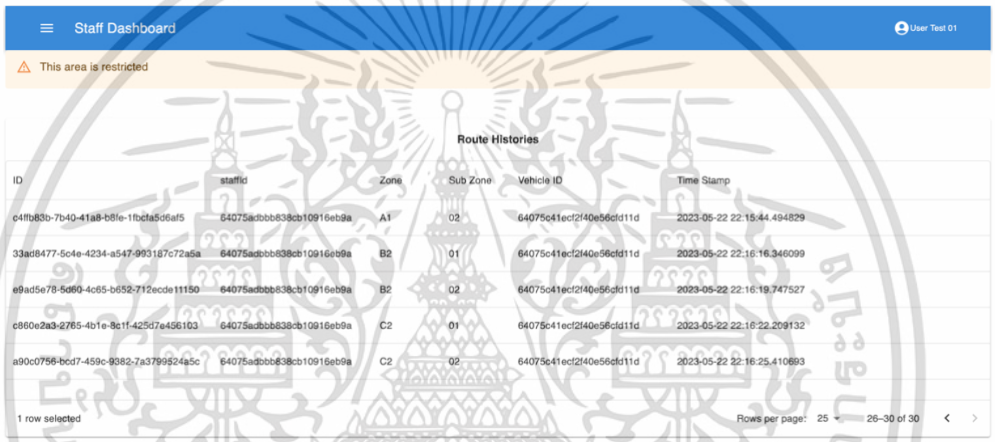
รูปที่ 4.7 ผลลัพธ์การบันทึกข้อมูลประวัติเส้นทางการทำงานรถโฟล์คลิฟท์ลงฐานข้อมูลคลังสินค้า

จากนั้นเซอร์วิสคลังสินค้าทำการส่งผลลัพธ์ข้อมูลการทำงานผ่าน Socket.IO ไปแสดงผลที่เว็บแอปพลิเคชันโดยผลลัพธ์จะแบ่งเป็น 2 กรณีคือกรณีการแจ้งข้อมูลรถเคลื่อนที่ผ่านพื้นที่อนุญาตผ่านได้ดังแสดงในรูปที่ 4.8 และ กรณีรถเคลื่อนที่ผ่านพื้นที่ห้ามผ่าน จะมีข้อความแสดงเตือนที่หน้าจอของเว็บแอปพลิเคชันดังแสดงในรูปที่ 4.9 เพื่อสั่งการให้พนักงานขับรถออกจากพื้นที่ห้ามผ่านทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



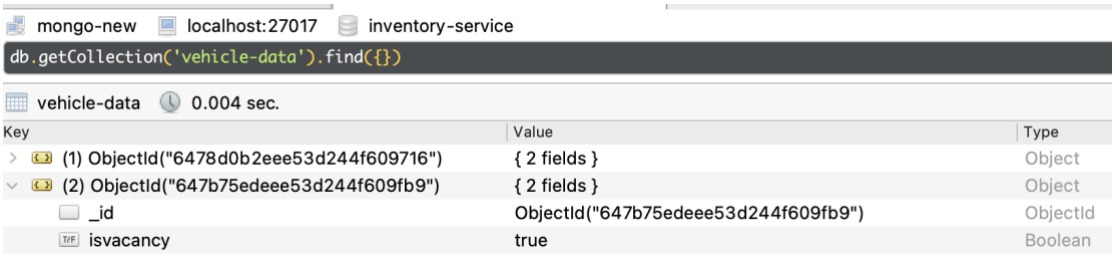
รูปที่ 4.8 ผลลัพธ์การแสดงข้อมูลรถโฟล์คลิฟท์ผ่านพื้นที่อนุญาตให้ผ่านได้บนเว็บแอปพลิเคชัน



รูปที่ 4.9 ผลลัพธ์การแสดงข้อมูลรถโฟล์คลิฟท์ผ่านพื้นที่ห้ามผ่านบนเว็บแอปพลิเคชัน

3.) หลังจากพนักงานขับรถโฟล์คลิฟท์ทำงานสำเร็จตามเป้าหมายงานที่ได้รับแล้วพนักงานสามารถนำรถไปจอดคืนบริเวณจุดจอดรถและทำการออกจากระบบข้อมูลผ่านเว็บแอปพลิเคชันได้ โดยกระบวนการทำงานของข้อมูลที่เกิดขึ้นคือ เว็บแอปพลิเคชันทำการเรียก API Method GET ไปที่ /auth/logout/:id ของเซอร์วิสการยืนยันตัวตน โดย id จะเป็นตัวแปรข้อมูลรหัสประจำรถโฟล์คลิฟท์ จากนั้นเซอร์วิสการยืนยันตัวตนจะเรียก API Method PATCH /vehicle/:id ไปที่เซอร์วิสคลังสินค้า โดย id จะเป็นตัวแปรข้อมูลรหัสประจำรถโฟล์คลิฟท์ เพื่อทำการปรับปรุงข้อมูลสถานะการใช้รถใหม่ที่ฐานข้อมูลคลังสินค้า collection vehicle-data ให้รถโฟล์คลิฟท์มีสถานะเป็นว่างดังผลลัพธ์แสดงในรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ผลลัพธ์การปรับปรุงข้อมูลสถานะการใช้รถโฟล์คลิฟท์ที่ฐานข้อมูลคลังสินค้า collection vehicle-data

เมื่อปรับปรุงข้อมูลสถานะการใช้รถโฟล์คลิฟท์ที่ฐานข้อมูลคลังสินค้าสำเร็จแล้วเซอร์วิสคลังสินค้าจะส่งผลลัพธ์ API กลับมาที่เซอร์วิสยืนยันตัวตนได้ผล API ดังตารางที่ 4.4 โดยผลลัพธ์แบ่งได้เป็นกรณีปรับปรุงข้อมูลสำเร็จและปรับปรุงข้อมูลไม่สำเร็จ

ตารางที่ 4.4 ผลลัพธ์การเรียก API เซอร์วิสคลังสินค้าเพื่อปรับปรุงข้อมูลการใช้รถโฟล์คลิฟท์

สถานะการทำงาน	Response message	HTTP Status
ทำงานสำเร็จ	“vehicleStatus”:”update successful”	200
ทำงานไม่สำเร็จ	“error”:”update vehicle status failed”	500

