การพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม

FORECASTING THE PRICE OF CONDOMINIUM ASSETS

วรากร บุญปัญญา

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต (คณิตศาสตร์ประยุกต์)

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์

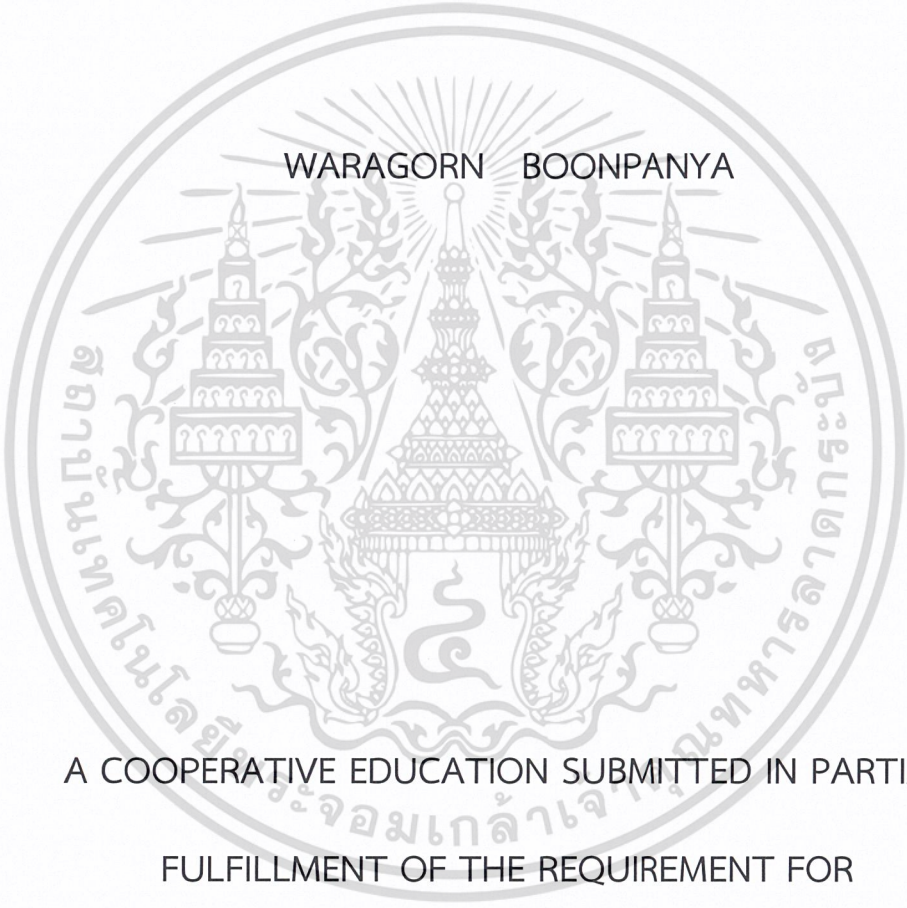
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FORECASTING THE PRICE OF CONDOMINIUM ASSETS

WARAGORN BOONPANYA



A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL

FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (APPLIED MATHEMATICS)

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม
	Forecasting the Price of Condominium Assets
ชื่อนักศึกษา	นางสาววรากร บุญปัญญา รหัสนักศึกษา 58050144
ปริญญา	วิทยาศาสตรบัณฑิต (คณิตศาสตร์ประยุกต์)
ภาควิชา	คณิตศาสตร์
ปีการศึกษา	2561
อาจารย์ที่ปรึกษา	ดร.บุษยมาส พิมพ์พรรณชาติ

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาสตรบัณฑิต (คณิตศาสตร์ประยุกต์) ประจำปีการศึกษา 2561

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.เดชา สมนะ ประธานกรรมการ	
ดร.เทิดขวัญ ช้างเผือก กรรมการ	
ดร.บุษยมาส พิมพ์พรรณชาติ กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา การพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม
ชื่อนักศึกษา นางสาววรากร บุญปัญญา รหัสนักศึกษา 58050144
ปริญญา วิทยาศาสตรบัณฑิต (คณิตศาสตร์ประยุกต์)
ภาควิชา คณิตศาสตร์
คณะ วิทยาศาสตร์
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา 2561
อาจารย์ที่ปรึกษา ดร.บุษยมาส พิมพ์พรรณชาติ
พนักงานพี่เลี้ยง นายปิติพล มานะวิบูลย์

บทคัดย่อ

คอนโดมิเนียมเป็นอสังหาริมทรัพย์ที่ได้รับความนิยมอย่างต่อเนื่อง ไม่ว่าจะเป็นเพื่อการลงทุนหรือที่อยู่อาศัย ทั้งนี้การซื้อขายคอนโดมิเนียมต้องอาศัยราคาประเมินเพื่อให้ครอบคลุมราคาในทุกๆ ด้าน สหกิจศึกษาจึงได้จัดทำโมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมในจังหวัดกรุงเทพและปริมณฑล โดยทางผู้จัดทำวิเคราะห์ข้อมูลพื้นฐานและปัจจัยต่างๆ ที่ส่งผลกับราคาประเมินของคอนโดมิเนียม โดยใช้โมเดล Random Forest แยกช่วงราคาเป็น 3 ช่วงคือ 1-5 ล้านบาท, 5-10 ล้านบาท และ 10-20 ล้านบาท และใช้โมเดลโครงข่ายประสาทเทียม (Artificial Neural Network) พยากรณ์ราคาสินทรัพย์คอนโดมิเนียม

คำสำคัญ : โครงข่ายประสาทเทียม, สินทรัพย์คอนโดมิเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Forecasting the Price of Condominium Assets
Student	Miss Waragorn Boonpanya Student ID 58050144
Degree	Bachelor of Science (Applied Mathematics)
Department	Mathematics
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2018
Advisor	Dr.Busayamas Pimpunchat
Job advisor	Mr.Pitiphol Manaviboon

Abstract

Condominium real estate assets are popular whether it is for investment or residential. Trading a condominium requires appraisal to cover price in all of the cases. Co-operative education has prepared a model to estimate asset price forecast of condominiums in Bangkok and metropolitan provinces. The basic data analysis vendors and factors that affect estimated price of condominiums was conducted by using a Random Forest model splitting the price into three ranges: 1-5 million baht; 5-10 million and 10-20 million and using an Artificial Neural Network model to forecast asset price of those condominiums.

Keywords : artificial neural network, condominium assets.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

สำหรับการจัดทำสหกิจศึกษาเรื่องการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมจนสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอขอบพระคุณ ดร.บุษยมาส พิมพ์พรรณชาติ เป็นอย่างสูงซึ่งเป็นอาจารย์ที่ปรึกษาในการทำหัวข้อสหกิจศึกษาฉบับนี้ ที่คอยให้คำปรึกษา แนะแนวทางในการดำเนินงานและให้ข้อมูลความรู้ต่าง ๆ ที่ใช้ในการทำหัวข้อสหกิจฉบับนี้ ตลอดจนช่วยตรวจสอบแก้ไขข้อผิดพลาดต่าง ๆ เพื่อให้การดำเนินงานเกิดความถูกต้องครบถ้วน

ขอขอบพระคุณ ผศ.ดร.เดชา สมนะ และ ดร.เทิศขวัญ ช้างเผือก ประธานกรรมการและกรรมการสอบหัวข้อสหกิจศึกษาที่ให้คำแนะนำในการแก้ไขข้อบกพร่องเพื่อให้หัวข้อสหกิจศึกษานี้สมบูรณ์ยิ่งขึ้น

ขอขอบคุณนายปิติพล มานะวิบูลย์ พนักงานพี่เลี้ยงและพี่ๆในทีมทุกคนที่คอยให้คำปรึกษา สอนงานและให้ความรู้ในทุกๆเรื่อง อบรม และดูแลเป็นอย่างดีตลอดการฝึกสหกิจศึกษารวมถึงเพื่อนๆที่ฝึกสหกิจด้วยกันที่ให้คำแนะนำต่างๆ

ขอขอบพระอาจารย์ทุกท่าน ที่ได้ประสิทธิ์ประสาทความรู้ตลอดจนข้อคิดต่างๆอันก่อให้เกิดประโยชน์ต่อการศึกษาค้นคว้า และเป็นแนวทางในการจัดทำหัวข้อสหกิจศึกษานี้จนประสบความสำเร็จ

นอกจากนี้คณะผู้จัดทำขอขอบพระคุณบิดา มารดา ที่ให้กำลังใจและสนับสนุนทางด้านทุนทรัพย์ในการศึกษาตลอดมา จนทำให้การทำหัวข้อสหกิจศึกษานี้สำเร็จด้วยดี รวมถึงเพื่อนๆและพี่ๆทุกคนที่ให้ความช่วยเหลือในด้านต่างๆเกี่ยวกับการทำหัวข้อสหกิจนี้

คุณค่าและประโยชน์ใดๆที่เป็นผลจากหัวข้อสหกิจศึกษานี้ผู้จัดทำขอมอบแด่ บิดา มารดา และครูอาจารย์ทุกท่านด้วยความเคารพยิ่ง

วรากร บุญปัญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ก

บทคัดย่อภาษาอังกฤษ.....ข

กิตติกรรมประกาศ.....ก

สารบัญ.....ง

สารบัญตาราง.....จ

สารบัญรูป.....ฉ

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญของปัญหา.....1

1.2 วัตถุประสงค์.....2

1.3 ขอบเขต.....2

1.4 ประโยชน์ที่คาดว่าจะได้รับ.....2

1.5 ขั้นตอนในการดำเนินงาน.....3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....4

2.1 แบบจำลองโครงข่ายประสาทเทียม (Artificial Neural Network).....4

2.1.1 ความหมายและหลักการของโครงข่ายประสาทเทียม.....5

2.1.2 ลักษณะของโครงข่ายประสาทเทียม.....7

2.1.3 ประเภทของการเรียนรู้ของโครงข่ายประสาทเทียม.....9

2.1.4 การเรียนรู้แบบแพร่ย้อนกลับ.....9

2.1.5 ฟังก์ชันการแปลงค่า.....12

2.1.6 หลักการทำงานของโครงข่ายประสาทเทียม.....14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2.1.7 การปรับค่าโครงข่ายประสาทเทียมที่เหมาะสม.....19

2.1.8 วิธีการประเมินผลลัพธ์จาก Model.....21

บทที่ 3 วิธีดำเนินงานวิจัย.....22

3.1 ขั้นตอนการทำโมเดลเพื่อพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม.....22

3.2 รายระเอียดของการสร้างโมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม.....24

3.3 โมเดลเพื่อแยกช่วงราคาเป็น 3 ช่วง.....25

3.3.1 การทำความเข้าใจในธุรกิจ (Business Understanding) 25

3.3.2 การทำความเข้าใจข้อมูล (Data Understanding) 25

3.3.3 การเตรียมข้อมูล (Data Preparation).....36

3.3.4 การสร้างโมเดล (Modeling).....42

3.4 โมเดลการพยากรณ์ราคาประเมินในช่วงราคา 1–5 ล้านบาท.....43

3.4.1 การทำความเข้าใจข้อมูล (Data Understanding) 43

3.4.2 การเตรียมข้อมูล (Data Preparation).....49

3.4.3 Model Training ในช่วงราคา 1–5 ล้านบาท.....54

3.5 โมเดลการพยากรณ์ราคาประเมินในช่วงราคา 5-10 ล้านบาท.....57

3.5.1 การทำความเข้าใจข้อมูล (Data Understanding).....57

3.5.2 การเตรียมข้อมูล (Data Preparation).....63

3.5.3 Model Training ในช่วงราคา 5-10 ล้านบาท.....67

3.6 โมเดลการพยากรณ์ราคาประเมินในช่วงราคา 10-20 ล้านบาท.....68

3.6.1 การทำความเข้าใจข้อมูล (Data Understanding) 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

3.6.2 การเตรียมข้อมูล (Data Preparation).....74

3.6.3 Model Training ในช่วงราคา 10-20 ล้านบาท.....78

3.7 การนำไปใช้งาน (Deployment).....79

บทที่ 4 ผลการวิจัยและการอภิปรายผล 80

บทที่ 5 สรุปผลวิจัยและข้อเสนอแนะ 87

5.1 สรุปผลการวิจัย.....87

5.2 ข้อเสนอแนะ 87

เอกสารอ้างอิง.....88

ภาคผนวก.....89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

3.1 แสดงเครื่องมือที่ใช้ในการสร้างโมเดล.....24

3.2 ตารางแสดงสเปคของเครื่องคอมพิวเตอร์แม่ข่ายที่ใช้ในการประมวลผลและทดสอบระบบ.....24

3.3 ชื่อคอลัมน์ของข้อมูลเชิงปริมาณ.....30

3.4 ชื่อคอลัมน์ของข้อมูลทั้งหมด.....31

3.5 จำนวนข้อมูลที่ขาดหาย.....35

3.6 แสดงปัจจัยและการเข้ารหัสของตัวแปร.....54

3.7 ตารางแสดง architecture ของโมเดลในช่วงราคา 1-5 ล้านบาท.....56

3.8 ตารางแสดง architecture ของโมเดลในช่วงราคา 5-10 ล้านบาท.....67

3.9 ตารางแสดง architecture ของโมเดลในช่วงราคา 10-20 ล้านบาท.....78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 โครงสร้างระบบประสาท....5
2.2 กระบวนการประมวลผลของโครงข่ายประสาทเทียม....7
2.3 โครงสร้างโครงข่ายประสาทเทียมแบบชั้นเดียว....8
2.4 โครงข่ายประสาทเทียมแบบหลายชั้น....8
2.5 การแพร่เดินหน้า (forward propagation)....10
2.6 โครงข่ายประสาทเทียมแบบแพร่เดินหน้า....11
2.7 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ....11
2.8 ฟังก์ชันกระตุ้น (activation function)....12
2.9 ฟังก์ชันการแปลงเชิงเส้น....12
2.10 ReLU Function....13
2.11 แสดงโครงสร้างการทำงานของ 1 Neuron....14
2.12 แสดงสถาปัตยกรรมของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ....15
2.13 ทิศทางการหาคำตอบของโครงข่ายประสาทเทียม....20
2.14 ลักษณะอัตราความผิดพลาดที่เกิด Peaking effect....20
3.1 วัฏจักรกระบวนการใช้ข้อมูล (Data Process Cycle)....22
3.2 data frame ของข้อมูลเชิงปริมาณ....29
3.3 data frame ของข้อมูลเชิงคุณภาพ....31
3.4 ค่า Correlation Coefficient ของข้อมูลทั้งหมด....34
3.5 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลเมื่อเทียบเป็นเปอร์เซ็นต์....34
3.6 ค่าสถิติเบื้องต้นของข้อมูลทั้งหมด....39
3.7 data frame คอลัมน์ที่มีค่าว่างหรือไม่มีค่า....40
3.8 กราฟ boxplot หลังตัดค่าผิดปกติ....41
3.9 แสดงค่าสถิติเบื้องต้นในช่วงราคา 1-5 ล้านบาท....43
3.10 แสดงการกระจายตัวของราคาในช่วง 1-5 ล้านบาท....43
3.11 แสดงจำนวนห้องอาบน้ำในช่วง 1-5 ล้านบาท....44
3.12 แสดงจำนวนห้องอาบน้ำเทียบกับราคาในช่วง 1-5 ล้านบาท....44
3.13 แสดงจำนวนห้องนอนในช่วง 1-5 ล้านบาท....45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.14 แสดงจำนวนห้องนอนเมื่อเทียบกับราคาในช่วง 1-5 ล้านบาท....45
3.15 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมในช่วง 1-5 ล้านบาท....46
3.16 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมเทียบกับราคาในช่วง 1-5 ล้านบาท....46
3.17 แสดงจำนวนอาคารรวมของคอนโดมิเนียมในช่วง 1-5 ล้านบาท....47
3.18 แสดงจำนวนอาคารรวมของคอนโดมิเนียมเทียบกับราคาในช่วง 1-5 ล้านบาท....47
3.19 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมในช่วง 1-5 ล้านบาท....48
3.20 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมเทียบกับราคาในช่วง 1-5 ล้านบาท....48
3.21 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลในช่วงราคา 1-5 ล้านบาท....49
3.22 คอลัมน์และจำนวนข้อมูลที่หายไปในช่วงราคา 1-5 ล้านบาท....49
3.23 คอลัมน์ที่มีค่า correlation ในช่วงราคา 1-5 ล้านบาท....50
3.24 แสดงแผนภาพคอลัมน์ที่มีค่า correlation ในช่วงราคา 1-5 ล้านบาท....50
3.25 คอลัมน์ที่มีค่าความเบ้ ในช่วงราคา 1-5 ล้านบาท....51
3.26 ก่อนตัดค่าผิดปกติในช่วงราคา 1-5 ล้านบาท....51
3.27 หลังตัดค่าผิดปกติในช่วงราคา 1-5 ล้านบาท....52
3.28 data farm การทำ one-hot encoding....53
3.29 แสดงการทำ Scaling data....53
3.30 การเจอคำตอบ Local ในโครงข่ายประสาทเทียม....55
3.31 train โมเดลในช่วงราคา 1-5 ล้านบาท....56
3.32 แสดงค่าสถิติเบื้องต้นในช่วงราคา 5-10 ล้านบาท....57
3.33 แสดงการกระจายตัวของราคาในช่วง 5-10 ล้านบาท....57
3.34 แสดงจำนวนห้องอาบน้ำในช่วง 5-10 ล้านบาท....58
3.35 แสดงจำนวนห้องนอนเมื่อเทียบกับราคาในช่วง 5-10 ล้านบาท....58
3.36 แสดงจำนวนห้องนอนในช่วง 5-10 ล้านบาท....59
3.37 แสดงจำนวนห้องนอนในช่วง 5-10 ล้านบาท....59
3.38 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมในช่วง 5-10 ล้านบาท....60
3.39 แสดงจำนวนอาคารรวมของคอนโดมิเนียมเทียบกับราคาในช่วง 5-10 ล้านบาท....60
3.40 แสดงจำนวนชั้นของคอนโดมิเนียมในช่วง 5-10 ล้านบาท....61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.41 แสดงจำนวนชั้นของคอนโดมิเนียมเทียบกับราคาในช่วง 5-10 ล้านบาท.....61
3.42 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมเทียบกับราคาในช่วง 5-10 ล้านบาท.....61
3.43 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมเทียบกับราคาในช่วง 5-10 ล้านบาท.....62
3.44 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลในช่วงราคา 5-10 ล้านบาท.....63
3.45 คอลัมน์และจำนวนข้อมูลที่หายไปในช่วงราคา 5-10 ล้านบาท.....63
3.46 คอลัมน์ที่มีค่า correlation ในช่วงราคา 5-10 ล้านบาท.....64
3.47 แสดงแผนภาพคอลัมน์ที่มีค่า correlation ในช่วงราคา 5-10 ล้านบาท.....64
3.48 คอลัมน์ที่มีค่าความเบ้ ในช่วงราคา 5-10 ล้านบาท.....65
3.49 ก่อนตัดค่าผิดปกติในช่วงราคา 5-10 ล้านบาท.....65
3.50 หลังตัดค่าผิดปกติในช่วงราคา 5-10 ล้านบาท.....66
3.51 train โมเดลในช่วงราคา 5-10 ล้านบาท.....67
3.52 แสดงค่าสถิติเบื้องต้นในช่วงราคา 10-20 ล้านบาท.....68
3.53 แสดงการกระจายตัวของราคาในช่วง 10-20 ล้านบาท.....68
3.54 แสดงจำนวนห้องอาบน้ำในช่วง 10-20 ล้านบาท.....69
3.55 แสดงจำนวนห้องอาบน้ำเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท.....69
3.56 แสดงจำนวนห้องนอนในช่วง 10-20 ล้านบาท.....70
3.57 แสดงจำนวนห้องนอนเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท.....70
3.58 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมในช่วงราคา 10-20 ล้านบาท.....71
3.59 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท.....71
3.60 แสดงจำนวนอาคารรวมของคอนโดมิเนียมในช่วง 10-20 ล้านบาท.....72
3.61 แสดงจำนวนอาคารรวมของคอนโดมิเนียมเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท.....72
3.62 แสดงจำนวนปีที่สร้างคอนโดมิเนียมในช่วง 10-20 ล้านบาท.....73
3.63 แสดงจำนวนปีที่สร้างคอนโดมิเนียมเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท.....73
3.64 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลในช่วงราคา10-20 ล้านบาท.....74
3.65 คอลัมน์และจำนวนข้อมูลที่หายไปในช่วงราคา 10-20 ล้านบาท.....74
3.66 คอลัมน์ที่มีค่า correlation ในช่วงราคา 10-20 ล้านบาท.....75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.67 แสดงแผนภาพคอลัมน์ที่มีค่า correlation ในช่วงราคา 10-20 ล้านบาท.....57
3.68 คอลัมน์ที่มีค่าความเบ้ในช่วงราคา 10-20 ล้านบาท.....76
3.69 ก่อนตัดค่าผิดปกติในช่วงราคา 10-20 ล้านบาท.....76
3.70 หลังตัดค่าผิดปกติในช่วงราคา 10-20 ล้านบาท.....77
3.71 train โมเดลในช่วงราคา 10-20 ล้านบาท.....78
4.1 แสดงค่า loss และ val loss ของโมเดลในช่วงราคา 1-5 ล้านบาท.....81
4.2 ข้อมูลเปรียบเทียบระหว่างค่าจริงและค่าทำนายในช่วงราคา 1-5 ล้านบาท.....82
4.3 แสดงค่า loss และ val loss ของโมเดลในช่วงราคา 5-10 ล้านบาท.....83
4.4 ข้อมูลเปรียบเทียบระหว่างค่าจริงและค่าทำนายในช่วงราคา 5-10 ล้านบาท.....84
4.5 แสดงค่า loss และ val loss ของโมเดลในช่วงราคา 10-20 ล้านบาท.....85
4.6 ข้อมูลเปรียบเทียบระหว่างค่าจริงและค่าทำนายในช่วงราคา 5-10 ล้านบาท.....86
4.7 แสดงค่าความถูกต้องของโมเดลการพยากรณ์เพื่อแบ่งช่วงราคา.....86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