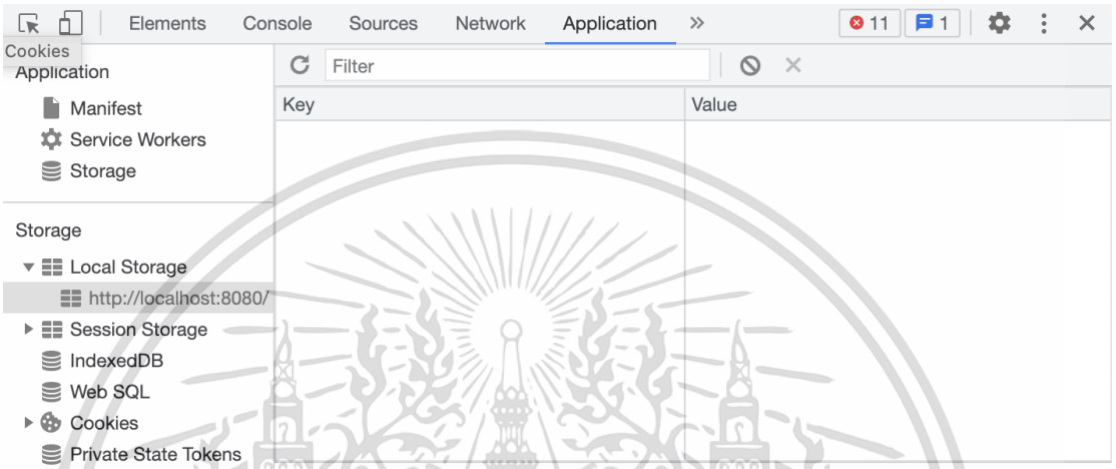
จากนั้นเซอร์วิสยืนยันตัวตนส่งผลลัพธ์การปรับปรุงข้อมูลสถานะการใช้รถโฟล์คลิฟท์ที่ได้รับจากเซอร์วิสคลังสินค้ากลับไปยังเว็บแอปพลิเคชันโดยผลลัพธ์แบ่งเป็นกรณีออกจากระบบสำเร็จและออกจากระบบไม่สำเร็จดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 ผลลัพธ์การออกจากระบบของ API เซอร์วิสยืนยันตัวตน

สถานะการทำงาน	Response message	HTTP Status
ออกจากระบบสำเร็จ	“success”:”logout successful”	200
ออกจากระบบไม่สำเร็จ	“error”:”logout failed”	500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซอร์วิสเว็บแอปพลิเคชันได้ผลลัพธ์การออกจากระบบจากเซอร์วิสยืนยันตัวตนแล้ว ในกรณีที่ออกจากระบบสำเร็จเซอร์วิสเว็บแอปพลิเคชันทำการลบข้อมูลรหัสพนักงาน (staffId) และ ชื่อพนักงาน (staffName) จาก local storage ออกได้ผลลัพธ์ดังรูปที่ 4.11 เป็นการจบการทำงานของการวิจัยการติดตามการทำงานของพนักงานขับรถโฟล์คลิฟท์ด้วยอินเทอร์เน็ตในทุกสิ่งและ Microservices



รูปที่ 4.11 ผลลัพธ์การลบข้อมูลรหัสพนักงานและชื่อพนักงานจาก local storage

4.3 ผลลัพธ์ของการทำนายปริมาณสินค้าของชุดข้อมูล Forecasts for Product Demand [38] ด้วยแบบจำลองการเรียนรู้ของเครื่อง

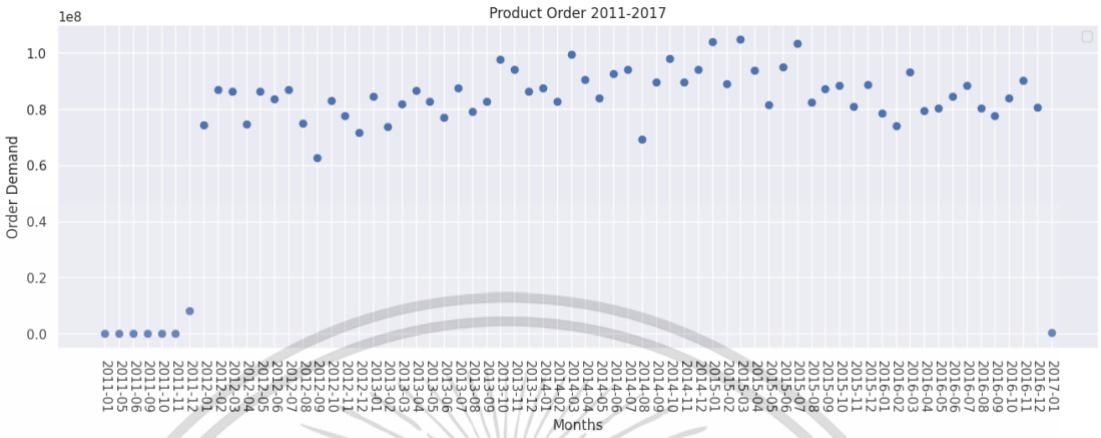
ผลลัพธ์การวิจัยทำนายปริมาณการสั่งซื้อสินค้าล่วงหน้าออกจากคลังสินค้า โดยใช้แบบจำลองการเรียนรู้ของเครื่องแบ่งได้เป็น 5 ส่วน ได้แก่ ผลลัพธ์การเตรียมและแบ่งชุดข้อมูล ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลอง Extreme Gradient Boosting (XGBoost) ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลอง Random Forest Regression และ การเปรียบเทียบประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องทั้ง 3 แบบจำลองโดยใช้ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) และ ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error)

4.3.1 ผลลัพธ์การเตรียมและแบ่งชุดข้อมูล

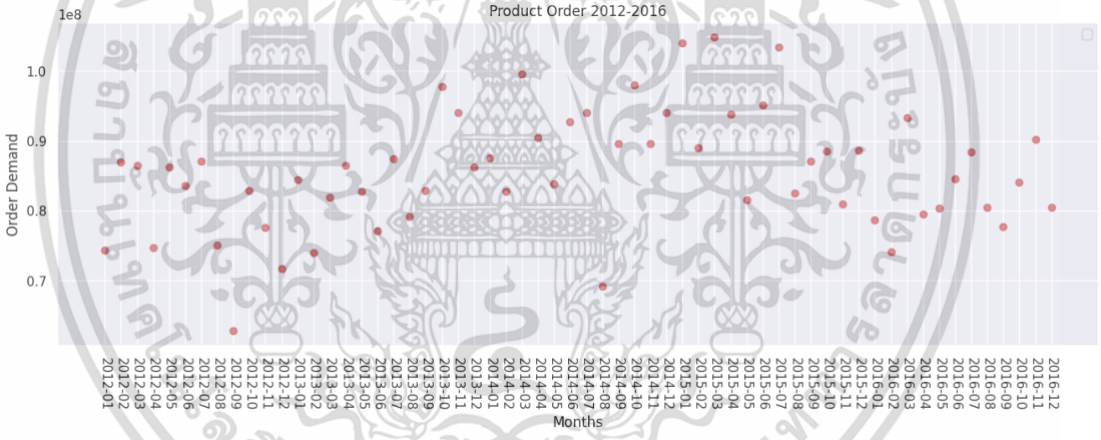
การนำเข้าชุดข้อมูล Forecasts for Product Demand [38] และทำการปรับกรอบเวลารวมปริมาณสินค้าให้เป็นช่วงเดือนได้ผลลัพธ์การเขียนกราฟแบบกระจัดกระจายดังรูปที่ 4.12 จะสังเกตได้ว่าข้อมูลปริมาณสินค้าของปี 2011 และปี 2017 มีปริมาณที่น้อยกว่าต่างจากชุดข้อมูลจึงทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกข้อมูลให้อยู่ในช่วงปี 2012 ถึงปี 2016 ได้ผลลัพธ์การเขียนกราฟการเขียนกราฟแบบกระจัดกระจายดังรูปที่ 4.13