คอนโดมิเนียมเป็นหนึ่งในประเภทของอสังหาริมทรัพย์ที่มีความต้องการอย่างต่อเนื่อง เนื่องด้วยอยู่ใกล้อาคารสำนักงาน หรือจุดศูนย์กลางทางธุรกิจ อีกทั้งยังสะดวกต่อการเดินทาง สามารถอาศัยอยู่ในพื้นทีกลางเมืองที่สามารถการเข้าถึงที่ทำงานโดยง่ายใกล้ และใช้เวลาไม่มาก คอนโดมิเนียมจึงตอบโจทย์กับปัญหาเหล่านั้นได้อย่างดี การซื้อคอนโดมิเนียมเป็นของตัวเองจึงได้ประโยชน์อย่างมากเพื่อการลงทุน หรือเป็นที่อยู่อาศัย

ปัจจุบันราคาประเมินจะขึ้นอยู่กับปัจจัยต่างๆ ไม่ว่าจะเป็นลักษณะของแปลงที่ดิน, ขนาดและการออกแบบทางสถาปัตยกรรมของห้องคอนโดมิเนียม, การเข้าถึงสาธารณูปโภค-สิ่งอำนวยความสะดวก และประโยชน์ใช้สอยต่างๆของคอนโดมิเนียม ดังนั้นการประเมินห้องในลักษณะเดียวกัน ทำเลที่ตั้งเหมือนกัน ราคาประเมินของผู้ประเมินแต่ละบุคคลอาจจะแตกต่างกันได้ เนื่องจากประสบการณ์และความชอบของแต่และบุคคล นอกจากนี้ยังช่วยแก้ปัญหาจากการประเมินราคาสินทรัพย์ที่เกินหรือต่ำกว่าความเป็นจริง ความล่าช้าของการประเมินราคา และค่าใช้จ่ายที่สูงของการลงพื้นที่ในการประเมินราคาสินทรัพย์

ด้วยสาเหตุนี้จึงมีการทำโมเดลนี้เพื่อพยากรณ์ราคาประเมินคอนโดมิเนียม โดยอาศัยข้อมูลพื้นฐานและปัจจัยต่างๆ ที่ส่งผลกับราคาประเมินของคอนโดมิเนียม เพื่อแก้ไขปัญหาความล่าช้าในการสำรวจและประเมินหลักทรัพย์ ลดค่าใช้จ่ายในส่วนนี้บริษัทประเมิน โดยการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมยังเพิ่มความแม่นยำ เที่ยงตรง และมีความน่าเชื่อถือ อีกทั้งยังสามารถใช้เป็นฐานการพิจารณาวงเงินกู้ของธนาคาร ตลอดจนการซื้อ-ขาย โดยอิงจากราคาประเมินสินทรัพย์ที่เราพยากรณ์ได้ ทางผู้จัดทำจึงมีความประสงค์ที่จะจัดสร้างโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1) เพื่อศึกษาปัจจัยที่มีผลต่อราคาประเมินสินทรัพย์คอนโดมิเนียม

2) เพื่อสร้างเครื่องมือในการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมที่มีความแม่นยำ

1.3 ขอบเขต

1) ข้อมูลที่ศึกษานำมาจากเว็บไซต์อสังหาริมทรัพย์

2) ใช้ภาษาไพทอน (python) ในการสร้างโมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม

3) ศึกษาราคาประเมินสินทรัพย์คอนโดมิเนียมในช่วงราคา 1-20 ล้านบาท

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ศึกษาปัจจัยที่มีผลต่อราคาประเมินสินทรัพย์คอนโดมิเนียม

2) ได้ศึกษาและสามารถดำเนินการสร้างโมเดลเพื่อเป็นเครื่องมือในการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนในการดำเนินงาน

กิจกรรมการดำเนินงาน	ระยะเวลาในการดำเนินงาน						
	ปี 2561	ปี 2562					
	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.
1.กำหนดหัวข้อ วัตถุประสงค์ ความสำคัญของงาน และขอบเขตของโครงงาน							
2.ศึกษาหาข้อมูลและทฤษฎีที่เกี่ยวข้องกับโครงงาน							
3.สำรวจข้อมูลด้วยกราฟต่างๆ (Data Exploration)							
4.เตรียมข้อมูล (Data processing)							
5.ตรวจสอบความถูกต้องของข้อมูล (Data validation)							
6.เลือกโมเดลที่ใช้ในการพยากรณ์							
7.ปรับปรุงโมเดลให้มีความแม่นยำมากขึ้น (Model evaluation)							
8.วิเคราะห์และสรุปผลการทดสอบ							
9.ทดสอบโมเดลกับข้อมูลจริง							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

โครงข่ายประสาทเทียมเป็นโมเดลทางคณิตศาสตร์ที่จำลองการทำงานของระบบประสาทซึ่งประกอบด้วยหน่วยปฏิบัติการพื้นฐานหลายหน่วยมาเชื่อมต่อกัน มีการส่งค่าในแต่ละหน่วยเพื่อใช้ในการคำนวณผลลัพธ์โครงข่ายประสาทเทียมเป็นระบบที่มีการปรับเปลี่ยนตัวเองในระหว่างการเรียนรู้ นิยมใช้ในการหาแบบจำลองของความสัมพันธ์ที่ซับซ้อนระหว่างตัวแปรต้นและตัวแปรตามในด้านการพยากรณ์มีการนำโครงข่ายประสาทเทียมมาใช้กันอย่างแพร่หลาย เพื่อการหารูปแบบความสัมพันธ์ระหว่างตัวแปรต้นและตัวแปรตาม การทำนายค่าของผลลัพธ์นั้นใกล้เคียงกับผลลัพธ์ที่ถูกต้อง ซึ่งล้วนแล้วแต่ให้ค่าความแม่นยำที่ดี

2.1 โมเดลโครงข่ายประสาทเทียม (Artificial Neural Network)

โครงข่ายประสาทเทียม (Artificial Neuron Network : ANN) เป็นศาสตร์แขนงหนึ่งของทางด้านปัญญาประดิษฐ์ (Artificial Intelligence : AI) มีรูปแบบโครงสร้างและการทำงานของการประมวลผลเหมือนกับสมองของสิ่งมีชีวิตซึ่งมีปรับเปลี่ยนตัวเองต่อการตอบสนองของอินพุตตามกฎของการเรียนรู้ (learning rule) หลังจากที่โครงข่ายได้เรียนรู้สิ่งที่ต้องการแล้ว โครงข่ายนั้นจะสามารถทำงานที่กำหนดไว้ได้โครงข่ายประสาทเทียมได้ถูกพัฒนาคิดค้นจากการทำงานของสมองมนุษย์โดยสมองมนุษย์ประกอบไปด้วยหน่วยประมวลผลเรียกว่า นิวรอน (เซลล์ประสาท หรือ neuron) จำนวนนิวรอนในสมองมนุษย์มีอยู่ประมาณและมีการเชื่อมต่อกันอย่างมากมาย สมองมนุษย์จึงสามารถกล่าวได้ว่าเป็นคอมพิวเตอร์ที่มีการปรับตัวเอง (adaptive) ไม่เป็นเชิงเส้น (nonlinear) และทำงานแบบขนาน (parallel) ในการดูแลจัดการการทำงานร่วมกันของนิวรอนในสมอง การคำนวณเชิงนิวรอนเป็นการคำนวณที่เลียนแบบมาจากการทำงานของสมองมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ความหมายและหลักการของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม คือ โมเดลทางคณิตศาสตร์ที่พัฒนาขึ้นเพื่อจำลองการทำงานของโครงข่ายประสาทในสมองมนุษย์ โดยที่โครงข่ายประสาทของมนุษย์ประกอบด้วยเซลล์ประสาท (Neuron) และจุดประสานประสาทหรือไซแนปส์ (Synapses) โดยโครงสร้างของการส่งสัญญาณประสาทประกอบจากการเชื่อมต่อระหว่างเซลล์ ประสาทหลายพันล้านเซลล์ เซลล์ประสาทแต่ละเซลล์ประกอบด้วยแขนงรับสัญญาณประสาทซึ่งเป็น เสมือนหน่วยรับข้อมูลป้อนเข้า เรียกว่า “เดนไดรท์” (dendrites) และส่วนปลายของเซลล์ประสาทใน การส่งสัญญาณประสาทซึ่งเป็นเสมือนหน่วย ส่งข้อมูลออกของเซลล์ เรียกว่า “แอคซอน” (axon) โดยการส่งสัญญาณประสาทดังกล่าวอาจทำให้ เกิดได้ทั้งการกระตุ้นและยับยั้ง ทั้งนี้นอกจากลักษณะ ดังกล่าวแล้ว วิธีการประมวลผลภายในเซลล์ ประสาทแต่ละเซลล์ยังมีการขยายหรือลดขนาด ของสัญญาณอีกด้วย โดยสัญญาณจากเดนไดรท์ ต่าง ๆ จะรวมกันเข้าสู่เซลล์ประสาท และหาก สัญญาณรวมมีความแรงเกินค่าระดับ (threshold) ของเซลล์ประสาทนั้นๆ เซลล์ประสาทก็จะส่ง สัญญาณออกทางแอคซอนต่อไป กระบวนการเรียนรู้ในสิ่งมีชีวิตจะมีผลให้ เกิดการสร้างไซแนปส์ระหว่างเซลล์ประสาทขึ้นมา ใหม่ หรือทำให้เกิดการเปลี่ยนสภาพของไซแนปส์ ต่างๆ ในโครงข่ายของเซลล์ประสาท โครงข่าย ประสาทของสิ่งมีชีวิตจึงไม่ได้ทำงานแบบเป็นลำดับ ขั้นตอน (sequential) แต่เพียงอย่างเดียวลักษณะโดยทั่วไปของโครงสร้างระบบประสาทแสดงดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างระบบประสาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายประสาทเทียมมีคุณลักษณะคล้าย กับการส่งผ่านสัญญาณประสาทในสมองของมนุษย์ กล่าวคือ มีความสามารถในการรวบรวมความรู้ (knowledge) โดยผ่านกระบวนการเรียนรู้ (learning process) และความรู้เหล่านี้จะจัดเก็บอยู่ใน โครงข่ายในรูปแบบค่าน้ำหนัก (weight) ซึ่งสามารถ ปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้สิ่งใหม่ๆ เข้าไป ค่าน้ำหนักทำหน้าที่เปรียบเสมือนความรู้ที่รวบรวม ไว้เพื่อใช้ในการแก้ปัญหาเฉพาะอย่างของมนุษย์

การประมวลผลต่างๆ เกิดขึ้นในหน่วย ประมวลผลย่อย เรียกว่า โหนด (node) ซึ่งโหนดเป็นการจำลองลักษณะการทำงานมาจากเซลล์ การส่งสัญญาณ (signal) ระหว่างโหนดที่เชื่อมต่อกัน (connection) จำลองมาจากการเชื่อมต่อ ของเดนไดร์ทและแอคซอนในระบบประสาทของ มนุษย์ ภายในโหนดจะมีฟังก์ชันกำหนดสัญญาณ ส่งออกที่เรียกว่า ฟังก์ชันกระตุ้น (activation function) หรือ ฟังก์ชันการแปลง (transfer function) ซึ่งทำหน้าที่เปรียบเสมือนกระบวนการ ทำงานในเซลล์

โครงข่ายประสาทเทียมประกอบด้วย 5 องค์ประกอบ ดังนี้

1.ข้อมูลป้อนเข้า (input) เป็นข้อมูล ที่เป็นตัวเลข หากเป็นข้อมูลเชิงคุณภาพต้องแปลงให้อยู่ในรูปเชิงปริมาณที่โครงข่ายประสาท เทียมยอมรับได้

2.ข้อมูลส่งออก (output) คือ ผลลัพธ์ ที่เกิดขึ้นจริง (actual output) จากกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม

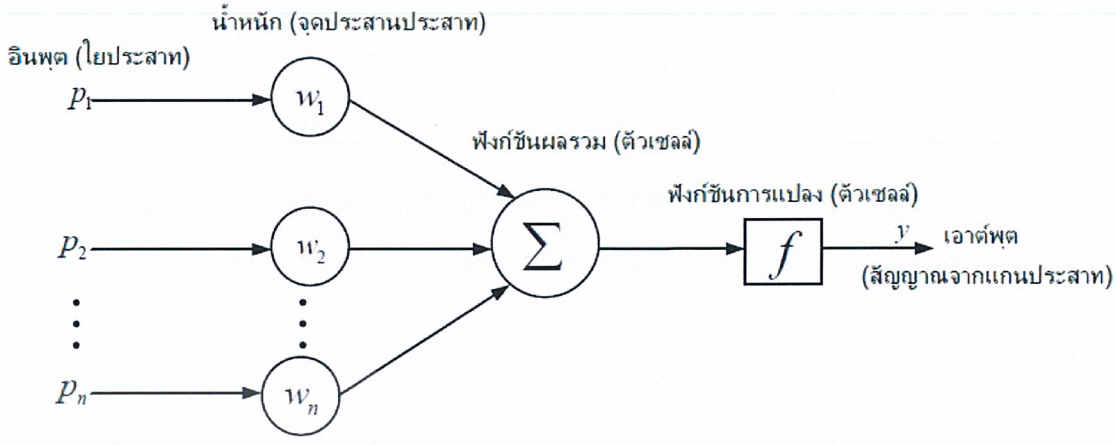
3. ค่าน้ำหนัก (weights) คือ สิ่งที่ได้จาก การเรียนรู้ของโครงข่ายประสาทเทียม หรือเรียกอีกอย่างหนึ่งว่า ค่าความรู้ (knowledge) ค่านี้จะ ถูกเก็บเป็นทักษะเพื่อใช้ในการจดจำข้อมูลอื่นๆ ที่อยู่ในรูปแบบเดียวกัน

4. ฟังก์ชันผลรวม (Summation function : S) เป็นผลรวมของข้อมูลป้อนเข้า (a_i) และค่าน้ำหนัก (w_i)

$$s = \sum_{i=1}^{n} a_i w_i$$

5. ฟังก์ชันการแปลง (transfer function) เป็นการคำนวณการจำลองการทำงานของ โครงข่ายประสาทเทียม เช่น ซิกมอยด์ฟังก์ชัน (sigmoid function) ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์ (hyperbolic tangent function) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



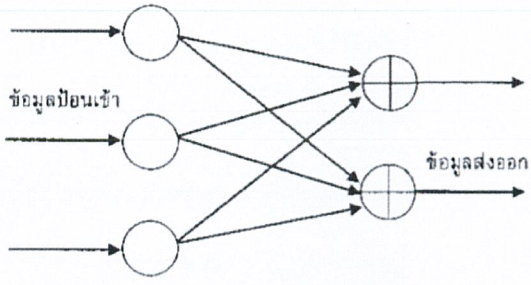
รูปที่ 2.2 กระบวนการประมวลผลของโครงข่ายประสาทเทียม