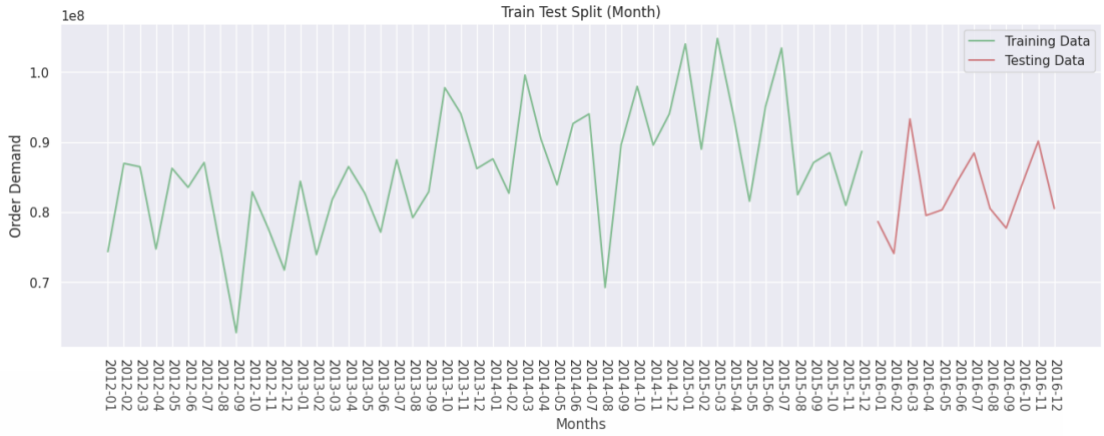
รูปที่ 4.12 ผลลัพธ์กราฟกระจัดกระจายปริมาณสินค้าต่อเดือนของปี 2011-2017



รูปที่ 4.13 ผลลัพธ์กราฟกระจัดกระจายปริมาณสินค้าต่อเดือนของปี 2012-2016

หลังจากเตรียมข้อมูลสำเร็จแล้วทำการแบ่งชุดข้อมูลฝึกที่ 80% (Training Dataset) และชุดข้อมูลทดสอบที่ 20% (Testing Dataset) ได้ผลลัพธ์การเขียนกราฟเส้นดังรูปที่ 4.14 โดยกราฟสีเขียวจะแทนชุดข้อมูลฝึกมีข้อมูลตั้งแต่ปี 2012 ถึงปี 2015 และกราฟสีแดงแทนชุดข้อมูลทดสอบประกอบด้วยข้อมูลของปี 2016

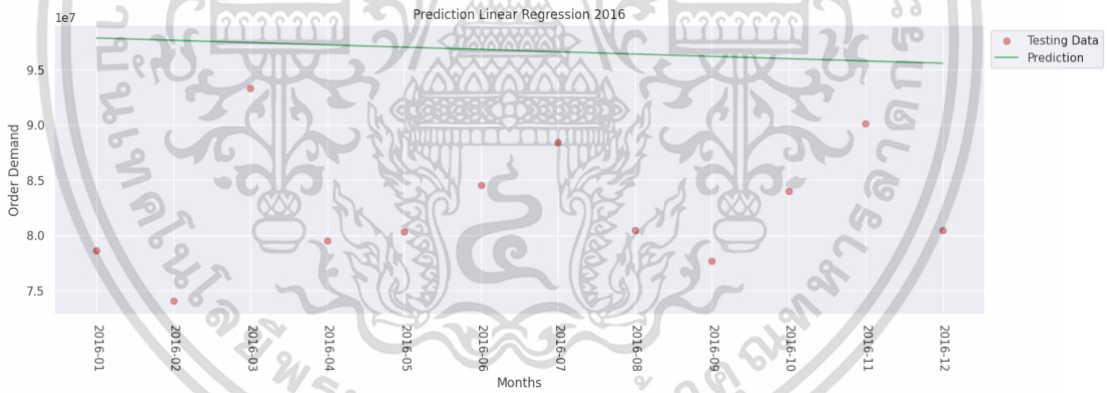
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 ผลลัพธ์กราฟเส้นข้อมูลชุดฝึกและชุดทดสอบ

4.3.2 ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น

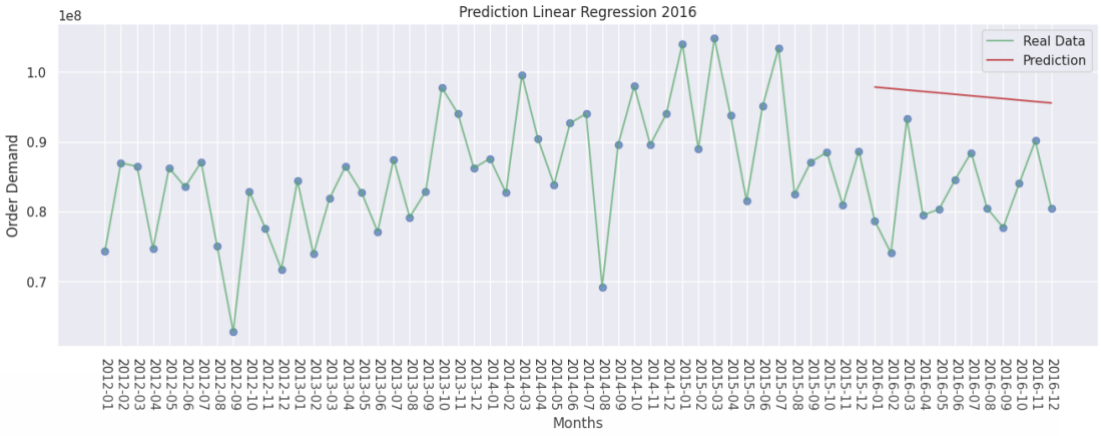
ผลลัพธ์การทำนายปริมาณสินค้าด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear-regression model) ได้ผลลัพธ์การเขียนกราฟเส้นของข้อมูลทำนายและกราฟข้อมูลกระจัดกระจายของข้อมูลทดสอบดังแสดงในรูปที่ 4.15