2.1.2 ลักษณะของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมประกอบด้วยเซลล์ ประสาทเทียม หรือโหนดจำนวนมากเชื่อมต่อกัน ซึ่งการเชื่อมต่อแบ่งออกเป็นกลุ่มย่อย เรียกว่า ชั้น (layer) ชั้นแรก เป็นชั้นนำข้อมูลเข้า เรียกว่า ชั้นรับข้อมูลป้อนเข้า (input layer) ส่วนชั้นสุดท้ายเรียกว่า ชั้นส่งข้อมูลออก (output layer) และชั้นที่อยู่ระหว่างชั้นรับข้อมูลป้อนเข้าและชั้นส่งข้อมูลออก เรียกว่า ชั้นแอบแฝง (hidden layer) ซึ่งโดยทั่วไปชั้นแอบแฝงอาจมีมากกว่า 1 ชั้นก็ได้ ด้วยเหตุนี้จึงสามารถแบ่งประเภทของโครงข่ายประสาทเทียมตามจำนวนชั้นของโครงข่ายแบบกว้างๆ ได้ 2 แบบ ได้แก่ โครงข่ายแบบชั้นเดียว (single layer) และโครงข่ายแบบหลายชั้น (multilayer)

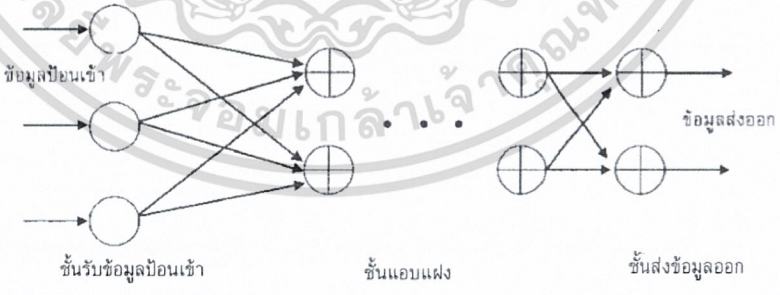
1.โครงข่ายแบบชั้นเดียว โครงข่ายแบบชั้นเดียว เป็นโครงข่าย ประสาทเทียมอย่างง่ายที่มีเพียงชั้นรับข้อมูลป้อนเข้า และชั้นส่งข้อมูลออกเท่านั้น โหนดในชั้นรับข้อมูล ป้อนเข้าทำหน้าที่รับข้อมูลเข้า (input value) แล้ว ส่งข้อมูลผ่านเส้นเชื่อมโยงต่างๆ ไปให้โหนดในชั้นส่งข้อมูลออก ความเข้มของสัญญาณ หรือ ปริมาณข้อมูลที่นำเข้าสู่โหนดในชั้นส่งข้อมูลออก จะขึ้นอยู่กับค่าน้ำหนักที่อยู่บนเส้นเชื่อมโยง โหนดในชั้นส่งข้อมูลออกจะนำข้อมูลที่ได้ รับมาคำนวณโดยใช้ฟังก์ชันทางคณิตศาสตร์ที่เรียกว่า ฟังก์ชันการแปลง (transfer function) ที่เหมาะสมกับปัญหา แล้วส่งผลลัพธ์ที่ได้ออกมาเป็นข้อมูลส่งออก เช่น โครงข่ายแบบชั้นเดียวแบบ เพอเซบตรอนอย่างง่าย (simple perceptron) และโครงข่ายโฮบฟิลด์ (Hopfield networks) ลักษณะโครงข่ายแบบชั้นเดียวแสดงดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โครงสร้างโครงข่ายประสาทเทียมแบบชั้นเดียว

2.โครงข่ายแบบหลายชั้น โครงข่ายแบบหลายชั้น เป็นโครงข่าย ที่มีชั้นแอบแฝงตั้งแต่ 1 ชั้นขึ้นไป โครงข่ายแบบ หลายชั้นจะใช้ในกรณีที่ปัญหามีความซับซ้อนซึ่งโครงข่ายแบบชั้นเดียวไม่สามารถแก้ปัญหาได้ จึงเพิ่มจำนวนโหนดที่มีการคำนวณ หรือชั้นแอบแฝง โครงข่ายแบบหลายชั้นเป็นโครงข่ายที่มีชั้นแอบแฝงตั้งแต่ 1 ชั้นขึ้นไป โครงข่ายแบบ หลายชั้นจะใช้ในกรณีที่ปัญหามีความซับซ้อน ซึ่ง โครงข่ายแบบชั้นเดียวไม่สามารถแก้ปัญหาได้ จึง เพิ่มจำนวนโหนดที่มีการคำนวณ หรือชั้นแอบแฝงให้กับโครงข่าย ตัวอย่างของโครงข่ายแบบหลายชั้น เช่น การแพร่ย้อนกลับ (back propagation) เซลฟ์ ออร์ แกนไนซิ่งแมปซ์ (self-organizing maps) และเคาน์เตอร์ พรอพะเกชัน (counter propagation) เป็นต้น ลักษณะโครงสร้างโครงข่ายแบบ หลายชั้นแสดงดังรูปที่ 2.4



รูปที่ 2.4 โครงข่ายประสาทเทียมแบบหลายชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ประเภทของการเรียนรู้ของโครงข่ายประสาทเทียม

1.การเรียนรู้แบบมีผู้สอน (supervised learning) ข้อมูลจะประกอบด้วยตัวอย่าง ข้อมูลที่ต้องการสอน และผลลัพธ์ที่ต้องการให้ โครงข่ายสร้าง เมื่อมีการนำข้อมูลในลักษณะเดียวกัน มาเป็นข้อมูลป้อนเข้า โครงข่ายจะกำหนดค่าผลลัพธ์ ที่เป็นเป้าหมายให้กับข้อมูลป้อนเข้าแต่ละตัว โครงข่ายจะนำค่าผิดพลาดระหว่างค่าเป้าหมายกับค่าผลลัพธ์ที่ได้ มาใช้ในการปรับค่าน้ำหนัก เพื่อให้ค่าผลลัพธ์ที่ใกล้เคียงกับเป้าหมายมากที่สุด ถ้าหาก เปรียบเทียบกับมนุษย์จะเหมือนกับการสอนนักเรียน โดยมีครูผู้สอนคอยให้คำแนะนำ ตัวอย่างแบบจำลองนี้ได้แก่ การแพร่ย้อนกลับ และเพอเซบตรอน (perceptron) เป็นต้น

2.การเรียนรู้แบบไม่มีผู้สอน (unsupervised learning) การเรียนรู้แบบนี้จะสอน โครงข่ายโดยการนำข้อมูลป้อนเข้าอย่างต่อเนื่อง เพียงอย่างเดียว ไม่มีการส่งค่าผลลัพธ์เป้าหมาย ให้กับข้อมูลป้อนเข้าแต่ละตัว การปรับน้ำหนักจะใช้ ข้อมูลที่นำมาสอนเป็นตัวปรับค่า โดยค่าน้ำหนักจะ ปรับตามกลุ่มที่ข้อมูลป้อนเข้าที่มีรูปแบบคล้ายคลึงกัน ถ้าหากเปรียบเทียบกับมนุษย์จะเหมือนกับการที่ เราสามารถแยกแยะพันธุ์พืช พันธุ์สัตว์ ตามลักษณะ รูปร่างของมันได้ด้วยตนเอง ตัวอย่างแบบจำลอง นี้ได้แก่ เคาน์เตอร์พรอพะเกชัน (counter propagation: CPN) แบบจำลองอะแดปทีฟรีโซแนนซ์เทียรี่ (Adaptive Resonance Theory neural networks : ART) เป็นต้น

2.1.4 การเรียนรู้แบบแพร่ย้อนกลับ

Werbos (1974) ได้เสนอแนวคิดของ การเรียนรู้แบบแพร่ย้อนกลับ จากนั้น Parker (1982) ได้นำเสนออีกครั้ง และเป็นที่รู้จักมากยิ่งขึ้น โดย Rumelhart และคณะ (1986) ในหนังสือ Parallel Distributed Processing ซึ่งได้กล่าว ถึงศักยภาพของโครงข่ายประสาทเทียม และ การเรียนรู้แบบแพร่ย้อนกลับ การเรียนรู้แบบแพร่ย้อนกลับ สามารถ แก้ปัญหาที่ต้องการรูปแบบ โดยการป้อนรูปแบบ เข้าไป โครงข่ายประสาทเทียมจะให้รูปแบบผลลัพธ์ ที่เกี่ยวข้องกัน (Dayhoff. 1990) การเรียนรู้แบบ แพร่ย้อนกลับเป็นวิธีการหนึ่งของโครงข่ายประสาท เทียมที่ง่ายต่อการเข้าใจ เนื่องจากกระบวนการ เรียนรู้ และปรับปรุงแก้ไขนั้นเป็นไปด้วยตนเอง ถ้าโครงข่ายประสาทเทียมให้คำตอบที่ผิด ค่าน้ำหนัก จะถูกปรับจนกว่าค่าความผิดพลาดจะน้อยลง หรือ อยู่ในเกณฑ์ที่ยอมรับได้ นั่นคือ ค่าที่ได้ในครั้งถัดไป จะมีความถูกต้องมากยิ่งขึ้น ด้วยโครงสร้างประสาท เทียมที่มีลักษณะเป็นชั้น แต่ละชั้นเชื่อมโยงกัน อย่างทั่วถึงเมื่อโครงข่ายประสาทเทียมได้รับข้อมูล ป้อนเข้า จะคำนวณค่าน้ำหนักของหน่วยรับข้อมูล ป้อนเข้าไปยังชั้นแอบแฝง และจากชั้นแอบแฝง ไปยังชั้นส่งข้อมูลออก เมื่อเกิดผลต่างระหว่างค่าผลลัพธ์จริงกับค่าผลลัพธ์เป้าหมายโครงข่ายประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะปรับค่าความผิดพลาดจากชั้นส่ง ข้อมูลออก และแพร่ย้อนกลับไปยังชั้นแอบแฝง จากนั้นจึงแพร่ย้อนกลับไปยังชั้นรับข้อมูลป้อนเข้า ตามลำดับ

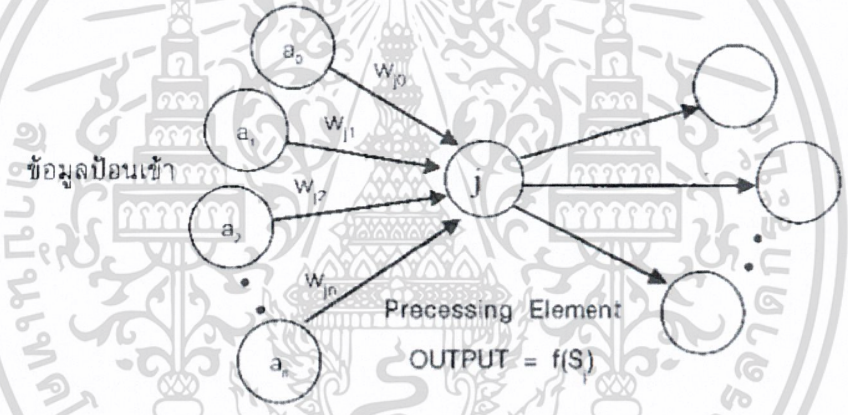
ขั้นตอนการเรียนรู้แบบแพร่ย้อนกลับ ประกอบด้วย 2 ขั้นตอน ดังนี้

1. การแพร่เดินหน้า (forward propagation) ขั้นตอนนี้เริ่มเมื่อเซลล์ประสาทได้รับข้อมูลป้อนเข้า แสดงดังรูปที่ 2.5 ซึ่งเป็นการคำนวณ ผลรวม ของผลลัพธ์ที่เข้ามายังหน่วยที่ j ดังสมการ

$$s = \sum_{i=1}^{n} a_i w_i$$

โดยที่ a_i = ข้อมูลจากหน่วยที่ i

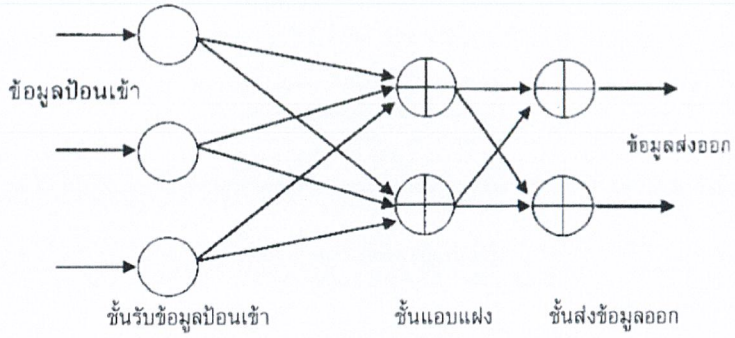
w_{ij} = ค่าน้ำหนักจากหน่วยที่ i ไปยังหน่วยที่ j



รูปที่ 2.5 การแพร่เดินหน้า (forward propagation)

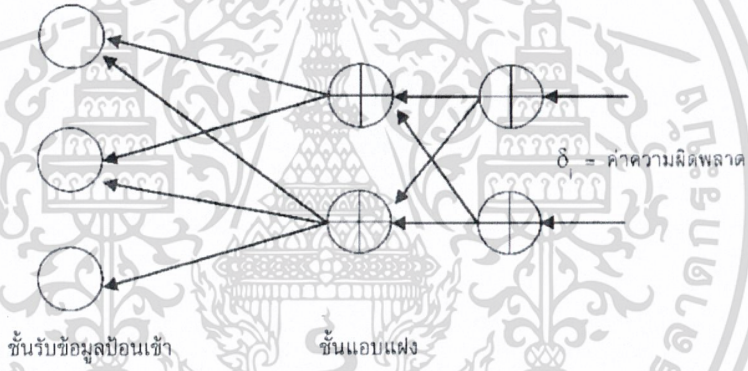
จากนั้นทำการแปลงค่าข้อมูลโดยการคำนวณค่าผลลัพธ์ f(S_j) โดยใช้ ฟังก์ชันการแปลง (transfer function) สำหรับชั้นรับข้อมูลป้อนเข้า จะไม่มีการ ประมวลผลเกิดขึ้น และไม่มีการแปลงค่าข้อมูล เนื่องจากแต่ละหน่วยประมวลผลจะใช้ค่าของข้อมูล ป้อนเข้าเป็นค่าของตนเอง โครงข่ายประสาทเทียมแบบการแพร่เดินหน้าแสดงดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 โครงข่ายประสาทเทียมแบบแพร่เดินหน้า

2. การแพร่ย้อนกลับ (back propagation) ขั้นตอนนี้ ค่าความผิดพลาด (δ) จะถูก คำนวณที่หน่วยประมวลผลทั้งหมดและคำนวณค่าน้ำหนักที่เปลี่ยนของทุกการเชื่อมโยง การ คำนวณนี้เริ่มที่ชั้นส่งข้อมูลออก และแพร่ย้อน กลับไปยังชั้นรับข้อมูลป้อนเข้า แสดงดังรูปที่ 2.7



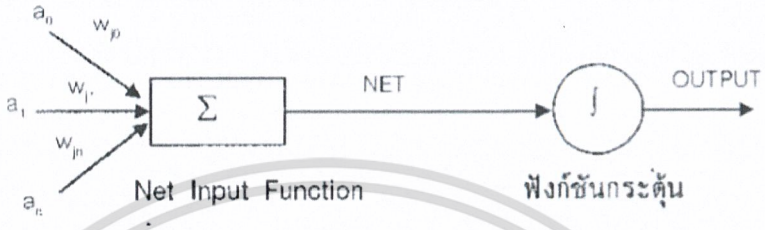
รูปที่ 2.7 โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

ขั้นปรับปรุงความผิดพลาด เกิดหลังจาก ที่ดำเนินการแพร่เดินหน้าแล้ว หน่วยประมวลผล แต่ละหน่วยในชั้นส่งข้อมูลออกจะให้ค่าผลลัพธ์ที่ ต้องนำมาเปรียบเทียบกับค่าผลลัพธ์เป้าหมายในชุดการสอนความแตกต่างนั้นคือ ค่าความผิดพลาด สำหรับแต่ละหน่วยในชั้นส่งข้อมูลออก ซึ่งค่าน้ำหนักของทุกการเชื่อมโยงไปยังชั้นส่งข้อมูลออกจะถูกปรับ จากนั้นค่าความผิดพลาดของหน่วยในชั้นแอบแฝง จะถูกปรับ เช่นกันกระบวนการนี้จะดำเนินไปจนกระทั่งจะถึงชั้นรับข้อมูลป้อนเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 ฟังก์ชันการแปลงค่า

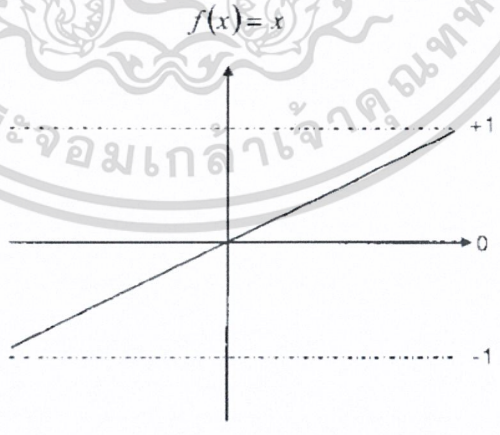
โครงข่ายประสาทเทียมโดยทั่วไป ประกอบด้วย 3 ชั้น คือ ชั้นนำข้อมูลป้อนเข้าชั้นแอบแฝง และชั้นส่งข้อมูลออก โดยชั้นแอบแฝง จะมีการใช้ฟังก์ชันการแปลง หรือฟังก์ชันกระตุ้น (activation function) แสดง ดังรูปที่ 2.8



รูปที่ 2.8 ฟังก์ชันกระตุ้น (activation function)

โดยปกติฟังก์ชันการแปลง (transfer function) สามารถแบ่งได้ 2 ประเภท คือ

1.ฟังก์ชันการแปลงเชิงเส้น (Linear Transfer Function) ฟังก์ชันการแปลงเชิงเส้น สามารถเรียนรู้เพียงความสัมพันธ์เชิงเส้นระหว่างข้อมูล ป้อนเข้าและข้อมูลส่งออก ดังนั้น จึงไม่สามารถหา คำตอบได้สำหรับบางกรณี อย่างไรก็ตาม ถ้าคำตอบ ที่ได้ไม่ใช่คำตอบที่ดีพอ ฟังก์ชันการแปลงเชิงเส้น จะหาค่าต่ำสุดของผลรวมค่าผิดพลาดกำลังสอง แต่ ถ้าอัตราการเรียนรู้มีค่าน้อย โครงข่ายประสาทเทียม จะหาคำตอบที่ใกล้เคียงเท่าที่จะเป็นไปได้ที่แสดง ลักษณะเชิงเส้นของลักษณะของโครงข่าย ฟังก์ชันการแปลงเชิงเส้น สามารถเขียน เป็นสมการทางคณิตศาสตร์และแสดงได้ดังรูป ที่ 2.9