รูปที่ 4.15 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าเปรียบเทียบกับกราฟกระจัดกระจายข้อมูลทดสอบปี2016ด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น

จากนั้นทำการเขียนกราฟเส้นสีเขียวของชุดข้อมูลทั้งหมดตั้งแต่ปี 2012 ถึงปี 2016 และเขียนกราฟเส้นสีแดงของผลลัพธ์ข้อมูลทำนายปี 2016 โดยกรอบเวลาจะเป็นระยะเดือนได้ผลลัพธ์ดังรูปที่ 4.16 โดยจะสังเกตได้ว่ากราฟข้อมูลทำนายจะเป็นเส้นตรงเฉียงลงเพื่อบ่งบอกทิศทางปริมาณสินค้ากำลังลดลงในปี 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าด้วยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้นเปรียบเทียบกับกราฟเส้นชุดข้อมูลจริง

จากนั้นนำผลลัพธ์ข้อมูลการทำนายมาวัดประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่องด้วยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) หรือ MAE ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) หรือ RMSE และ ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) หรือ MAPE ได้ผลลัพธ์ดังตารางที่ 4.6

ตารางที่ 4.6 ผลลัพธ์การวัดประสิทธิภาพแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model)

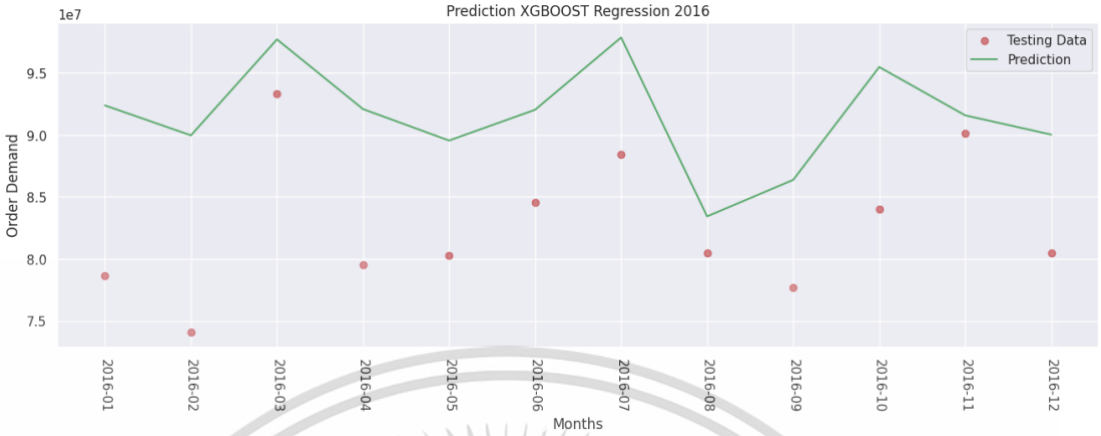
ตัวชี้วัด	ค่าชี้วัดความคลาดเคลื่อน	ร้อยละความคลาดเคลื่อน
ความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error)	14,063,931.67	-
ความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error)	15,130,123.25	-
ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error)	-	17.51

4.3.3 ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลอง Extreme Gradient Boosting

ผลลัพธ์การทำนายปริมาณสินค้าด้วยแบบจำลอง Extreme Gradient Boosting ได้ผลลัพธ์การเขียนกราฟเส้นของข้อมูลทำนายและกราฟข้อมูลกระจัดกระจายของข้อมูลทดสอบดังแสดงในรูป

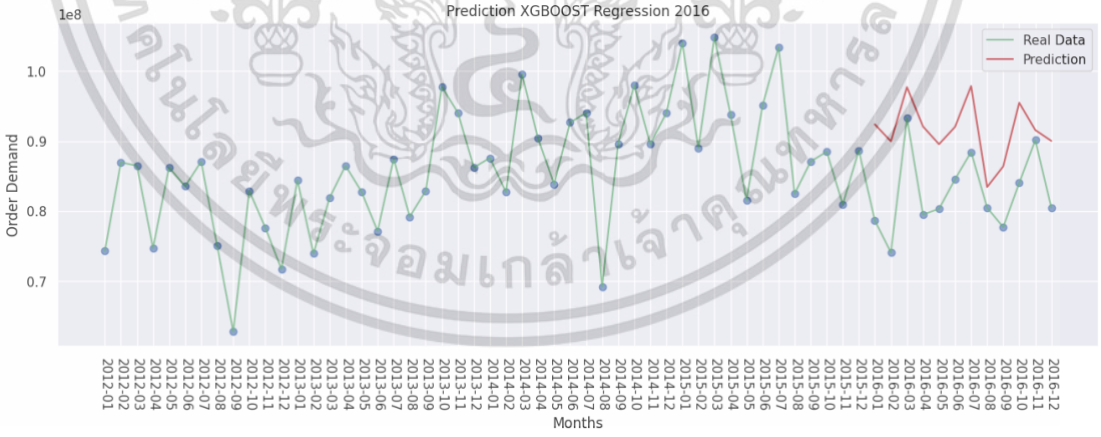
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 4.17 โดยจะสังเกตได้ว่ากราฟเส้นข้อมูลทำนายแสดงทิศทางได้ใกล้เคียงกับข้อมูลทดสอบที่มีความละเอียดมากขึ้น



รูปที่ 4.17 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าเปรียบเทียบกับกราฟกระจัดกระจายข้อมูลทดสอบปี2016ด้วยแบบจำลอง Extreme Gradient Boosting

จากนั้นทำการเขียนกราฟเส้นสีเขียวของชุดข้อมูลทั้งหมดตั้งแต่ปี 2012 ถึงปี 2016 และเขียนกราฟเส้นสีแดงของผลลัพธ์ข้อมูลทำนายปี 2016 โดยกรอบเวลาจะเป็นระยะเดือนได้ผลลัพธ์ดังรูปที่ 4.18 โดยจะสังเกตได้ว่ากราฟข้อมูลทำนายจะแสดงทิศทางใกล้เคียงกับทิศทางของข้อมูลจริง โดยทิศทางมีขึ้นและลงบ่งบอกรายละเอียดได้มากขึ้น



รูปที่ 4.18 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าด้วยแบบจำลอง Extreme Gradient Boosting เปรียบเทียบกับกราฟเส้นชุดข้อมูลจริง

จากนั้นนำผลลัพธ์ข้อมูลการทำนายมาวัดประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่องด้วยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) หรือ MAE ค่ารากที่สองของค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

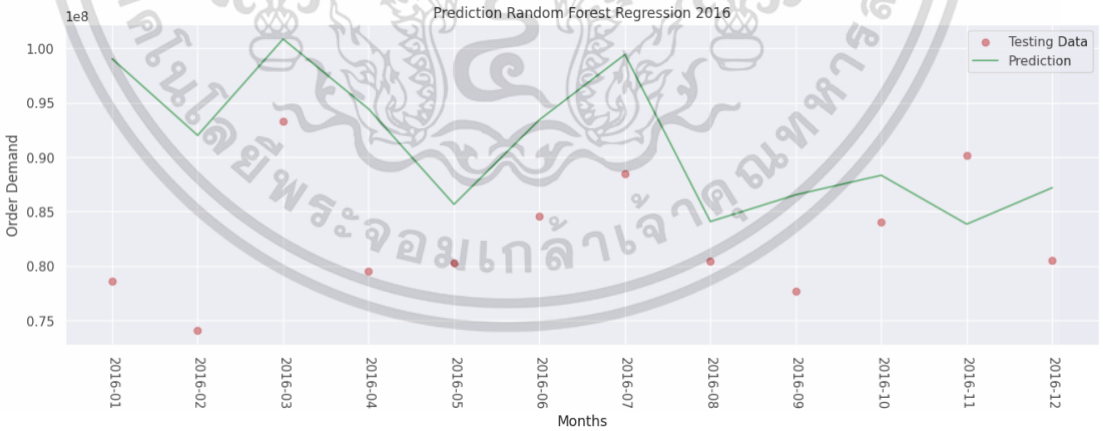
ความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) หรือ RMSE และ ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) หรือ MAPE ได้ผลลัพธ์ดังตารางที่ 4.7

ตารางที่ 4.7 ผลลัพธ์การวัดประสิทธิภาพแบบจำลอง Extreme Gradient Boosting

ตัวชี้วัด	ค่าชี้วัดความคลาดเคลื่อน	ร้อยละความคลาดเคลื่อน
ความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error)	8,907,540.08	-
ความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error)	9,826,971.44	-
ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error)	-	11.04

4.3.4 ผลลัพธ์การทำนายข้อมูลด้วยแบบจำลอง Random Forest Regression

ผลลัพธ์การทำนายปริมาณสินค้าด้วยแบบจำลอง Random Forest Regression ได้ผลลัพธ์การเขียนกราฟเส้นของข้อมูลทำนายและกราฟข้อมูลกระจัดกระจายของข้อมูลทดสอบดังแสดงในรูปที่ 4.19 โดยจะสังเกตได้ว่ากราฟเส้นข้อมูลทำนายแสดงทิศทางได้ความละเอียดมากขึ้น

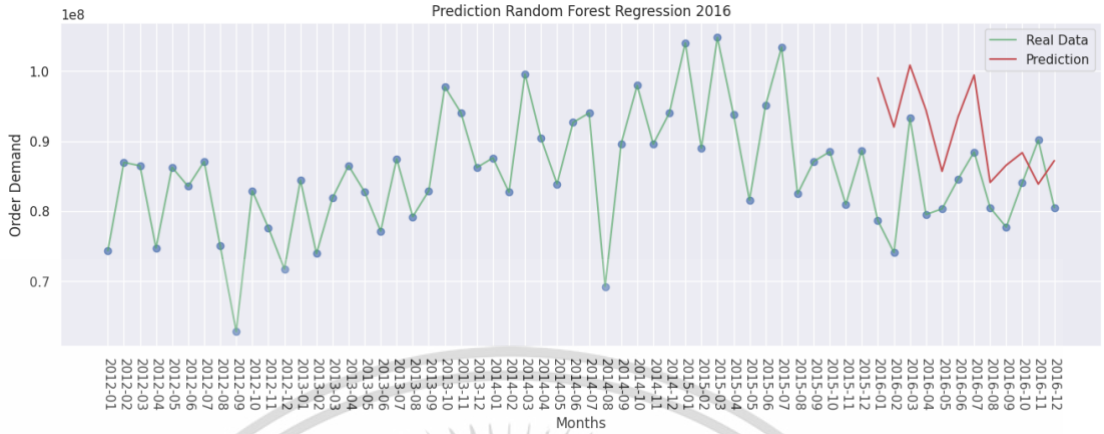


รูปที่ 4.19 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าเปรียบเทียบกับกราฟกระจัดกระจายข้อมูลทดสอบปี2016ด้วยแบบจำลอง Random Forest Regression

จากนั้นทำการเขียนกราฟเส้นสีเขียวของชุดข้อมูลทั้งหมดตั้งแต่ปี 2012 ถึงปี 2016 และเขียนกราฟเส้นสีแดงของผลลัพธ์ข้อมูลทำนายปี 2016 โดยกรอบเวลาจะเป็นระยะเดือนได้ผลลัพธ์ดัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.20 โดยจะสังเกตได้ว่ากราฟข้อมูลทำนายจะแสดงทิศทางใกล้เคียงกับทิศทางของข้อมูลจริง โดยทิศทางมีขึ้นและลงบ่งบอกรายละเอียดได้มากขึ้น



รูปที่ 4.20 ผลลัพธ์กราฟเส้นทำนายปริมาณสินค้าด้วยแบบจำลอง Random Forest Regression เปรียบเทียบกับกราฟเส้นชุดข้อมูลจริง

จากนั้นนำผลลัพธ์ข้อมูลการทำนายมาวัดประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่องด้วยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) หรือ MAE ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) หรือ RMSE และ ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) หรือ MAPE ได้ผลลัพธ์ดังตารางที่ 4.8

ตารางที่ 4.8 ผลลัพธ์การวัดประสิทธิภาพแบบจำลอง Random Forest Regression

ตัวชี้วัด	**ค่าชี้วัดความคลาดเคลื่อน**	**ร้อยละความคลาดเคลื่อน**
ความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error)	9,656,894.01	-
ความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error)	10,962,040.06	-
ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error)	-	11.91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 ผลลัพธ์การเปรียบเทียบประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องทั้ง 3 แบบจำลอง