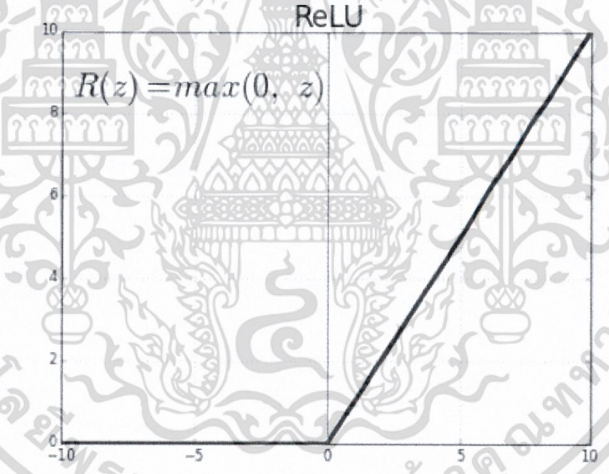
รูปที่ 2.9 ฟังก์ชันการแปลงเชิงเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ฟังก์ชันการแปลงไม่ใช่เชิงเส้น (Nonlinear Transfer Function)

2.1 ReLU Function

จะบีบช่วงข้อมูลป้อนเข้า ที่ไม่จำกัดให้เป็นช่วงของข้อมูลส่งออกที่จำกัด โดยที่ ช่วงของข้อมูลส่งออกจะอยู่ในช่วงศูนย์ถึงอนันต์ ฟังก์ชันการแปลงแบบ ReLU จะแสดงลักษณะของข้อเท็จจริง ที่มีความชันเข้าใกล้ศูนย์เมื่อข้อมูลป้อนเข้ามีจำนวนมาก ขั้นตอนวิธีการฝึกฝนแบบแพร่ย้อนกลับมีวัตถุประสงค์เพื่อลดผลกระทบที่ส่งผลต่อขนาดของ อนุพันธ์ไม่ให้มีผลกระทบต่อค่าน้ำหนักปัจจุบันโดย ขนาดของการเปลี่ยนแปลงค่าน้ำหนักถูกกำหนดโดย ค่าปัจจุบันที่แยกออกมา ค่าปัจจุบันสำหรับค่าน้ำหนัก และความโน้มเอียงแต่ละค่าจะเพิ่มขึ้นเมื่ออนุพันธ์ ของฟังก์ชันสัมพันธ์กับค่าน้ำหนักที่มีเครื่องหมายเดียวกันสำหรับการกระทำซ้ำรอบสอง และค่าปัจจุบัน จะลดลงเมื่ออนุพันธ์ที่สัมพันธ์กับน้ำหนักเปลี่ยนแปลง เครื่องหมายจากการกระทำซ้ำรอบก่อน ถ้าอนุพันธ์เท่ากับศูนย์แสดงว่าค่าปัจจุบันยังคงเดิม

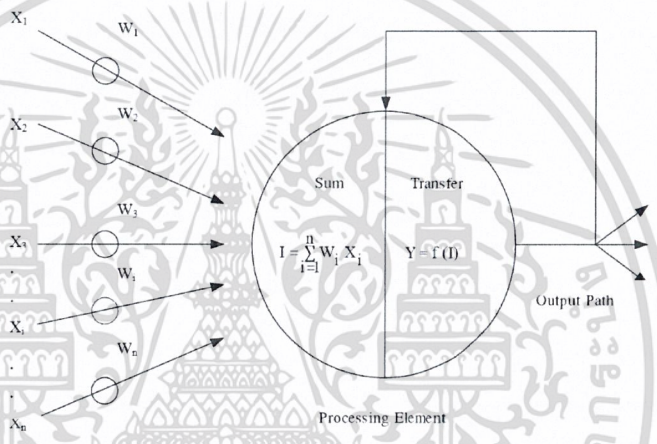


รูปที่ 2.10 ReLU Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 หลักการทำงานของโครงข่ายประสาทเทียม

Artificial Neurons คือหน่วยพื้นฐานของ ANNs ซึ่งจำลองการทำงานของฟังก์ชัน ทั้ง 4 ใน Neurons ของสมองมนุษย์ อย่างไรก็ตาม Artificial Neurons มีโครงสร้างการทำงานที่ ง่ายกว่า Neurons ในสมองมนุษย์มาก รูปที่ 2.11 แสดงโครงสร้างการทำงานของ 1 Neuron ถ้ากำหนดให้ข้อมูลป้อนเข้าสู่ Artificial Neurons คือ $x_1,\ x_2, \ldots, x_n$ ข้อมูลแต่ละตัวจะถูกคูณ ด้วย Weight $w_1,\ w_2, \ldots, w_n$ ตามลำดับ Artificial Neurons จะนำข้อมูลทั้ง n มารวมกัน ซึ่งเรียกว่า Activation ก่อนที่จะแปลง (Transfer) ข้อมูลเป็นผลลัพธ์แล้วส่งให้ Neurons ตัวอื่นนำไปประมวลผลในลักษณะเดียวกัน เพื่อได้ผลลัพธ์ในขั้นตอนสุดท้าย



รูปที่ 2.11 แสดงโครงสร้างการทำงานของ 1 Neuron

วิธีการประมวลผลเบื้องต้นของ Artificial Neurons โดยการนำข้อมูลทั้งหมดมา รวมกัน (Activation) ซึ่งสามารถแสดงในรูปของการหาผลบวก (Summation) ดังสมการ

$$I = \sum x_i * w_i$$

หลังจากนั้น Artificial Neurons จะแปลงค่าผลบวก หรือ Activation (I) เป็น ผลลัพธ์ (Y) โดยใช้ฟังค์ชั่นทางคณิตศาสตร์ที่ไม่ยุ่งยากซับซ้อนมากนัก เช่น Logistic, Linear, Hyperbolic Tangent, Sine หรือ Gaussian เป็นต้น

$$Y = f(I)$$

ฟังก์ชัน $f(I)$ คือ Transfer Function หรือ Response Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นสุดท้าย Artificial Neurons ในส่วนที่ผลิตผลลัพทธ์ (Output Layer) จะนำ Y ที่ได้จาก Neurons ต่าง ๆ มาประมวลเป็นผลลัพทธ์ในขั้นสุดท้าย (Z) ในทำนองเดียวกับ Neurons ตัว อื่นๆ คือการ Weight การหาผลบวก (Summation) และแปลงค่า (Transfer)

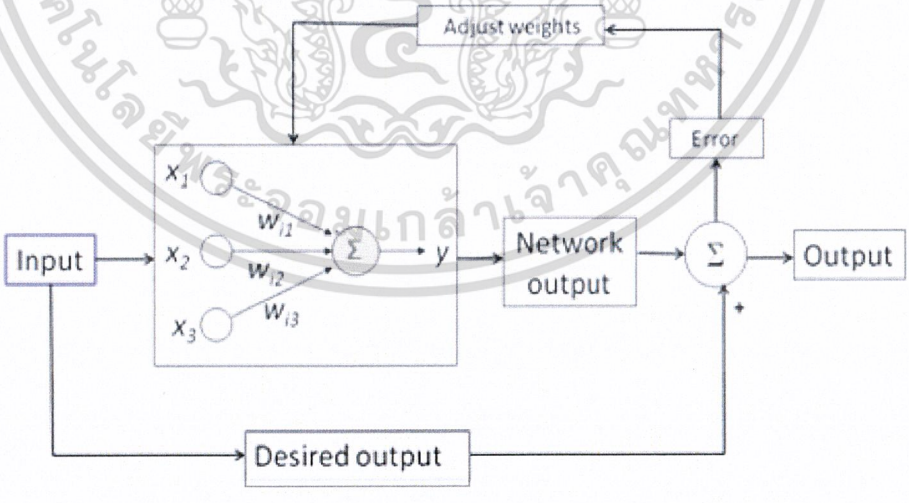
งานวิจัยนี้ได้เลือกใช้อัลกอริทึมของ โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (back propagation algorithm) ในการสร้างแบบจำลอง เนื่องจากข้อมูลที่ใช้สำหรับเป็นปัจจัยนำข้าเป็นข้อมูลย้อนหลังที่ต้องอาศัยการเรียนรู้หลายๆ รอบเพื่อให้ได้ข้อมูลผลลัพธ์ที่มีความน่าเชื่อถือ

สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับประกอบด้วยชั้นของข้อมูลเบื้องต้น 3 ชั้น ได้แก่ ชั้นอินพุต (input layer) ชั้นซ่อน (hidden layer) และชั้นเอาท์พุต (output layer) และ การปรับสอนโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ ประกอบด้วย 3 ขั้นตอน คือ

1) ขั้นตอนการคำนวณไปข้างหน้า (feed forward) จากชั้นอินพุตไปยังชั้นซ่อน และไปสู่ชั้นเอาท์พุต

2) ขั้นตอนการคำนวณและการ แทนค่ากลับของผลรวมของความผิดพลาดของผลลัพธ์กับค่าเป้าหมาย

3) ขั้นตอนการปรับค่าน้ำหนัก (weight) ไปเรื่อย ๆ ขั้นตอนนี้จะขึ้นอยู่กับขั้นตอนการ คำนวณไปข้างหน้าเพียงขั้นตอนเดียว โดยจะนำค่าน้ำหนักที่ได้จากการเรียนรู้ไปใช้ในการพยากรณ์ต่อไป ซึ่งสถาปัตยกรรมของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับแสดงในรูปที่ 2.12



รูปที่ 2.12 แสดงสถาปัตยกรรมของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.12 x แทนเวกเตอร์ของข้อมูลนำเข้า มีขนาด n มิติ และ w แทนเวกเตอร์น้ำหนักหรือความสำคัญที่ให้กับข้อมูลนำเข้าแต่ละตัว ฟังก์ชันผลรวมสามารถหาค่าได้จากสมการ

$$f(x) = \sum_{i=1}^{n}(x_i + y_i + Bias)$$

โดยที่ x_i แทนค่าข้อมูลเข้าตัวที่ i

y_i แทนค่าข้อมูลออกตัวที่ i

w_i แทนค่าน้ำหนักของนิวรอนตัวที่ i

n แทนจำนวนนิวรอนชั้นข้อมูลเข้า

$Bias$ แทนค่าความโน้มเอียง

i มีค่าตั้งแต่ 1 ถึง n

- output เป็นค่าตัวเลขที่ต่อเนื่องใช้ ReLU Function มีสมการดังนี้

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

โดยที่ $f(x)$ เป็นฟังก์ชันกระตุ้น

x แทนค่าข้อมูลนำเข้า

- output เป็นค่าตัวเลขที่ต่อเนื่องใช้ Hyperbolic Tangent Function มีสมการดังนี้

$$f(x) = \frac{1 - e^{-x}}{1 + e^{x}}$$

โดยที่ $f(x)$ เป็นฟังก์ชันกระตุ้น

e แทนฟังก์ชันเอกซ์โพเนนเชียล

x แทนค่าข้อมูลนำเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Back Propagation) นั้น ข้อมูลจากชั้นข้อมูลนำเข้า จะถูกคำนวณและส่งผ่านฟังก์ชันถ่ายโอน (Transfer Function) จากชั้นแฝง (Hidden Layer) ไปยังชั้นแสดงผล (Output Layer) โดยหลักการของกระบวนการเรียนรู้ของแบบจำลองคือ การเปลี่ยนแปลงค่าถ่วงน้ำหนัก (Weight) ของแต่ละการเชื่อมต่อเพื่อปรับผลลัพธ์ของแบบจำลองให้ ใกล้เคียงกับค่าความจริงมากที่สุดโดยอาศัยการแพร่กลับ เพื่อปรับค่าน้ำหนักในการเชื่อมต่อระหว่างนิวรอนให้เหมาะสม การปรับค่าน้ำหนักนี้จะขึ้นกับความแตกต่างของค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตที่ต้องการ โดยชุดข้อมูลที่ใช้ฝึกหัด (Training) โครงข่ายประสาทเทียมจะมีคำตอบไว้คอยตรวจดู ว่าโครงข่ายประสาทเทียมให้คำตอบที่ถูกต้องหรือไม่ ถ้าคำตอบไม่ถูกวงจรข่ายก็จะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้น ซึ่งการปรับน้ำหนักทำได้ตามสมการ

$$\Delta w_{ji}(n+1) = \eta\delta(n) * y_i(n) + \alpha\Delta w_{ji}(n)$$

โดยที่ x_i คือ ค่าข้อมูลด้านเข้าที่ node i

w_i คือ ค่าถ่วงน้ำหนักที่ node i

Δw_{ji} คือ ค่าปรับแก้ค่าถ่วงน้ำาหนักระหว่าง node i และ j

η คือ ค่าอัตราการเรียนรู้

α คือ ค่าโมเมนตัม

δ_j คือ ค่าผลต่างระหว่างค่าจริงกับค่าที่ได้จากการ คำนวณในรูปของอนุพันธของ Transfer function ของ node j

y_i คือ ค่าผลลัพธ์ของแบบจำลองที่ node j และ n 1 +n คือ ค่าที่แสดงถึงรอบของการปรับแก้ที่ n หรือ 1+n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานวิจัยนี้ได้ใช้ค่าความ คลาดเคลื่อนกำลังสองเฉลี่ย (Mean Squared Error หรือ MSE) เป็นดัชนีชี้วัดความแม่นยำของแบบจำลองโครงข่าย ประสาทเทียม ค่าเฉลี่ยความผิดพลาดของโครงข่ายเป็นผลรวมค่าเฉลี่ยความผิดพลาดยกกำลังสอง (Least Mean Square Error หรือ LMS) ที่เกิดจากการสอนโครงข่าย (Trian Network) ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Squared Error หรือ MSE) คำนวณได้จากสมการ

$$MSE = \sum_{t=1}^{N} (E_t)^2 / N$$

โดยที่ N คือ จำนวนของข้อมูลทั้งหมดที่ใช้ในการพยากรณ์

E_t คือ ผลต่างระหว่างค่าจริงของข้อมูลตัวที่ t ได้ กับค่าของข้อมูลตัวที่ t ซึ่งได้มาจากการพยากรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7 การปรับค่าโครงข่ายประสาทเทียมที่เหมาะสม

1. การเลือกตัวแปรอินพุตสำหรับโครงข่ายประสาทเทียม (Input network consideration)

- ควรลดจำนวนตัวแปรที่มีความซ้ำซ้อน (Redundant) และมีทิศทางเดียวกัน (Hight covariance)
- ควรพิจรณาตัวแปรที่มีค่าสหสัมพันธ์ (High correlation variable) มาก่อน และเพิ่มตัวแปรที่มีค่าสหสัมพันธ์รองลงมาในแบบจำลอง

2. ตัววัดสมรรถนะของโครงข่ายประสาทเทียม (Performance of Neural Network)

- Epoch คือ รอบการฝึกสอนของโครงข่ายประสาทเทียม
- Batch Size คือ เวลาที่โครงข่ายประสาทเทียมใช้ในการฝึกสอนต่อ 1 epoch
- ตัววัดความผิดพลาดที่ต่างจากเป้าหมาย เช่น ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Squared Error หรือ MSE) เป็นต้น จะใช้เป็นค่าเป้าหมายของการฝึกสอนของโครงข่ายได้ โดยทั่วไปค่าความผิเพลาดน้อยๆยิ่งดี
- อัตราการเรียนรู้ (Learning rate) คือ อัตราการเรียนรู้ที่ต้องการของโครงข่ายประสาทเทียม ถ้าเลือกให้มีค่ามากเกินไป จะเกิดการแกว่งในการหาคำตอบ (Oscillate) หรือไม่สามารถหาคำตอบได้เพราะไม่สเถียรในการค้นหาคำตอบ (Unstable)

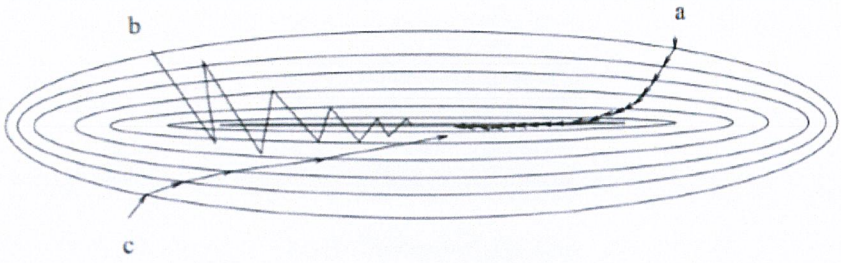
3. การเพิ่มสมรรถนะของโครงข่ายประสาทเทียม (Neural Network Performance Improvement)

- ปรับเปลี่ยนฟังก์ชันกระตุ้นและจำนวนนิวรอนในชั้นเพื่อให้ได้ค่าความผิดพลาด เวลาที่น้อยในการฝึกสอน บางโครงข่ายที่ซับซ้อนมากๆ อาจจะหยุดการฝึกสอนเนื่องจากครบจำนวนรอบ (Epoch) และคำตอบที่ได้เป็นเพียง (Best found)
- ปรับเงื่อนไขในการหยุดสอน (Stopping criteria) เพื่อหาคำตอบที่ดีกว่า ใช้ในกรณีที่เงื่อนไขการฝึกสอนเข้มงวดเกินไป (Tighten criteria)

4. อัตราการเรียนรู้และโมเมนตัม (Learning rate and Momentum)

อัตราการเรียนรู้ที่น้อยและไม่มีโมเมนตัม (กราฟ a) ดังรูปที่ 2.13 คำตอบของโครงข่ายจะค่อยๆลู่เข้าหา global minimum อย่างช้าๆ อัตราการเรียนรู้ที่มากและไม่มีโมเมนตัม (กราฟ b) ดังรูปที่ 2.13 คำตอบของโครงข่ายจะมีการสั่น (Oscillation) และไม่เสถียร (Unstable) อัตราการเรียนรู้ที่เหมาะสมและมีโมเมนตัม (กราฟ c) รูปที่ 2.13 โครงข่ายจะลู่เข้าหาคำตอบได้ดีที่สุด

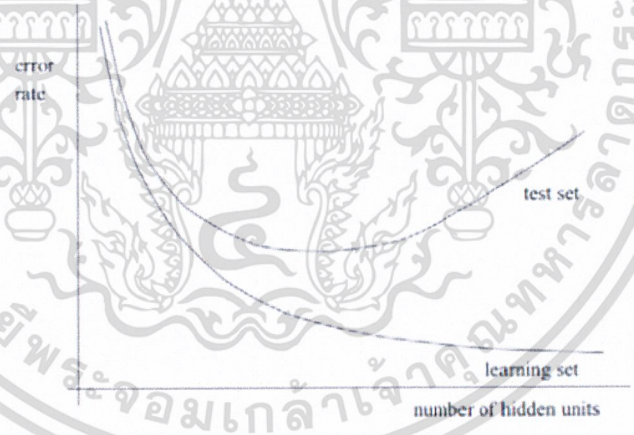
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ทิศทางการหาคำตอบของโครงข่ายประสาทเทียม