เมื่อนำชุดข้อมูลฝึกไปเรียนรู้กับแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear-Regression Model) แบบจำลอง Extreme Gradient Boosting (XGBoost) และแบบจำลอง Random Forest Regression เรียบร้อยแล้ว จากนั้นทำการเปรียบเทียบประสิทธิภาพแบบจำลองทั้ง 3 แบบด้วยค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) หรือ MAE ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) หรือ RMSE และ ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) หรือ MAPE ได้ผลลัพธ์ดังตารางที่ 4.9

ตารางที่ 4.9 ผลลัพธ์การเปรียบเทียบความคลาดเคลื่อน MAE RMSE และ MAPE

แบบจำลอง	**ความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error)**	**ความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error)**	**ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error)**
การวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model)	14,063,931.67	15,130,123.25	17.51
Extreme Gradient Boosting	8,907,540.08	9,826,971.44	11.04
Random Forest Regression	9,656,894.01	10,962,040.06	11.91

ผลลัพธ์ของค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) ของแบบจำลอง Extreme Gradient Boosting มีค่าน้อยที่สุด โดยมีค่าอยู่ที่ 8,907,540.08 ถัดมาแบบจำลอง Random Forest Regression มีค่าอยู่ที่ 9,656,894.01 ซึ่งมากขึ้นมาจากแบบจำลอง Extreme-Gradient Boosting อยู่ที่ 749,353.93 ถัดมาแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear-Regression Model) จะมีค่ามากที่สุดอยู่ที่ 14,063,931.67 มากขึ้นมาจากแบบจำลอง Random-Forest Regression อยู่ที่ 4,407,037.66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ของค่าความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) ของแบบจำลอง Extreme Gradient Boosting มีค่าน้อยที่สุด โดยมีค่าอยู่ที่ 9,826,971.44 ถัดมาแบบจำลอง Random Forest Regression มีค่าอยู่ที่ 10,962,040.06 ซึ่งมากขึ้นมาจากแบบจำลอง Extreme-Gradient Boosting อยู่ที่ 1,135,068.62 ถัดมาแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear-Regression Model) จะมีค่ามากที่สุดอยู่ที่ 15,130,123.25 มากขึ้นมาจากแบบจำลอง Random-Forest Regression อยู่ที่ 4,168,083.19

ผลลัพธ์ของค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) ของแบบจำลอง Extreme Gradient Boosting มีค่าน้อยที่สุด โดยมีค่าอยู่ที่ร้อยละ 11.04 ถัดมาแบบจำลอง Random Forest Regression มีค่าอยู่ที่ร้อยละ 11.91 ซึ่งมากขึ้นมาจากแบบจำลอง Extreme Gradient Boosting อยู่ที่ร้อยละ 0.87 ถัดมาแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) จะมีค่ามากที่สุดอยู่ที่ร้อยละ 17.51 มากขึ้นมาจากแบบจำลอง Random Forest Regression อยู่ที่ร้อยละ 5.6

เมื่อเปรียบเทียบแบบจำลองทั้ง 3 แบบจำลองแล้วสามารถเรียงประสิทธิภาพได้ตามค่าความคลาดเคลื่อได้โดยแบบจำลองที่มีค่าความคลาดเคลื่อนที่น้อยกว่าจะมีประสิทธิภาพความแม่นยำที่ดีกว่าจึงเรียงลำดับแบบจำลองได้เป็นแบบจำลอง Extreme Gradient Boosting มีความแม่นยำมากที่สุดรองลงมาเป็นแบบจำลอง Random Forest Regression และสุดท้ายคือแบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) ที่มีความแม่นยำน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัย

5.1 สรุปผลและวิจารณ์การทดลอง

องค์ประกอบของงานวิจัยนี้ประกอบด้วย การติดตามการทำงานของรถโฟล์คลิฟท์ภายในคลังสินค้าด้วยอินเทอร์เน็ตในทุกสรรพสิ่งและ Microservices และ การทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่อง โดยการติดตามการทำงานของรถโฟล์คลิฟท์เริ่มจากพนักงานขับรถเข้าสู่ระบบเว็บแอปพลิเคชันจากนั้นเมื่อพนักงานเริ่มขนส่งสินค้าผ่านพื้นที่ต่างๆภายในคลังสินค้า ระบบกล้องจะอ่านภาพและส่งข้อมูลผ่านโพรโทคอล MQTT มายัง MQTT Broker โดยกำหนด topic เพื่อให้สามารถระบุผู้รับและผู้ส่งได้ โดยการใช้การรับส่งข้อมูลระหว่างอินเทอร์เน็ตในทุกสรรพสิ่งและ Microservices ด้วย MQTT นั้นส่งผลให้การรับส่งข้อมูลเป็นไปแบบเวลาจริง (Real time) สามารถสื่อสารแบบสองทิศทางได้ (Bi-directional Communication) รองรับเครือข่ายการสื่อสารขนาดใหญ่ โดยการทำงานของ MQTT จะทำงานบนเครือข่ายของ TCP/IP จากนั้นเซอร์วิสคลังสินค้าทำการประมวลผลข้อมูลที่ได้รับมาและจัดเก็บข้อมูลลงฐานข้อมูลเพื่อให้ผู้ใช้งานหรือผู้จัดการคลังสินค้าสามารถดึงข้อมูลไปใช้วิเคราะห์วางแผนการทำงานหรือวัดประสิทธิภาพการทำงานของพนักงานได้ จากนั้นเซอร์วิสคลังสินค้าทำการส่งข้อมูลผ่าน Socket.IO กลับมาแสดงผลที่เว็บแอปพลิเคชันได้แบบเวลาจริง (Real time) เนื่องจากการทำงานของ Socket.IO ทำงานบนพื้นฐานของ WebSocket ซึ่งเป็นการสื่อสารแบบสองทิศทางพร้อมกัน (Full-duplex Communication) บนโพรโทคอล TCP นอกจากนี้ Socket.IO สามารถทำการเชื่อมต่ออย่างอัตโนมัติหากการเชื่อมต่อสูญหาย ทำให้การทำงานของระบบมีประสิทธิภาพ

การทำนายปริมาณสินค้าด้วยแบบจำลองการเรียนรู้ของเครื่องนั้นได้ใช้ชุดข้อมูล จากแหล่งข้อมูลแบบเปิดของเว็บไซต์ Kaggle ชุดข้อมูลชื่อ Forecasts for Product Demand [38] โดยกระบวนการเตรียมข้อมูลจะเลือกคุณสมบัติข้อมูลเฉพาะวันที่ Date และ ปริมาณสินค้า Order_Demand ในการใช้ทำแบบจำลองการเรียนรู้ของเครื่อง ซึ่งการวิเคราะห์ข้อมูลทำให้ทราบได้ว่าปริมาณสินค้าในปี 2011 และปี 2017 มีข้อมูลบางเดือนขาดหายไปส่งผลให้จำนวนสินค้าต่างจากชุดข้อมูลปีอื่นๆมากเกินไป จึงทำการเลือกชุดข้อมูลใหม่ให้อยู่ในกรอบเวลาปี 2012 ถึง 2016 จากนั้นทำการแบ่งชุดข้อมูลฝึกที่80%และชุดข้อมูลทดสอบที่20% ผลลัพธ์ได้เป็น ข้อมูลชุดฝึกประกอบด้วยข้อมูลตั้งแต่ปี 2012 ถึง 2015 และชุดทดสอบจะเป็นปี 2016 จากนั้นทำการนำข้อมูลชุดฝึกเข้าไปเรียนรู้ในแบบจำลอง 3 แบบจำลองได้แก่ แบบจำลองการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression Model) แบบจำลอง Extreme Gradient Boosting (XGBoost) และ แบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Random Forest Regression โดยผลลัพธ์การวัดและเปรียบเทียบประสิทธิภาพแบบจำลองจะใช้ ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error) หรือ MAE ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error) หรือ RMSE และ ค่าเฉลี่ยร้อยละความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error) หรือ MAPE โดยแบบจำลองการวิเคราะห์การถดถอยเชิงเส้นมีความแม่นยำน้อยที่สุด ซึ่งมีค่า RMSE อยู่ที่ 15,130,123.25 MAE ที่ 14,063,931.67 และ MAPE ที่ร้อยละ 17.51 ถัดมาแบบจำลอง Random Forest Regression จะมีความแม่นยำมากขึ้นโดยค่า RMSE อยู่ที่ 10,962,040.06 MAE ที่ 9,656,894.01และ MAPE ที่ร้อยละ 11.91 และ แบบจำลอง Extreme Gradient Boosting (XGBoost) ที่มีความแม่นยำมากที่สุดโดยมีค่าความคลาดเคลื่อนน้อยที่สุด ซึ่งมีค่า RMSE อยู่ที่ 9,826,971.44 MAE ที่ 8,907,540.08 และ MAPE ที่ร้อยละ 11.04

5.2 ข้อเสนอแนะ

งานวิจัยนี้ต้องการนำอินเทอร์เน็ตในทุกสรรพสิ่งและ Microservices มาประยุกต์ใช้กับการจัดการคลังสินค้าเพื่อติดตามการทำงานของพนักงานและเก็บข้อมูลการทำงานต่างๆ เพื่อใช้วางแผนงานรวมถึงจัดการทรัพยากรต่างๆ และการใช้แบบจำลองการเรียนรู้ของเครื่องเพื่อทำนายปริมาณสินค้าในอนาคต เพื่อเตรียมการวางแผนจัดการปริมาณการผลิตได้ โดยการประยุกต์ใช้นี้ต้องการนำเสนอเพื่อเป็นแบบอย่างการประยุกต์ใช้อินเทอร์เน็ตในทุกสรรพสิ่งและ Microservices และการทำนายปริมาณด้วยแบบจำลองการเรียนรู้ของเครื่อง ซึ่งสามารถนำไปใช้กับอุตสาหกรรมอื่นๆได้ เช่น การขนส่งสินค้าโดยติดตามสถานะของการขนส่ง หาเส้นทางที่ขนส่งได้รวดเร็วที่สุด อุตสาหกรรมการเกษตรโดยการควบคุมวิเคราะห์ข้อมูลเพื่อให้ได้ผลผลิตทางการเกษตรที่ดี อุตสาหกรรมพลังงานโดยการทำนายปริมาณการใช้พลังงานนำข้อมูลมาปรับปรุงโครงพื้นฐานให้ใช้พลังงานได้อย่างคุ้มค่าที่สุดเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] P. Toorakidsana. **“Python คืออะไร? เป็นภาษาที่ง่ายที่สุดจริงหรือ?”** [Online]. เข้าถึงได้จาก : https://blog.skooldio.com/what-is-python. 2021.

[2] นิรนาม. **“Python คืออะไร 2023”** [Online]. เข้าถึงได้จาก : https://aws.amazon.com/th/what-is/python. 2023.

[3] N. Chuntra. **“OpenCV คืออะไร?”** [Online]. เข้าถึงได้จาก : https://medium.com/@nut.ch40/opencv-คืออะไร-8771e2a4c414. 2018.

[4] Somchai. **“ทำ QR Code Decoder”** [Online]. เข้าถึงได้จาก : http://raspberrypi-thailand.blogspot.com/2018/06/qr-code-decoder.html. 2561.

[5] Saixii. **“JSON คืออะไร”** [Online]. เข้าถึงได้จาก : https://saixiii.com/what-is-json. 2017.

[6] นิรนาม. **“เข้าใจ JSON ใน 5 นาที”** [Online]. เข้าถึงได้จาก : https://www.borntodev.com/2020/02/28/what-is-json. 2563.

[7] นิรนาม. **“RESTful API คืออะไร”** [Online]. เข้าถึงได้จาก : https://aws.amazon.com/th/what-is/restful-api. 2023.

[8] นิรนาม. **“MQTT คืออะไร”** [Online]. เข้าถึงได้จาก : https://aws.amazon.com/th/what-is/mqtt. 2023.

[9] นิรนาม. **“MQTT protocol”** [Online]. เข้าถึงได้จาก : https://www.javatpoint.com/mqtt-protocol. 2023.

[10] K. Muangsen. **“ทำไมถึงต้องใช้ Typescript ทั้งที่ก็มี JavaScript อยู่แล้ว ?”** [Online]. เข้าถึงได้จาก : https://www.borntodev.com/2021/05/13/ทำไมถึงต้องใช้-typescript. 2564.

[11] K. L. **“ทำความรู้จัก Nest JS และวิธีการใช้!”** [Online]. เข้าถึงได้จาก : https://kritsadapk.medium.com/ทำความรู้จัก-nest-js-และวิธีการใช้-6a498bef8b01. 2020.

[12] นิรนาม. **“Mongodb คืออะไร”** [Online]. เข้าถึงได้จาก : https://appmaster.io/th/blog/mongodb-khuue-aair. 2022.