5. ผลของจำนวนนิวรอนในชั้นซ่อนต่ออัตราความผิดพลาด

การเพิ่มนิวรอนในชั้นซ่อน (Hidden neural) มากเกินไป จะทำให้อัตราความผิดพลาดในชุดฝึกสอนน้อยลง แต่ในขณะเดียวกัน จะทำให้โครงข่ายประสาทเทียมประมาณค่าฟังก์ชันเข้ากับสัญญาณรบกวนมากเกินไป (Fit to noise) ทำให้เกิดการฝึกสอนที่มากเกินความจำเป็น (Over-training) ซึ่งลักษณะที่โครงข่ายประสาทเทียมมีความอัตราผิดพลาดของชุดเรียนรู้ที่น้อย แต่มีอัตราการเกิดความผิดพลาดของชุดทดสอบมาก เราเรียกว่า 'Peaking effect'



รูปที่ 2.14 ลักษณะอัตราความผิดพลาดที่เกิด Peaking effect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 วิธีการประเมินผลลัพธ์จาก Model

ในการสร้างโมเดลโครงข่ายประสาทเทียม (Artificial Neural Network) จะมีการแยกชุดข้อมูลออกเป็น ข้อมูลฝึกสอน (Training) กับ ข้อมูลทดสอบ (Testing) ซึ่งผลลัพธ์ที่ได้จากการใช้โมเดล จะถูกนำมาเปรียบเทียบกับผลลัพธ์ที่เกิดขึ้นจริงในส่วนนี้

การประเมินผลลัพธ์สามารถประเมินได้หลายรูปแบบ ขึ้นอยู่กับเป้าหมายของการสร้างโมเดล แบ่งออกเป็น 4 ประการ ได้แก่ ความแม่นยำ, ความเร็วในการประมวลผล, ความยากลำบากในการใช้งาน, ความพึงพอใจในตัวโมเดล ซึ่งบางโมเดล อาจใช้การประเมินที่ต่างกัน ไม่จำเป็นต้องครบทั้ง 4 ประการก็ได้ สำหรับงานวิจัยนี้จะเลือกความแม่นยำเป็นตัวชี้วัดการประเมินผลลัพธ์

ผลลัพธ์ที่ได้จากโมเดล ประเภท Supervised Model หรือโมเดลประเภทที่มี Target ตัวตั้งต้นให้สามารถเปรียบเทียบได้ ก็จะสามารถหาความแม่นยำของโมเดลได้โดย

- R-Squared (Coefficient of Determination) หรือ ค่าสัมประสิทธิ์แสดงการตัดสินใจ เป็นค่าที่ใช้พิสูจน์ว่าโมเดลที่ได้นั้นเหมาะสมหรือไม่ โดยค่า R-Squared จะอยู่ระหว่า 0-1 ซึ่งโมเดลที่สมบูรณ์แบบที่สุด จะมีค่าเท่ากับ หรือใกล้เคียง 1 โดยทั่วไปควรมีค่า 0.6 ขึ้นไป และจะดีมากเมื่อมีค่า 0.8 ขึ้นไป แนวคิดของ R-Squared คือ เทียบระหว่างค่าความความคลาดเคลื่อน กับค่าเฉลี่ย

  สมการหาค่า R-Squared (Coefficient of Determination) คือ

$$R^2 = 1 - \frac{\sum_i^n e_i^2}{\sum_i^n (y_i - \bar{y})^2}$$

- Root Mean Square Error (RMSE) หรือ รากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง จุดประสงค์ของค่านี้ เพื่อสังเกตระยะห่างของความคาดเคลื่อน เพราะในบางกรณีอาจต้องการโมเดลที่มีความคาดเคลื่อนได้ แต่ต้องอยู่ในช่วงที่รับได้

  สมการหาค่า Root Mean Square Error (RMSE) คือ

$$RMSE = \sqrt{\frac{1}{n} * \sum_i^n (a_i - y_i)^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินงานวิจัย

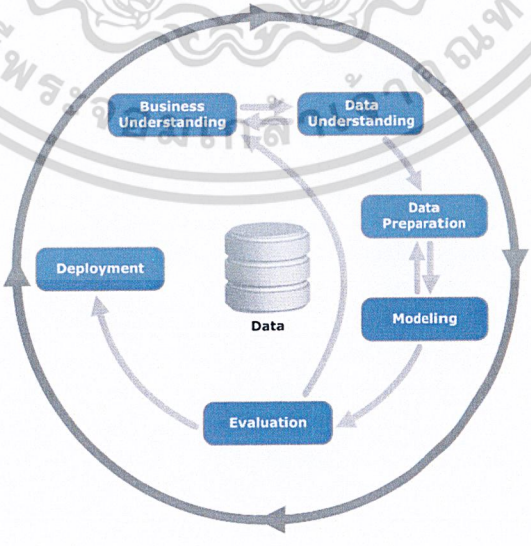
แนวทางการทำโมเดลเพื่อพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมมี 6 ขั้นตอนและเพื่อให้เกิดความเข้าใจในเบื้องต้น จะกล่าวถึงขั้นตอนและรายระเอียดการทำโมเดลเพื่อพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมคอนโดมิเนียมตลอดจนองค์ประกอบอื่นๆ ที่เกี่ยวข้อง

3.1 ขั้นตอนการทำโมเดลเพื่อพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม

ขั้นตอนหลักๆ ในการทำโมเดลเพื่อพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม มี 6 ขั้นตอนดังนี้

1. การทำความเข้าใจในธุรกิจ (Business Understanding)
2. การทำความเข้าใจข้อมูล (Data Understanding)
3. การเตรียมข้อมูล (Data Preparation)
4. การสร้างโมเดล (Modeling)
5. การประเมินผลจากโมเดล (Model Evaluation)
6. การนำไปใช้งาน (Deployment)

ซึ่งทั้ง 6 ขั้นตอนข้างต้นเรียกว่า วัฏจักรกระบวนการใช้ข้อมูล (Data Process Cycle) ในการทำ Data Science นี้ จึงเป็นสิ่งสำคัญที่ควรทำความเข้าใจ เพื่อมีมุมมองการทำงาน และลักษณะงานที่ชัดเจน



รูปที่ 3.1 วัฏจักรกระบวนการใช้ข้อมูล (Data Process Cycle)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภาพขั้นต้นแสดงถึงวัฏจักรกระบวนการใช้ข้อมูล (Data Process Cycle) เพื่อเป็นขั้นตอนการทำโมเดลเพื่อพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม โดยมีรายละเอียดดังนี้

1. การทำความเข้าใจในธุรกิจ (Business Understanding) เป็นขั้นตอนแรก เป็นการทำความเข้าใจปัญหา และความต้องการขององค์กร เพื่อวางเป้าหมายในการทำงาน ส่วนนี้เป็นส่วนที่สำคัญที่สุด เพราะไม่ใช่แค่จะมองปัญหาเพียงอย่างเดียว แต่มองไปถึงแนวทางการแก้ปัญหาขั้นต้นอีกด้วย ที่สำคัญที่สุด คือ ต้องมองไปถึงการนำไปใช้งานจริงอีกด้วย

2. การทำความเข้าใจข้อมูล (Data Understanding) การทำความเข้าใจข้อมูล เมื่อมีความเข้าใจตรงกันแล้วว่าต้องการทำโครงการเรื่องใด ก็จะเข้าสู่ขั้นตอนการเก็บรวบรวมข้อมูล และทำความเข้าใจว่าข้อมูลที่จะนำมาใช้ควรมีลักษณะอย่างไร

3. การเตรียมข้อมูล (Data Preparation) โดยปกติระบบประมวลผลข้อมูล นำเข้าข้อมูล จะอยู่ในรูปแบบที่จำกัด (Fixed Known Format) แต่ในความเป็นจริงข้อมูลส่วนใหญ่ ไม่ได้จัดเก็บในลักษณะดังกล่าวจึงต้องการกระบวนการแปลงข้อมูล (Data Transformation) หรือเปลี่ยนชนิดข้อมูล (Data Conversion) เพื่อให้ข้อมูลอยู่ในลักษณะหรือรูปแบบที่ง่ายต่อการนำไปประมวลผลหรือวิเคราะห์ต่อไป

4. การสร้างโมเดล (Modeling) การสร้างรูปแบบความสัมพันธ์ (Relational Pattern) อาจจะอยู่ในรูปของแบบจำลองบนซอฟต์แวร์ (Computer Model) หรือสมการความสัมพันธ์ (Equation) ก็ได้

5. การประเมินผลจากโมเดล (Model Evaluation) หลังจากที่ได้แบบจำลองแล้ว ต้องทำการประเมินผลว่าแบบจำลองนั้นมีความถูกต้องแม่นยำมากน้อยเพียงใด โดยอาจทดลองในระบบเสมือน (Simulation) หรือนำไปประมวลผลกับข้อมูลจริงที่มีอยู่เพื่อเปรียบเทียบผล ของการวิเคราะห์ว่าถูกต้องเป็นร้อยละเท่าใด

6. การนำไปใช้งาน (Deployment) หลังจากที่ได้แบบจำลองที่มีคุณภาพและความถูกต้องแม่นยำตามที่เราต้องการก็สามารถนำไปใช้งานจริง โดยอาจต้องมีการปรับแต่งเพื่อความเหมาะสมในสภาวะจริง อีกทั้งยังต้องติดตั้งร่วมกับระบบอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 รายระเอียดของการสร้างโมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม

3.2.1 เครื่องมือที่ใช้ในการพัฒนา

ภาษาที่ใช้ในการเขียนโปรแกรมประกอบไปด้วย Python โดยผ่าน Anaconda ซึ่งมี Library ที่หลากหลายซึ่งเหมาะสมต่อการพัฒนาระบบ โดยจะประกอบไปด้วย

ตารางที่ 3.1 แสดงเครื่องมือที่ใช้ในการสร้างโมเดล

ประเภท	ชื่อ
การเขียนโปรแกรม	Jupyter Notebook

ตารางที่ 3.2 ตารางแสดงสเปคของเครื่องคอมพิวเตอร์แม่ข่ายที่ใช้ในการประมวลผลและทดสอบระบบ

หน่วยประมวลผลกลาง (CPU)	8 cores
หน่วยความจำหลัก (RAM)	31.3 G
มี Power Supply แบบ Swap	7.8 G

3.2.2 ขั้นตอนในการดำเนินงาน

รายละเอียดการพัฒนาโมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียม

โดยงานวิจัยนี้สามารถแบ่งการทำงานเป็น 4 รูปแบบดังนี้

1. โมเดลเพื่อแยกช่วงราคาเป็น 3 ช่วงคือ 1-5 ล้านบาท, 5-10 ล้านบาท และ 10-20 ล้านบาท

2. โมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมในช่วงราคา 1–5 ล้านบาท

3. โมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมในช่วงราคา 5–10 ล้านบาท

4. โมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมในช่วงราคา 10–20 ล้านบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โมเดลเพื่อแยกช่วงราคาเป็น 3 ช่วงคือ 1-5 ล้านบาท, 5-10 ล้านบาท และ 10-20 ล้านบาท

3.3.1 การทำความเข้าใจในธุรกิจ (Business Understanding)

ราคาประเมินจะขึ้นอยู่กับปัจจัยต่างๆ ไม่ว่าจะเป็นลักษณะทางกายภาพของแปลงที่ดิน , การใช้ประโยชน์ , การเข้าถึงสาธารณูปโภค-สิ่งอำนวยความสะดวก , การออกแบบทางสถาปัตยกรรม อย่างขนาดของห้อง แบบห้อง สัดส่วนห้อง ดังนั้นการประเมินห้องในลักษณะเดียวกัน ทำเลที่ตั้งเหมือนกัน ราคาประเมินของผู้ประเมินแต่ละบุคคลอาจจะแตกต่างกันได้เนื่องจากประสบการณ์ความชอบของแต่และบุคคล นอกจากนี้ยังมีการประเมินราคาที่เกินกว่าความเป็นจริง การส่งเจ้าหน้าที่ไปสำรวจและประเมินราคาหลักทรัพทย์มีค่าใช้จ่ายที่สูงอีกทั้งยังเสียเวลาสำหรับการลงพื้นที่และมีความล่าช้าอย่างมาก ด้วยสาเหตุนี้จึงมีการทำโมเดลเพื่อพยากรณ์ราคาประเมินคอโดมิเนียมโดยอาศัยข้อมูลพื้นฐานและปัจจัยต่างๆที่ส่งผลกับราคาประเมินของคอนโดมิเนียม เพื่อแก้ไขปัญหาความล่าช้าในการสำรวจและประเมินหลักทรัพย์ ลดค่าใช้จ่ายในส่วนนี้บริษัทประเมิน

3.3.2 การทำความเข้าใจข้อมูล (Data Understanding)

การทำความเข้าใจข้อมูลนี้ มีองค์ประกอบหลักด้วยกัน 4 หัวข้อ ดังนี้

- **การเก็บรวบรวมข้อมูล** ซึ่งข้อมูลอาจมาจากหลายแหล่งก็เป็นได้
- **การระบุโครงสร้างข้อมูล** เช่น รายละเอียดภายในแต่ละคอลัมน์ เป็นต้น
- **การวิเคราะห์ข้อมูลเบื้องต้น** เช่น การเขียนกราฟเพื่อดูแนวโน้มเบื้องต้น การวิเคราะห์ค่าความสัมพันธ์ หรือ Correlation การหาค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และอื่นๆ เพื่อสร้างให้เกิดมุมมอง และหาความเกี่ยงโยงของข้อมูลแต่ละตัว
- **การตรวจสอบคุณภาพของข้อมูล** เช่น จำนวนชุดของข้อมูลมีขาดหายหรือไม่ อาจมีข้อความที่พิมพ์ผิด เป็นต้น ซึ่งหากข้อมูลนั้นไม่มีคุณภาพ อาจต้องหาข้อมูลจากแหล่งอื่นเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดการทำความเข้าใจข้อมูล (Data Understanding)

1. การเก็บรวบรวมข้อมูล

อธิบายถึงรายละเอียดของข้อมูลที่ได้รับมาคืออยู่ในไฟล์เจสัน (json file) สามารถอ่านออกได้ด้วยตาเปล่า โดย format จะมีรูปแบบเป็น คู่ Key-Value และมีความยืดหยุ่นของโครงสร้างข้อมูล จำนวนข้อมูลที่ได้รับมีทั้งหมด 85,698 ชุด

ตัวอย่าง ข้อมูลในไฟล์เจสัน (json file) 1 ชุด

{

"amenity_bbq_area": false,

"bathrooms": 1,

"ad_type": "sale",

"amenity_fitness": true,

"amenity_24_hours_security": true,

"amenity_cctv": true,

"internal_area": 72.0,

"property_type": "condo",

"amenity_playground": false,

"amenity_shop_on_premise": true,

"price_per_internal_area": 118055.56,

"building_floor": 28,

"amenity_elevator": true,

"district": "watthana",

"amenity_parking_lots": true,

"internal_area_unit": "sqm",

"subdistrict": "watthana",

"crawled_datetime": 1543333264,

"latitude": 13.7452858,

"latest_price_sale": 8500000.0,

"province": "bangkok",

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"description": "asoke tower 72 sq.m. 1 bedroom, 1 bathroom fully furnished and ready to move in bts ratchathewe sale 8,500,000 please contact us for more details",
"elasticsearch_status": 1,
"towers": 1,
"bedrooms": 1,
"raw_address": "thanon asok montri, khwaeng khlong toei nuea, khet watthana",
"amenity_wifi": false,
"link": "https://www.hipflat.co.th/en/listings/bangkok-condo-qbxlllgr",
"geo": [
100.5625686,
13.7452858
],
"_id": "hipflat_post_100472283946328468500285720226290606691610742755",
"year_built": 1986,
"elasticsearch_partition": "2018-10",
"title": "for sale 1 bed condo near mrt phetchaburi, bangkok, thailand",
"price_histories": [
{
"date": "2009-04-01",
"price_sale": 8500000.0
},
{
"date": "2018-11-15",
"price_sale": 8500000.0
},
{
"date": "2018-11-22",
"price_sale": 8500000.0
},

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                "date": "2018-11-27",
                "price_sale": 8500000.0
            }
        ],
        "longitude": 100.5625686,
        "amenity_restaurant_on_premise": true,
        "amenity_sauna": false,
        "amenity_swimming_pool": true,
        "data_source": "hipflat",
        "property_name_en": "asoke tower"
    },
```


จะสามารถเห็นข้อมูลได้อย่างคราวๆ ว่ามีลักษณะเป็นอย่างไร โดย key คือ ชื่อคอลัมน์ และ value คือ แถว จากข้อมูลเบื้องต้นสังเกตเห็นได้ว่า คอลัมน์ amenity จะมีการใส่ค่า true หรือ false ข้อมูลจะมีทั้งข้อมูลอยู่ในรูปของตัวเลข เช่น คอลัมน์ bathrooms, internal_area, building_floor, price_per_internal_area, towers, bedrooms, year_built เป็นต้น ข้อมูลอยู่ในรูปของข้อความหรือตัวหนังสือ ซึ่งอธิบายลักษณะของข้อมูล เช่น คอลัมน์ district , subdistrict, province เป็นต้น ในบางคอลัมน์ เช่น description อธิบายถึงรายละเอียดทั้งหมดของคอโดมิเนียมมีลักษณะเป็นข้อความ ซึ่งในคอลัมน์นี้จะไม่สามารถนำคอลัมน์นี้เข้าไปพยากรณ์ได้

ในไฟล์เจสันนั้นไม่เห็นลักษณะของข้อมูลได้อย่างชัดเจนดังนั้นจึงโหลดข้อมูลให้อยู่ใน data frame เพื่อให้ดูข้อมูลที่มีลักษณะเป็นตารางนั้นให้เกิดความเข้าใจมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การระบุโครงสร้างข้อมูล

แบ่งประเภทข้อมูลตามลักษณะของข้อมูล สามารถแบ่งได้ 2 ประเภทดังนี้

1. ข้อมูลเชิงปริมาณ (Quantitative data) หมายถึง ข้อมูลที่อยู่ในรูปตัวเลข (numerical data) ที่แสดงถึงปริมาณ อาจเป็นค่าที่ไม่ต่อเนื่อง (discrete) คือค่าที่เป็นจำนวนเต็มหรือจำนวนนับ หรือเป็นค่าที่ต่อเนื่อง (continuous) คือค่าที่มีจุดทศนิยมได้

2. ข้อมูลเชิงคุณภาพ (Qualitative data) หมายถึง ข้อมูลที่ไม่สามารถวัดออกมาเป็นจำนวนได้แต่อธิบายลักษณะของข้อมูลนั้นๆ

อธิบายถึงรายละเอียดของข้อมูลไฟล์เจสัน (json file) จากนั้นจากนั้นโหลดข้อมูลเข้าใน jupyter notebook ให้อยู่ใน data frame อยู่ในรูปตารางเพื่อให้ทำความเข้าใจง่ายแบ่งตามลักษณะของข้องมูลโดยเลือกคอลัมน์ที่ข้อมูลอยู่ในรูปของตัวเลขหรือข้อมูลเชิงปริมาณ ดังรูป

	bathrooms	bedrooms	building_floor	building_floor_extract	crawled_datetime	elasticsearch_status	internal_area	internal_area_extract	latest_price_rent
0	1	1	28	NaN	1543333264	1	72.00	NaN	NaN
1	1	1	33	NaN	1543903567	1	55.00	55.00	NaN
2	2	2	28	NaN	1543547908	1	80.00	NaN	NaN
3	1	1	8	NaN	1529603990	1	31.00	NaN	NaN
4	2	2	23	NaN	1529972074	1	81.00	NaN	NaN
5	1	1	36	NaN	1545980613	1	25.50	25.50	NaN
6	1	1	28	NaN	1543241124	1	34.53	NaN	NaN
7	1	1	8	NaN	1535034606	1	54.70	NaN	NaN
8	2	2	20	NaN	1542582828	1	100.00	NaN	NaN
9	1	1	30	NaN	1542172122	1	31.00	31.00	NaN
10	1	1	8	NaN	1546704014	1	50.00	NaN	NaN
11	1	1	24	NaN	1546872954	1	34.00	34.00	NaN
12	1	1	56	NaN	1529495865	1	24.00	NaN	NaN
13	1	1	51	NaN	1545416377	1	32.67	NaN	NaN
14	2	2	7	NaN	1543372653	1	67.29	NaN	NaN
15	1	1	45	NaN	1535557632	1	42.50	NaN	NaN
16	1	1	30	NaN	1546497473	1	35.26	35.26	NaN
17	1	1	8	8.0	1546756288	1	45.50	45.50	NaN
18	2	2	41	NaN	1543329448	1	109.00	NaN	NaN
19	1	1	7	NaN	1542625263	1	66.00	NaN	NaN
20	1	1	26	NaN	1531332631	1	41.00	NaN	NaN
21	1	1	34	NaN	1543369436	1	59.00	NaN	NaN
22	2	2	48	NaN	1542564628	1	86.50	NaN	NaN
23	1	1	31	NaN	1543546273	1	50.00	NaN	NaN

รูปที่ 3.2 data frame ของข้อมูลเชิงปริมาณ

จากรูป แสดงข้อมูลเชิงปริมาณของคอลัมน์ต่างๆ ซึ่งสังเกตได้ว่าลักษณะของข้อมูลมีเพียงตัวเลขเพียงอย่างเดียว และในบางคอลัมน์มีค่าเป็น NaN คือ ไม่มีค่าหรือไม่ทราบค่า ขั้นต่อไปจะอธิบายถึงชื่อคอลัมน์และความหมายของคอลัมน์นั้นๆทั้งหมด ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ชื่อคอลัมน์ของข้อมูลเชิงปริมาณ

ชื่อคอลัมน์	ความหมาย
bathrooms	จำนวนห้องอาบน้ำ
bedrooms	จำนวนห้องนอน
building_floor	ตำแหน่งชั้นของคอนโดมิเนียม
building_floor_extract	ชั้นของคอนโดมิเนียม
crawled_datetime	เวลาที่ดึงข้อมูล
elasticsearch_status	สถานะที่ดึงข้อมูล
internal_area	พื้นที่ภายในคอนโดมิเนียม
internal_area_extract	พื้นที่ภายในคอนโดมิเนียมเพิ่มเติม
latest_price_rent	ราคาเช่าคอนโดมิเนียม
latest_price_sale	ราคาขายคอนโดมิเนียม
latitude	ละติจูด
livingrooms	จำนวนห้องนั่งเล่น
longitude	ลองติจูด
parking_lots	จำนวนที่จอดรถ
parking_lots_extract	จำนวนที่จอดรถเพิ่มเติม
price_per_internal_area	ราคาต่อพื้นที่ภายในคอนโดมิเนียม
towers	จำนวนอาคารรวมของคอนโดมิเนียม
year_built	ปีที่สร้างคอนโดมิเนียม

คอลัมน์ที่แสดงข้างต้นมีทั้งคอลัมน์ที่ใช้ภายในองค์กร เช่น คอลัมน์ elasticsearch_status เป็นต้น คอลัมน์ที่ไม่ส่งผลต่อการพยากร์เช่น คอลัมน์ price_per_internal_area เป็นต้น และคอลัมน์ที่หากนำไปเข้าโมเดลอาจทำให้โมเดลไม่มีควสมแม่นยำเช่น คอลัมน์ latitude, longitude เป็นต้น ซึ่งในส่วนตรงนี้ไม่สามารถนำไปพยากรณ์ได้ อธิบายถึงข้อมูลที่ไม่สามารถวัดออกมาเป็นจำนวนได้แต่อธิบายลักษณะของข้อมูลนั้นๆ หรือข้อมูลเชิงคุณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

_parking_lots	amenity_playground	amenity_restaurant_on_premise	...	property_type	province	raw_address	road	room_area_unit	room_floor	room_floor_extr
True	False	True	...	condo	bangkok	thanon asok montri, khwaeng khlong toei nuea, ...	NaN	NaN	NaN	N
True	True	True	...	condo	bangkok	776 sukhumvit rd, khwaeng khlong tan, khet khl...	NaN	NaN	NaN	N
True	False	False	...	condo	bangkok	thanon sukhum vit, khwaeng phra khanong, khet ...	NaN	NaN	NaN	N
True	NaN	NaN	...	condo	bangkok	soi sali nimit, phra khanong nuea, vadhana, ba...	NaN	NaN	NaN	N
True	NaN	NaN	...	condo	bangkok	2105 sukhumvit rd, khwaeng bang chak, khet phr...	NaN	NaN	NaN	N
True	True	True	...	condo	bangkok	2317 sukhumvit rd, khwaeng bang chak, khet phr...	NaN	NaN	NaN	N
True	False	False	...	condo	bangkok	65 soi sukhumvit 19, khwaeng khlong toei nuea, ...	NaN	NaN	NaN	N
True	True	NaN	...	condo	chonburi	soi thep prasit 9, muang pattaya, amphoe bang ...	NaN	NaN	NaN	

รูปที่ 3.3 data frame ของข้อมูลเชิงคุณภาพ

จากรูป แสดงข้อมูลเชิงคุณภาพของคอลัมน์ต่างๆ ซึ่งสังเกตได้ว่าลักษณะของข้อมูลมีเพียงข้อความอย่างเดียว และคอลัมน์ amenity มีค่าเป็น Boolean คือ มีค่า true และ false ขั้นต่อไปจะอธิบายถึงชื่อคอลัมน์และความหมายของคอลัมน์นั้นๆทั้งหมด ดังนี้

ตารางที่ 3.4 ชื่อคอลัมน์ของข้อมูลทั้งหมด

ชื่อคอลัมน์	ความหมาย
_id	ไอดีของคอนโดมิเนียม
ad_type	ประเภท/ชนิด
amenity_24_hours_security	อำนวยความสะดวกเรื่องความปลอดภัย
amenity_bbq_area	อำนวยความสะดวกเรื่องพื้นที่บาบิคิว
amenity_cctv	อำนวยความสะดวกเรื่องกล้องวงจรปิด
amenity_elevator	อำนวยความสะดวกเรื่องลิฟท์
amenity_fitness	อำนวยความสะดวกเรื่องการออกกำลังกาย
amenity_parking_lots	อำนวยความสะดวกเรื่องที่จอดรถ
amenity_playground	อำนวยความสะดวกเรื่องพื้นที่นั่งเล่น
amenity_restaurant_on_premise	อำนวยความสะดวกเรื่องร้านอาหาร
	อำนวยความสะดวกเรื่องห้องซาวน่า
amenity_shop_on_premise	อำนวยความสะดวกเรื่องร้านสะดวกซื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

amenity_swimming_pool	อำนวยความสะดวกเรื่องสระว่ายน้ำ
amenity_wifi	อำนวยความสะดวกเรื่อง Wi-Fi
data_source	เป็นคอลัมน์ของการบอกแหล่งที่มา
description	บอกรายระเอียดของคอนโดมิเนียม
district	เขตที่ตั้งคอนโดมิเนียม
elasticsearch_partition	คอลัมน์ที่บอกว่าดึงข้อมูลในเดือน-ปี ที่เท่าไร
floor	ชั้นของคอนโดมิเนียม
geo	เป็นคอลัมน์ที่บอก latitude, longitude
internal_area_unit	คอลัมน์ที่บอกหน่วยของพื้นที่ภายในคอนโดมิเนียม
internal_area_unit_extract	คอลัมน์ที่บอกหน่วยของพื้นที่ภายในคอนโดมิเนียม
latest_updated_date	ราคาอัพเดพตามวันที่
link	บอกชื่อลิงค์จากการดึงข้อมูลมา
post_date	วันที่ประกาศขาย
price_histories	ประวัติการขาย
property_name_en	ชื่อคอนโดมิเนียม
property_type	เป็นคอลัมน์ที่บอกว่าเป็นประเภทใด ในที่นี้คือ คอนโดมิเนียม
province	จังหวัดของคอนโดมิเนียม
raw_address	ที่อยู่คอนโดมิเนียม
road	ชื่อถนน
room_area_unit	คอลัมน์ที่บอกหน่วยของพื้นที่ภายในคอนโดมิเนียม
room_floor	ชั้นของคอนโดมิเนียม
room_floor_extract	ชั้นของคอนโดมิเนียมเพิ่มเติม
subdistrict	อำเภอที่ตั้งคอนโดมิเนียม
title	บอกรายละเอียดของคอนโดว่ามีกี่ห้องนอน ห้องน้ำ
view_extract	บอกรายละเอียดเพิ่มเติมของห้องที่มีวิวต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การวิเคราะห์ข้อมูลเบื้องต้น

การวิเคราะห์ค่าความสัมพันธ์ (Correlation) การหาความสัมพันธ์ของตัวแปรกันต่อในลักษณะมีแนวโน้มที่จะไปด้วยกัน หรือตรงกันข้ามกัน แนวทางการใช้งาน Correlation เป็นการดูทิศทางความสัมพันธ์ระหว่างตัวแปร 2 ตัว โดยมี Correlation Coefficient (r) หรือ ค่าสัมประสิทธิ์สหสัมพันธ์เป็นตัวบ่งชี้ถึงความสัมพันธ์นี้ ซึ่งค่าสัมประสิทธิ์สหสัมพันธ์นี้จะมีค่าอยู่ระหว่าง -1.0 ถึง +1.0 ซึ่งหากมีค่าใกล้ -1.0 นั้นหมายความว่า ตัวแปรทั้งสองตัวมีความสัมพันธ์กันอย่างมากในทางตรงกันข้าม เช่น เมื่อตัวแปรค่าหนึ่งเพิ่มขึ้น อีกตัวแปรหนึ่ง จะมีค่าลดลงอย่างมีนัยสำคัญ หากมีค่าใกล้ +1.0 นั้นหมายความว่า ตัวแปรทั้งสองมีความสัมพันธ์กันโดยตรงอย่างมาก เช่น ค่าหนึ่งเพิ่ม อีกค่าเพิ่ม ค่าหนึ่งลด อีกก็ค่าลด และหากมีค่าเป็น 0 นั้นหมายความว่า ตัวแปรทั้งสองไม่มีความสัมพันธ์ต่อกัน

สมการหาค่า Correlation Coefficient (r) คือ

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

โดยที่ x คือ ตัวแปรต้น (Independent Variable) , $\bar{x}$ คือ ค่าเฉลี่ยของตัวแปรต้น

y คือ ตัวแปรตาม (dependent Variable) , $\bar{y}$ คือ ค่าเฉลี่ยของตัวแปรตาม