[13] T. Babparn. **“Becomes CloudDev - เริ่มต้นใช้ AWS EC2”** [Online]. เข้าถึงได้จาก : https://tpbabparn.medium.com/becomes-clouddev-series-เริ่มต้นใช้-aws-ec2-2dbc6f6058fc. 2019.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[14] นิรนาม. **“React คืออะไร? ไขข้อสงสัยสำหรับมือใหม่ + แนวทางการหัด React ตั้งแต่เริ่มต้น”** [Online]. เข้าถึงได้จาก : https://www.designil.com/react-คืออะไร. 2017.

[15] นิรนาม. **“Socket.io 101”** [Online]. เข้าถึงได้จาก : https://medium.com/@idealphase/socket-io-101-47f93c33af2f. 2017.

[16] T. Das. **“Google Colab: Everything you Need to Know”** [Online]. Available : https://geekflare.com/google-colab. 2022.

[17] P. Ngarmtrakulchol. **“Cheatsheet วิธีใช้ และเทคนิคใน Pandas (Python) ฉบับสมบูรณ์”** [Online]. เข้าถึงได้จาก : https://blog.datath.com/cheatsheet-pandas. 2023.

[18] N. Anuchit. **“Building simple graphs with Matplotlib”** [Online]. เข้าถึงได้จาก : https://stackpython.co/tutorial/building-simple-graphs-with-matplotlib. 2020.

[19] Aakashns. **“Data Visualization using Python, Matplotlib and Seaborn”** [Online]. Available : https://jovian.com/aakashns/python-matplotlib-data-visualization. 2023.

[20] Ravikiran A S. **“An Interesting Guide to Visualize Data Using Python Seaborn”** [Online]. Available : https://www.simplilearn.com/tutorials/python-tutorial/python-seaborn. 2023.

[21] A. McDonald. **“Seaborn Pairplot: Enhance Your Data Understanding with a Single Plot”** [Online]. Available : https://towardsdatascience.com/seaborn-pairplot-enhance-your-data-understanding-with-a-single-plot-bf2f44524b22. 2022.

[22] นิรนาม. **“รู้จักใช้ Scikit-learn เหมือนมีโปรในเกม”** [Online]. เข้าถึงได้จาก : https://medium.com/achieve-space/รู้จักใช้-scikit-learn-เหมือนมีโปรในเกม-4cebd3195518. 2020.

[23] นิรนาม. **“เจาะลึก “Linear Regression” คืออะไร พร้อมตัวอย่างง่ายๆใน excel”** [Online]. เข้าถึงได้จาก : https://digi.data.go.th/en/blog/linear-regression-en. 2022.

[24] L. Lowphansirikul. **“สอน Machine Learning — Linear Regression”** [Online]. เข้าถึงได้จาก : https://medium.com/@lalital/สอน-machine-learning-linear-regression-bf07d230a7ee. 2018.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[25] V. Kanade. **“What Is Linear Regression? Types, Equation, Examples, and Best Practices for 2022”** [Online]. Available : https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression. 2023.

[26] S. Dobilas. **“XGBoost: Extreme Gradient Boosting — How to Improve on Regular Gradient Boosting?”** [Online]. Available : https://towardsdatascience.com/xgboost-extreme-gradient-boosting-how-to-improve-on-regular-gradient-boosting-5c6acf66c70a. 2021.

[27] D. Leventis. **“XGBoost Mathematics Explained”** [Online]. Available : https://dimleve.medium.com/xgboost-mathematics-explained-58262530904a. 2018.

[28] W. Daroontham. **“เจาะลึก Random Forest !!!— Part 2 of รู้จัก Decision Tree, Random Forest, และ XGBoost”** [Online]. เข้าถึงได้จาก : https://medium.com/@witchapongdaroontham/เจาะลึก-random-forest-part-2-of-รู้จัก-decision-tree-random-forest-และ-xgboost-79b9f41a1c1c. 2018.

[29] N. Beheshti. **“Random Forest Regression”** [Online]. เข้าถึงได้จาก : https://towardsdatascience.com/random-forest-regression-5f605132d19d. 2022.

[30] นิรนาม. **“MAE, MSE, RMSE — เลือกใช้ยังไงดีมาลองดูที่ความหมาย”** [Online]. เข้าถึงได้จาก : https://medium.com/c-g-datacommunity/mse-rmse-mae-เลือกใช้ยังไงดีมาลองดูที่ความหมาย-17b37b0b14b3. 2020.

[31] S. Acharya. **“What are RMSE and MAE?”** [Online]. Available : https://towardsdatascience.com/what-are-rmse-and-mae-e405ce230383. 2021.

[32] M. Dhouioui., T. Frikha. “Intelligent Warehouse Management System” **2020 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS),** 2020. pp. 1-5.

[33] F. Men., J. Guo., Y. Luan. “IoT Warehouse Management System Based on ACO Path Planning” **2022 IEEE 5th International Conference on Electronics Technology (ICET),** 2022. pp. 1008-1013.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[34] D. Gkoulis., C. Bardaki., E. Politi., I. Routis., M. Nikolaidou., G. Dimitrakopoulos., D. Anagnostopoulos. "An Event-based Microservice Platform for Autonomous Cyber-Physical Systems: the case of Smart Farming" **2021 16th International Conference of System of Systems Engineering (SoSE)**, 2021. pp. 31-36.

[35] T. G. Patil., V. Venkatesh., V. R. Patil., Kanchan. A. Purohit., S. Karwa. "A Web Application for Customer Segment based Black Friday Sales Forecast" **2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N),** 2021. pp. 1620-1624.

[36] K. L. Keung., C. K. M. Lee., Y. H. Yiu. "A Machine Learning Predictive Model for Shipment Delay and Demand Forecasting for Warehouses and Sales Data," **2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),** 2021. pp. 1010-1014.

[37] Puneet., S. Sharma., D. Deepika., G. Singh. "Intelligent Warehouse Stocking Using Machine Learning" **2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC),** 2021. pp. 1-6.

[38] F. Zhao **"Forecasts for Product Demand"** [Online]. Available : https://www.kaggle.com/datasets/felixzhao/productdemandforecasting. 2017.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ผลงานทางวิชาการที่ได้รับการตีพิมพ์

[1] P. Sithiyopasakul *et al*., "Inventory Management System based on IoT and Microservices Architecture Design," *2023 International Electrical Engineering Congress (iEECON)*, Krabi, Thailand, 2023, pp. 395-399, doi: 10.1109/iEECON56657.2023.10126548.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายชวินกร ปิยะธนานุกูล
วัน เดือน ปีเกิด	11 กันยายน 2542
ที่อยู่	195 เขต ประเวศ แขวง ประเวศ กรุงเทพฯ 10250 โทร. 0929521989
ประวัติการศึกษา	2564 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมสารสนเทศ (เกียรตินิยมอันดับ 2) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้