ทั้งนี้ การที่ตัวแปรทั้งสองตัวมี Correlation Coefficient แสดงออกถึง ความสัมพันธ์นั้น หมายความว่า ตัวแปรทั้งสองมีแนวโน้มจะไปในทิศทางเดียวกัน แต่ไม่ได้หมายความว่าตัวแปรทั้งสองนั้นเป็นปัจจัย หรือเป็นเหตุผลของกันและกัน

โดยมีสมมติฐานทางสถิติ คือ

$$\mathrm{H_0} : r = 0$$

$$\mathrm{H_a} : r \neq 0$$

สามารถทดสอบได้ด้วย T-test

$$\mathrm{t} = \frac{\mathrm{r}\sqrt{\mathrm{N}-2}}{\sqrt{1-\mathrm{r}^2}} \; ; \; \mathrm{df} = \mathrm{N}-2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อหาค่า Correlation Coefficient กับตัวแปรอื่นๆกับราคาของคอนโดมิเนียม ดังภาพต่อไปนี้

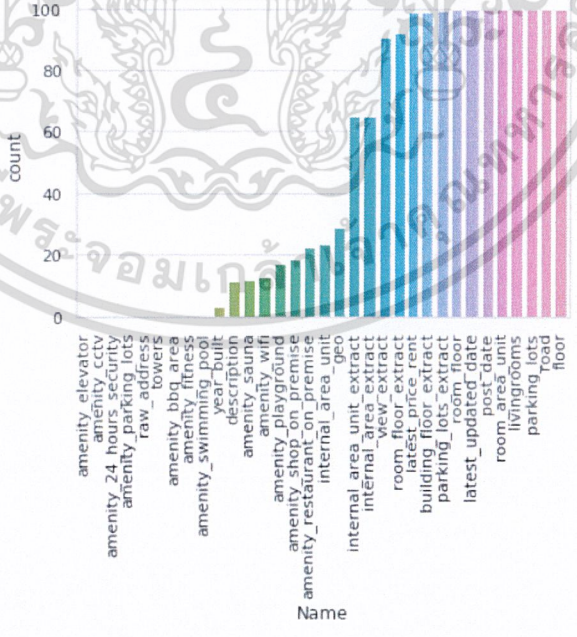
```
latest_price_sale          1.000000
price_per_internal_area    0.987086
parking_lots               0.825174
internal_area_extract      0.635796
latest_price_rent          0.136182
building_floor_extract     0.123610
bathrooms                  0.056178
building_floor             0.020004
bedrooms                   0.018555
latitude                   0.008344
crawled_datetime           0.004182
internal_area             -0.000079
towers                    -0.004245
longitude                 -0.005054
year_built                -0.008052
parking_lots_extract      -0.056034
elasticsearch_status            NaN
livingrooms                     NaN
Name: latest_price_sale, dtype: float64
```

รูปที่ 3.4 ค่า Correlation Coefficient ของข้อมูลทั้งหมด

พบว่า คอลัมน์ price_per_internal_area มีความสัมพันธ์กับคอลัมน์ latest_price_sale ถึง 0.987086

4. การตรวจสอบคุณภาพของข้อมูล

ตรวจสอบจำนวนชุดข้อมูลที่มีการขาดหาย เมื่อฟล็อตเป็นกราฟแท่ง



รูปที่ 3.5 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลเมื่อเทียบเป็นเปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนชุดข้อมูลที่ขาดหายเมื่อเทียบเป็นเปอร์เซ็นต์

ตารางที่ 3.5 จำนวนข้อมูลที่ขาดหาย

ชื่อคอลัมน์	จำนวนข้อมูลที่หาย(เปอร์เซ็นต์)
amenity_elevator	0.033840
amenity_cctv	0.045509
amenity_24_hours_security	0.057178
amenity_parking_lots	0.087517
raw_address	0.093351
towers	0.149362
amenity_bbq_area	0.206539
amenity_fitness	0.441084
amenity_swimming_pool	0.497094
year_built	3.299960
description	11.224299
amenity_sauna	11.993279
amenity_wifi	12.806600
amenity_playground	17.404140
amenity_shop_on_premise	18.866251
amenity_restaurant_on_premise	22.419426
internal_area_unit	23.355271
geo	28.641275
internal_area_unit_extract	65.166048
internal_area_extract	65.206889
view_extract	90.485192
room_floor_extract	92.149175
latest_price_rent	98.736260
building_floor_extract	98.927630
parking_lots_extract	99.071157

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

room_floor	99.936988
latest_updated_date	99.967327
post_date	99.967327
room_area_unit	99.967327
livingrooms	99.984830
parking_lots	99.985997
road	99.988331
floor	99.989498

3.3.3 การเตรียมข้อมูล (Data Preparation)

ขั้นตอนในการเตรียมข้อมูล ประกอบไปด้วย

- **การเลือกข้อมูล** คือ เลือกคอลัมน์สำหรับการนำไปพยากรณ์
- **การทำความสะอาดข้อมูล** คือ พิจรณาข้อมูลที่ขาดหาย (Missing Value) หมายถึงข้อมูลที่วัดเก็บบางส่วนมีค่าว่าง ข้อมูลที่มีค่าผิดปกติหรือเกินความเป็นจริง (Outlier) หมายถึงข้อมูลมีค่ามากกว่าหรือน้อยกว่าค่าที่ควรจะเป็น
- **การจัดรูปแบบของข้อมูลให้อยู่ในรูปแบบพร้อมสร้างโมเดล** คือ การเปลี่ยนรูปข้อมูล (Data Transformation) เพื่อให้ข้อมูลอยู่ในรูปของตัวเลขพร้อมสำหรับการประมวลผลโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การเลือกข้อมูล

จากข้างต้น เมื่อดูจำนวนข้อมูลที่ขาดหายมากกว่า 30 เปอร์เซ็นต์ พบว่า มี 15 คอลัมน์ ซึ่งหากมีจำนวนที่ขาดหายมาก จะไม่สามารถนำข้อมูลไปพยาการณ์ได้ (ข้อมูลไม่เพียงพอ) เนื่องจากบางคอลัมน์เป็นคอลัมน์ที่ใช้กันภายในบริษัท เช่น elasticsearch partition เป็นคอลัมน์ที่บอกว่าดึงข้อมูลในเดือน-ปี ที่เท่าไร และคอลัมน์ที่เป็นประโยคเช่น raw address เป็นการบอกที่อยู่ ซึ่งไม่สามารถนำเข้าโมเดลได้ ซึ่งตรงนี้จะมีคอลัมน์ที่บอกว่าตั้งอยู่เขต จังหวัด อยู่แล้ว ในส่วนตรงนี้เราจะตัดคอมลัมน์ดังกล่าวทิ้งไป

latitude, longitude เป็นคอลัมน์ที่บอกที่ตั้งตามพิกัดโลกของคอนโดมิเนียม เนื่องจากเมื่อค่า latitude, longitude เปลี่ยนไปจะส่งผลต่อการพยากรณ์ราคา ซึ่งทำให้พยากรณ์ราคาผิดไปเกินความเป็นจริง ดังนั้นจึงไม่ควรนำทั้งสองคอลัมน์เข้าโมเดล

คอลัมน์ของตัวเลขที่ไม่มีผลต่อการพยาการณ์ราคา ดังนั้นจึงตัดคอลัมน์ทิ้งไปดังนี้

1. id เป็นคอลัมน์รหัสของคอนโดมิเนียมและแหล่งที่มา ดังตัวอย่าง hipflat_post_19813048186288904371722235067988996591554787688113

2. data_source เป็นคอลัมน์ของการบอกแหล่งที่มา พบว่ามี 2 แหล่งคือ hipflat = 85670 แถว Thinkofliving = 28 คอลัมน์

3. ad_type เป็นคอลัมน์ที่บอกประเภทการขาย

4. description เป็นคอลัมน์ที่บอกรายระเอียดของคอนโดมิเนียม

5. geo เป็นคอลัมน์ที่บอก latitude, longitude ตัวอย่างเช่น [100.5625686, 13.7452858]

6. internal_area_unit, internal_area_unit_extract, room_area_unit เป็นคอลัมน์ที่บอกหน่วยของพื้นที่ภายในคอนโดมิเนียม มีหน่วยเป็น sqm

7. latest_updated_date, post_date เป็นคอลัมน์ที่บอกวันที่ที่อัพเดทราคาขายคอนโดมิเนียมและวันที่ที่ลงประกาศขายคอนโดมิเนียม

8. link เป็นคอลัมน์ที่บอกชื่อลิงค์จากการดึงข้อมูลมา

9. property_name_en เป็นคอลัมน์ที่บอกชื่อคอนโดมิเนียม

10. raw_address เป็นคอลัมน์ที่บอกที่อยู่คอนโดมิเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. title เป็นคอลัมน์ที่บอกรายละเอียดของคอนโดว่ามีกี่ห้องนอน ห้องน้ำ เป็นต้น

12. property_type เป็นคอลัมน์ที่บอกว่าเป็นประเภทใด ในที่นี้คือคอนโดมิเนียม

13. crawled_datetime เป็นคอลัมน์ที่บอกว่าดึงข้อมูลมาในวัน-เดือน-ปี

14. elasticsearch_status เป็นคอลัมน์ที่ใช้ภายในบริษัทที่บอกว่าดึงข้อมูลมาแล้ว มีค่าเป็นเลข 1 ทั้งหมด

15. price_per_internal_area เป็นคอลัมน์ของพื้นที่ภายในคอนโดมิเนียมหารด้วยราขาขาย ซึ่งไม่ควรนำเข้าโมเดลเนื่องจากเป็นการบอกเฉลยของราคาขายอยู่แล้ว เมื่อนำโมเดลมาใช้งานจริงอาจทำให้โมเดลพยากรณ์ได้ผิด และตัวโมเดลนั้นอาจไม่มีแม่นยำ

ตัวแปรที่สามารถนำไปพยาการณ์ราคาคอนโดมิเนียมมีดังนี้ คือ

ข้อมูลเชิงปริมาณ

1. bathrooms จำนวนห้องน้ำ

2. bedrooms จำนวนห้องนอน

3. building_floor ตำแหน่งชั้นของคอนโดมิเนียม

4. year built ปีที่สร้างคอนโดมิเนียม

5. internal area พื่นที่ภายในคอนโดมิเนียม

6. latest price sale ราขาขาย

7. towers จำนวนอาคารทั้งหมดในโครงการ

ข้อมูลเชิงคุณภาพ

8. subdistrict อำเภอที่คอนโดมิเนียมตั้ง

9. province จังหวัด

10. district เขตที่คอนโดมิเนียมตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. คอลัมน์ amenity มีทั้งหมด 12 คอลัมน์ ดังนี้

amenity_24_hours_security, amenity_bbq_area, amenity_cctv, amenity_elevator, amenity_fitness, amenity_parking_lots, amenity_playground, amenity_sauna, amenity_restaurant_on_premise, amenity_sauna, amenity_shop_on_premise, amenity_swimming_pool, amenity_wifi

ลักษณะของคอลัมน์ amenity มีสามค่าคือ True, False, NaN หมายถึง เมื่อคอนโดมิเนียมมีการอำนวยสะดวกใดจะใส่คำว่า True หากในรายการนั้นไม่มีจะใส่ค่า False และค่า NaN คือไม่ทราบว่าคอนโดมิเนียมมีการอำนวนความสะดวก (เป็นการดึงข้อมูลมาและไม่ระบุในคอลัมน์นั้น)

2. การทำความสะอาดข้อมูล

โดยเริ่มจาก Exploratory data analysis (EDA) เพื่อให้เราเข้าใจในลักษณะของข้อมูล ในช่วงราคานี้มีข้อมูลทั้งหมด 85,698 แถว และ 22 คอลัมน์ ดูค่าสถิติเบื้องต้นดังนี้

	bathrooms	bedrooms	building_floor	internal_area	latest_price_sale	towers	year_built
count	85698	85698	85698	85698	85698	85570	82870
mean	1	1	28	209	10669586	1	2012
std	0	2	15	26459	135688235	1	6
min	1	1	2	0	1	1	1981
25%	1	1	17	34	4340000	1	2009
50%	1	1	30	48	6900000	1	2013
75%	2	2	38	76	12000000	2	2016
max	56	99	77	6500000	39600000000	27	2023

รูปที่ 3.6 ค่าสถิติเบื้องต้นของข้อมูลทั้งหมด

จากรูปข้างต้น จะเห็นค่าสถิติอย่างคราวๆของคอลลัมน์ที่เป็นข้อมูลเชิงปริมาณ จากการสังเกตจะเห็นได้ว่า คอลัมน์ bathrooms, bedrooms, building_floor, internal_area, latest_price_sale, tower มีค่าผิดปกติเกินความเป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 ข้อมูลที่ขาดหาย (Missing Value)

เมื่อสังเกตคอลัมน์ที่มีค่าว่างหรือไม่มีค่า เมื่อเป็นจำนวนทั้งหมดและเทียบเป็นเปอร์เซ็นต์ดังนี้

	Total	Percent
amenity_restaurant_on_premise	19213	22.419426
amenity_shop_on_premise	16168	18.866251
amenity_playground	14915	17.404140
amenity_wifi	10975	12.806600
amenity_sauna	10278	11.993279
year_built	2828	3.299960
amenity_swimming_pool	426	0.497094
amenity_fitness	378	0.441084
amenity_bbq_area	177	0.206539
towers	128	0.149362
amenity_parking_lots	75	0.087517
amenity_24_hours_security	49	0.057178
amenity_cctv	39	0.045509
amenity_elevator	29	0.033840
subdistrict	0	0.000000
province	0	0.000000
latest_price_sale	0	0.000000
internal_area	0	0.000000
district	0	0.000000
building_floor	0	0.000000
bedrooms	0	0.000000
bathrooms	0	0.000000

รูปที่ 3.7 data frame คอลัมน์ที่มีค่าว่างหรือไม่มีค่า

จากข้างต้นจะเห็นว่า year built มี 3.29 เปอร์เซ็นต์ และ towers มี 0.14 เปอร์เซ็นต์ของจำนวนที่หายไปดังนั้นจึงเลือกวิธีการลบคอลัมน์นั้นทิ้งไปเนื่องจากหากเติมค่านั้นอาจทำให้ข้อมูลผิดพลาดได้ และในส่วนของ amenity ทั้งหมดหารลบคอลัมน์ที่เป็น NaN ทิ้งไปอาจทำให้ข้อมูลนั้นไม่พียงพแต่อการพยากรณ์ดั้งนั้นจึงเลือกวิธีการเติมค่าที่หายไปด้วยค่า False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ข้อมูลที่มีค่าผิดปกติหรือเกินความเป็นจริง (Outlier)

จากค่าสถิติเบื้องต้น เห็นได้ว่า ในคอลัมน์ที่เป็นข้อมูลเชิงปริมาณ เช่น คอลัมน์ bathrooms, bedrooms เป็นต้น มีค่ามากเกินกว่าความเป็นจริง ซึ่งตรวจสอบข้อมูลแล้วพบว่าไม่มีข้อมูลนั้นอยู่จริง ซึ่งในส่วนนี้คือค่าผิดปกติ จะตัดค่าผิดปกตินี้โดยวิธี พิสัยระหว่างควอไตล์ (Interquartile Range)

ค่าควอไตล์ (Quartile) หมายถึง ค่าทางสถิติที่แบ่งข้อมูลออกมาเป็น 4 ส่วนเท่าๆ กัน ส่วนละ 25% หรือ ¼ ตามขนาดการกระจายของข้อมูลทั้งหมด โดยควอไตล์ที่ 1 (Q_1) จะหมายถึง ตัวสถิติที่ทำการแบ่งให้ข้อมูล 25% หรือ 1 ใน 4 ส่วนมีค่าน้อยกว่าค่าของมัน ส่วนควอไตล์ที่ 3 (Q_3) จะหมายถึง ตัวสถิติที่ทำการแบ่งให้ข้อมูล 75% หรือ 3 ใน 4 ส่วนมีค่าน้อยกว่าค่าของมัน โดยควอไตล์สามมารถคำนวนได้จากสมการ

$$Q_r = \frac{r(n+1)}{4}$$

เมื่อ Q_r คือ ค่าควอไตล์ตำแหน่งที่ r

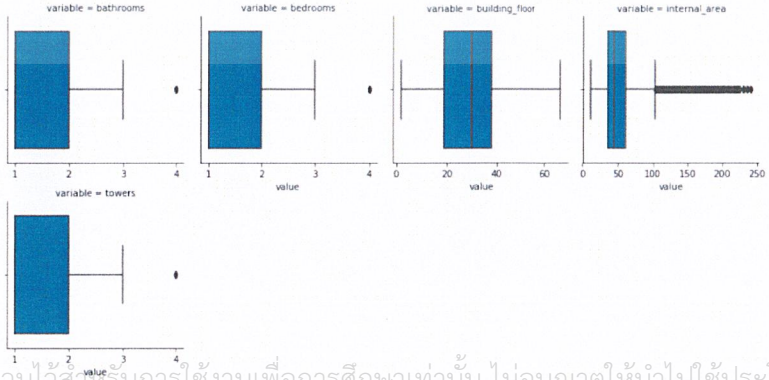
r คือ ตำแหน่งของควอไตล์ (ที่มีค่าเท่ากับ 1,2 และ 3)

n คือ จำนวนข้อมูล

ดังนั้น พิสัยระหว่างควอไตล์ (Interquartile Range) หมายถึง ค่าแตกต่างระหว่าง ควอไตล์ที่ 3 (Q_3) และควอไตล์ที่ 1 (Q_1) ซึ่งเขียนแทนด้วยสัญลักษณ์ (IQR) ดังสมการ

$$IQR = Q_3 - Q_1$$

หลังตัดค่าผิดปกติของ คอลัมน์ bathrooms, bedrooms , building_floor , internal_area , towers



รูปที่ 3.8 กราฟ boxplot หลังตัดค่าผิดปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การจัดรูปแบบของข้อมูลให้อยู่ในรูปแบบพร้อมสร้าง Model

คอลัมน์ district, subdistrict, province เป็นข้อมูลเชิงคุณภาพไม่สามารถนำไปกรณ์ได้ดังนั้นจึงใช้เทคนิคที่เรียกว่า one-hot encoding คือการเข้ารหัสข้อมูลให้เป็น binary (0 กับ 1)
คอลัมน์ latest_price_sale มีราคาตั้งแต่ 1-39,600,000,000 ล้านบาท แต่เนื่องจากทำโมเดลในช่วงราคา 1-20 ล้านบาท ดังนั้นจึงแบ่งช่วงราคาออกเป็นทั้ง 3 กลุ่ม คือ

- กลุ่มที่ 1 มีช่วงราคา 1-5 ล้านบาท
- กลุ่มที่ 2 มีช่วงราคา 5-10ล้านบาท
- กลุ่มที่ 3 มีช่วงราคา 10-20 ล้านบาท

ซึ่งหาก คอลัมน์ latest_price_sale มีช่วงราคาจะจัดอยู่ในกลุ่มนั้น

3.3.4 การสร้างโมเดล (Modeling)

โมเดลเพื่อแยกช่วงราคาเป็น 3 ช่วงคือ 1-5 ล้านบาท, 5-10 ล้านบาท และ 10-20 ล้านบาท เป็นSupervised Model คือ โมเดลประเภทที่มี Target หรือคำตอบของโมเดลเพื่อให้สามารถเปรียบเทียบกับผลลัพธ์ที่เกิดขึ้นจริงได้ ในลักษณะการทำโมเดลเพื่อแยกช่วงราคาเป็นการทำ Classification คือ การพยากรณ์ที่มีลักษณะการทำงานเป็นการแบ่งกลุ่มของข้อมูล ตัวอย่างเช่น Yes/No หรือกลุ่ม (A, B, C) เป็นต้น ในที่นี้คือเป็นการแบ่งช่วงของราคาคอนโดมิเนียม ใช้โมเดล Random Forests

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โมเดลการพยากรณ์ราคาประเมินในช่วงราคา 1–5 ล้านบาท

3.4.1 การทำความเข้าใจข้อมูล (Data Understanding)

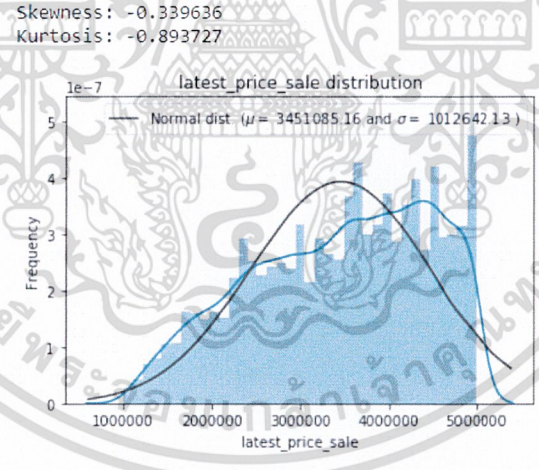
โดยเริ่มจาก Exploratory data analysis (EDA) เพื่อให้เราเข้าใจในลักษณะของข้อมูลในช่วงราคา 1-5 ล้านบาท ในช่วงราคานี้มีข้อมูลทั้งหมด 27,187 แถว และ 22 คอลัมน์ ดูค่าสถิติเบื้องต้นดังนี้

[2]:

	bathrooms	bedrooms	building_floor	internal_area	latest_price_sale	towers	year_built
count	27187	27187	27187	27187	27187	27167	25787
mean	1	1	22	244	3451085	2	2012
std	0	2	12	25550	1012660	1	5
min	1	1	2	0	1000000	1	1984
25%	1	1	8	28	2670000	1	2010
50%	1	1	24	33	3570000	1	2014
75%	1	1	32	40	4300000	2	2016
max	56	99	74	3790000	5000000	27	2022

รูปที่ 3.9 แสดงค่าสถิติเบื้องต้นในช่วงราคา 1-5 ล้านบาท

วิเคราะห์ตัวแปร latest price sale

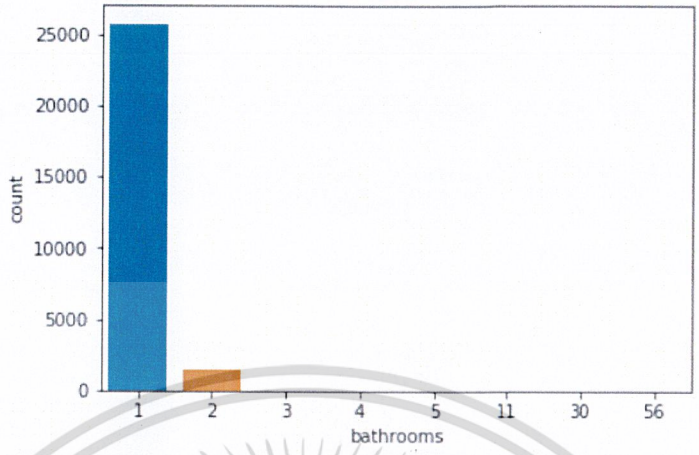


รูปที่ 3.10 แสดงการกระจายตัวของราคาในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่า ช่วงราคา 1-5 ล้านบาทมีการกระจายตัวใกล้เคียงการกระจายแบบปกติ นั่นทำให้เกิดข้อผิดพลาดในการพยากรณ์น้อยลง

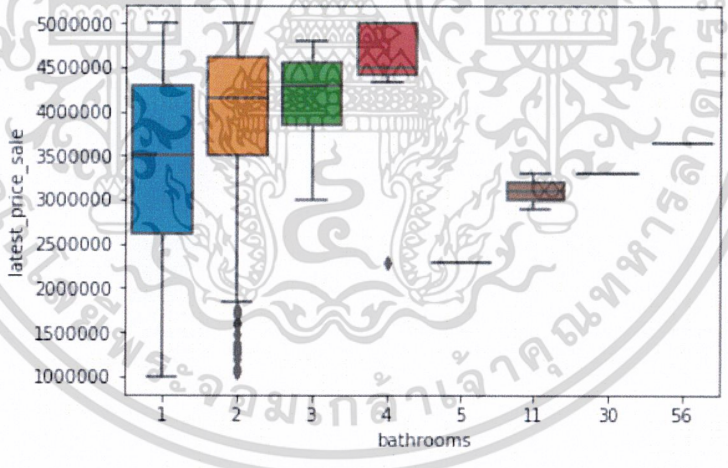
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร bathrooms



รูปที่ 3.11 แสดงจำนวนห้องอาบน้ำในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 1-5 ล้านบาท ที่ 1 ห้องอาบน้ำมีจำนวน 25,746 ห้อง, 2 ห้องอาบน้ำมีจำนวน 1,412 ห้อง, 3 ห้องอาบน้ำมีจำนวน 17 ห้อง, 4 ห้องอาบน้ำมีจำนวน 7 ห้อง, 5 ห้องอาบน้ำมีจำนวน 2 ห้อง, 11 30 และ 56 ห้องอาบน้ำมีจำนวน 1 ห้อง

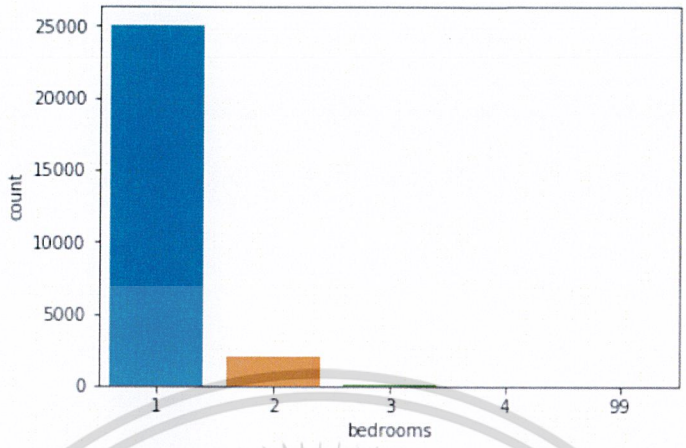


รูปที่ 3.12 แสดงจำนวนห้องอาบน้ำเทียบกับราคาในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่าตัวแปรห้องน้ำเมื่อเทียบกับราคาพบว่าส่วนใหญ่ ช่วงราคา 1-5 ล้านบาท จะมี1-4 ห้องน้ำ และที่ 11 ห้องน้ำจะอยู่ในช่วงราคา 3-3.5 ล้านบาท

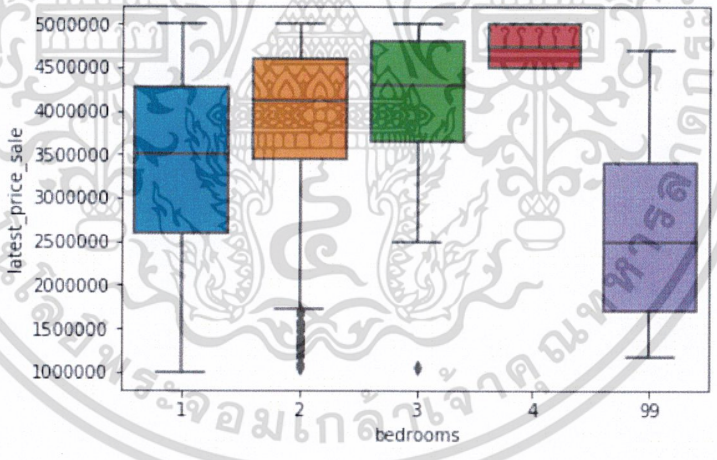
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร bedrooms



รูปที่ 3.13 แสดงจำนวนห้องนอนในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 1-5 ล้านบาท ที่ 1 ห้องนอนมีจำนวน 25,081 ห้อง, 2 ห้องนอนมีจำนวน 2,026 ห้อง, 3 ห้องนอนมีจำนวน 51 ห้อง, 4 ห้องนอนมีจำนวน 25 ห้อง, 99 ห้องนอนมีจำนวน 4 ห้อง

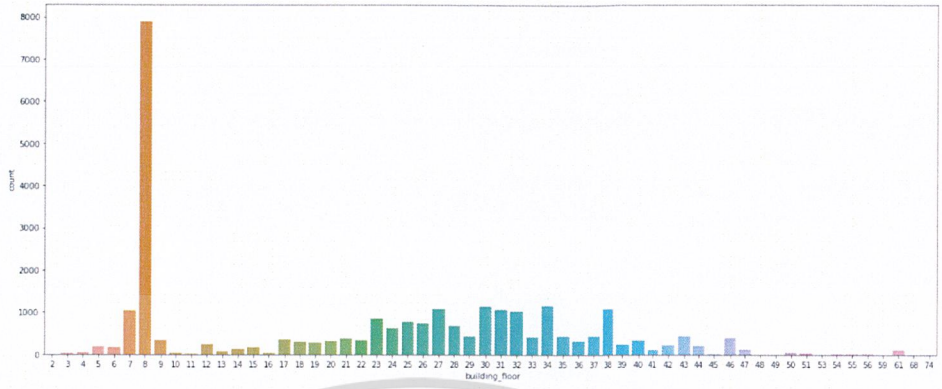


รูปที่ 3.14 แสดงจำนวนห้องนอนเมื่อเทียบกับราคาในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่าตัวแปรห้องนอนเมื่อเทียบกับราคาพบว่าส่วนใหญ่ ช่วงราคา 1-5 ล้านบาท จะมี1-4 ห้องนอน และที่ 99 ห้องนอนเมื่อลองตรวจสอบแล้วพบว่าไม่มีจริง ส่วนนี้เราจะตัดค่าผิดปกติออกไป

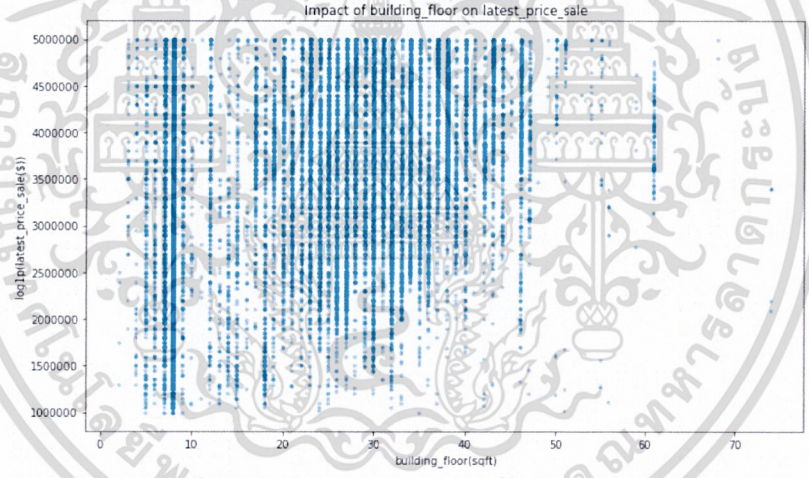
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร building floor



รูปที่ 3.15 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 1-5 ล้านบาท พบว่าชั้นที่ 8 มีจำนวนคอนโดมิเนียมมากสุดถึง 7,902 ห้อง รองลงมาคือชั้นที่ 34 คือ 1,153 ห้อง และ 30 คือ 1,138 ห้อง

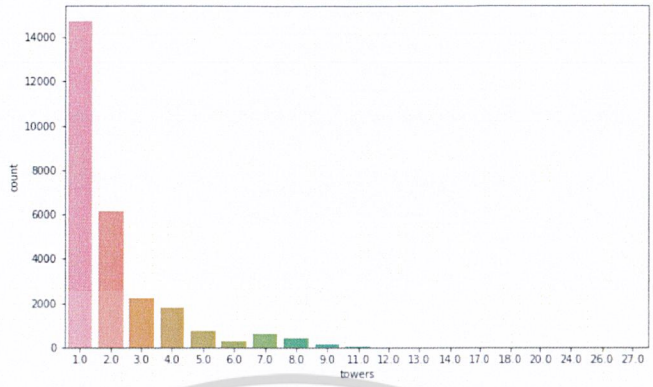


รูปที่ 3.16 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมเทียบกับราคาในช่วง 1-5 ล้านบาท

จากภาพการเปรียบเทียบระหว่าชั้นคอนโดมิเนียมและราคาขายในช่วงราคา 1-5 ล้านบาท พบว่าชั้นที่ 20-30 ชั้นจะอยู่ในช่วงราคา 2-5 ล้านบาท และชั้น 50-60 จะอยู่ในช่วงราคา 3.5 ล้านบาทขึ้นไป

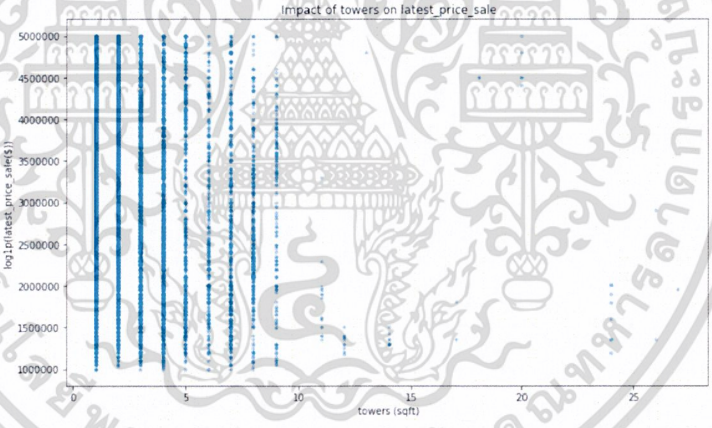
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร towers



รูปที่ 3.17 แสดงจำนวนอาคารรวมของคอนโดมิเนียมในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 1-5 ล้านบาท อาคารรวมของคอนโดมิเนียมส่วนมากจะอยู่ที่ 1-6 อาคาร เป็นส่วนใหญ่โดยที่ 1 อาคารมีจำนวน 14,704 อาคาร, 2 อาคารมีจำนวน 6,161 อาคาร, 3 อาคารมีจำนวน 2,218 อาคาร

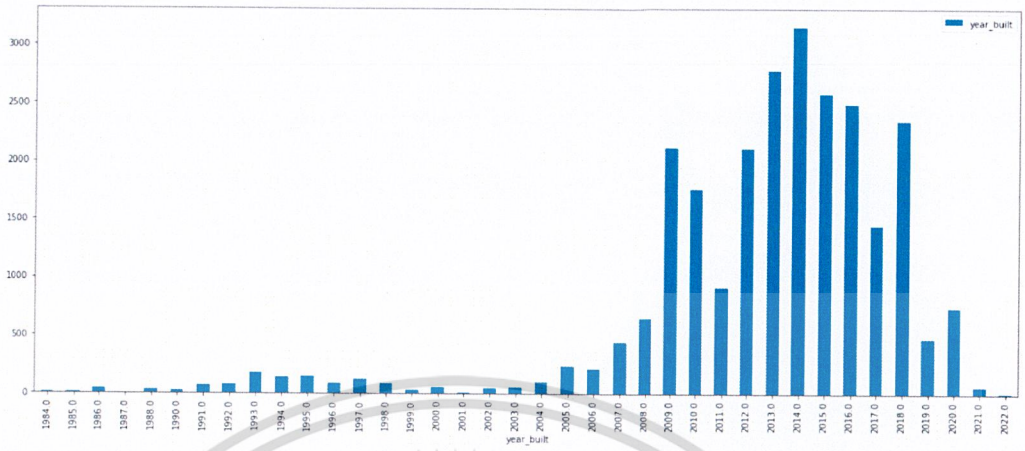


รูปที่ 3.18 แสดงจำนวนอาคารรวมของคอนโดมิเนียมเทียบกับราคาในช่วง 1-5 ล้านบาท

จากภาพการเปรียบเทียบระหว่างอาคารรวมของคอนโดมิเนียมและราคาขายในช่วงราคา 1-5 ล้านบาท จะเห็นได้ว่า 1-10 อาคารรวมของคอนโดมิเนียมมีราคาตั้งแต่ 1-5 ล้านบาท และส่วนน้อยจะมีราคาต่ำกว่า 2 ล้านบาท

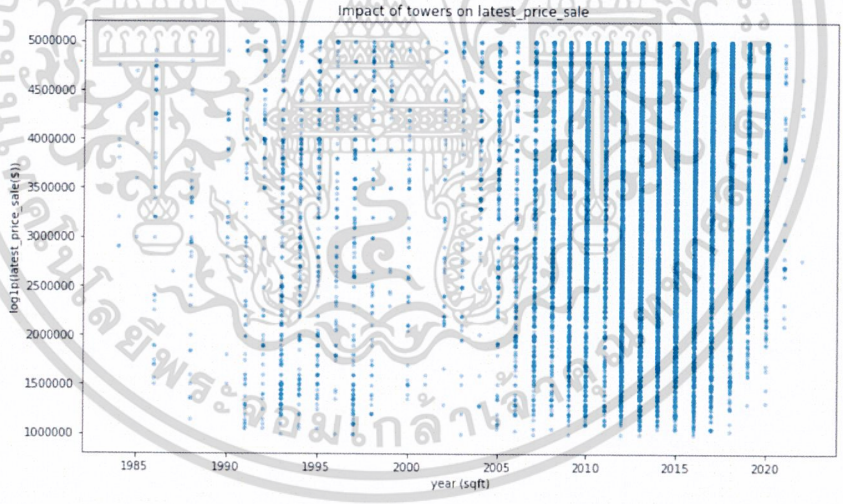
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร year built



รูปที่ 3.19 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมในช่วง 1-5 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 1-5 ล้านบาท ปีที่สร้างคอนโดมิเนียมมากที่สุดคือ 2014 จำนวน 3157 ห้องรองลงมาคือปี 2013 คือ 2,781 ห้อง เมื่อดูแนวโน้มจะเห็นว่าจะอยู่ในช่วงระหว่างปี 2008-2018 และที่กำลังสร้างในอนาคต



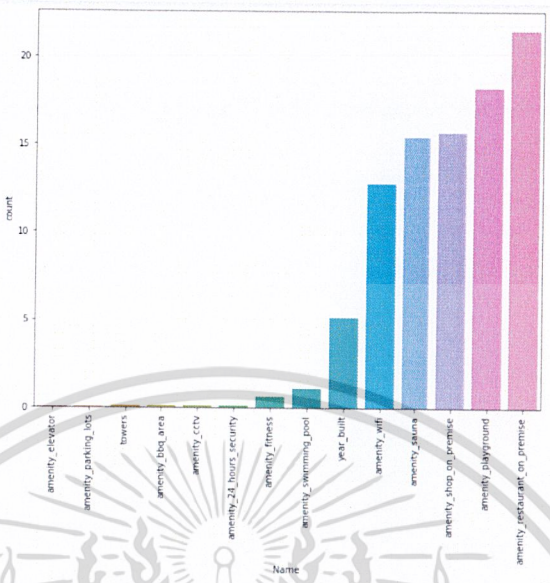
รูปที่ 3.20 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมเทียบกับราคาในช่วง 1-5 ล้านบาท

จากภาพการเปรียบเทียบระหว่างปีที่สร้างคอนโดและราคาขายเห็นได้ว่าช่วงราคา 1-5 ล้านบาท พบว่าช่วงปี 2010-2010 ที่สร้างคอนโดมิเนียมมีช่วงราคาตั้งแต่ 1-5 ล้านบาทและนอกจากนั้นจะมีการการจายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การเตรียมข้อมูล (Data Preparation)

วิเคราะห์จำนวนข้อมูลที่หายไป



รูปที่ 3.21 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลในช่วงราคา 1-5 ล้านบาท

	Total	Percent
amenity_restaurant_on_premise	5854	21.532350
amenity_playground	4962	18.251370
amenity_shop_on_premise	4269	15.702358
amenity_sauna	4195	15.430169
amenity_wifi	3481	12.803914
year_built	1400	5.149520
amenity_swimming_pool	299	1.099790
amenity_fitness	161	0.592195
amenity_24_hours_security	29	0.106669
amenity_cctv	23	0.084599
amenity_bbq_area	21	0.077243
towers	20	0.073565
amenity_parking_lots	17	0.062530
amenity_elevator	14	0.051495
subdistrict	0	0.000000
province	0	0.000000
latest_price_sale	0	0.000000

รูปที่ 3.22 คอลัมน์และจำนวนข้อมูลที่หายไปในช่วงราคา 1-5 ล้านบาท

จากข้างต้นจะเห็นว่า year built มี 5.14 เปอร์เซ็นต์ของจำนวนที่หายไปดังนั้นจึงเลือกวิธีการลบคอลัมน์นั้นทิ้งไปเนื่องจากหากเติมค่านั้นอาจทำให้ข้อมูลผิดพลาดได้ และในส่วนของ amenity ทั้งหมดหารลบคอลัมน์ที่เป็น NaN ทิ้งไปอาจทำให้ข้อมูลนั้นไม่พียงพอต่อการพยากรณ์ดั้งนั้นจึงเลือกวิธีการเติมค่าที่หายไปด้วยค่า False

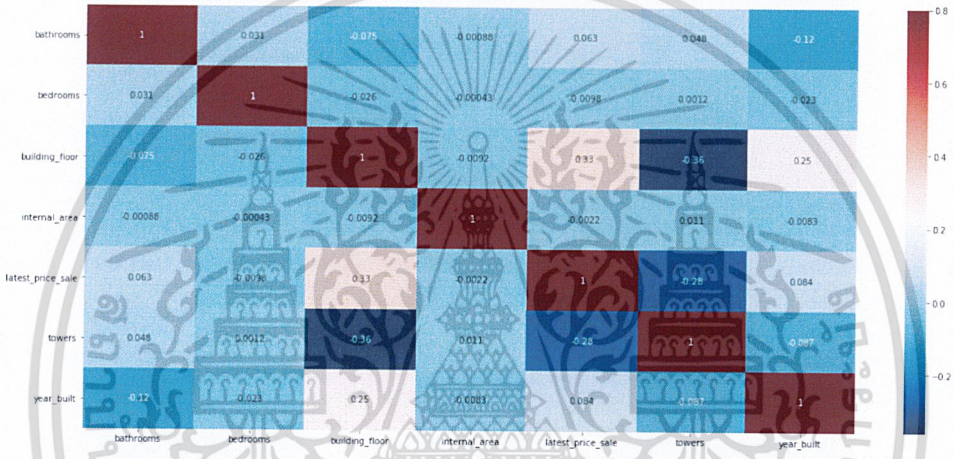
เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดูค่า correlation

```
latest_price_sale    1.000000
building_floor       0.334985
year_built           0.083720
bathrooms            0.063457
internal_area       -0.002249
bedrooms            -0.009822
towers              -0.282880
Name: latest_price_sale, dtype: float64
```

รูปที่ 3.23 คอลัมน์ที่มีค่า correlation ในช่วงราคา 1-5 ล้านบาท

พบว่าในช่วงราคา 1-5 ล้าน ตัวแปรชั้นของคอนโดมิเนียมมีความสัมพันธ์กับราคาขาย มากที่สุด 0.3



รูปที่ 3.24 แสดงแผนภาพคอลัมน์ที่มีค่า correlation ในช่วงราคา 1-5 ล้านบาท

เมื่อตรวจสอบค่า collinearity ของตัวแปรทั้งหมดในช่วงราคา 1-5 ล้านบาทพบว่าไปมีคอลัมน์ใดที่สัมพันธ์กันเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตัดข้อมูลที่ผิดปกติ (outliner)

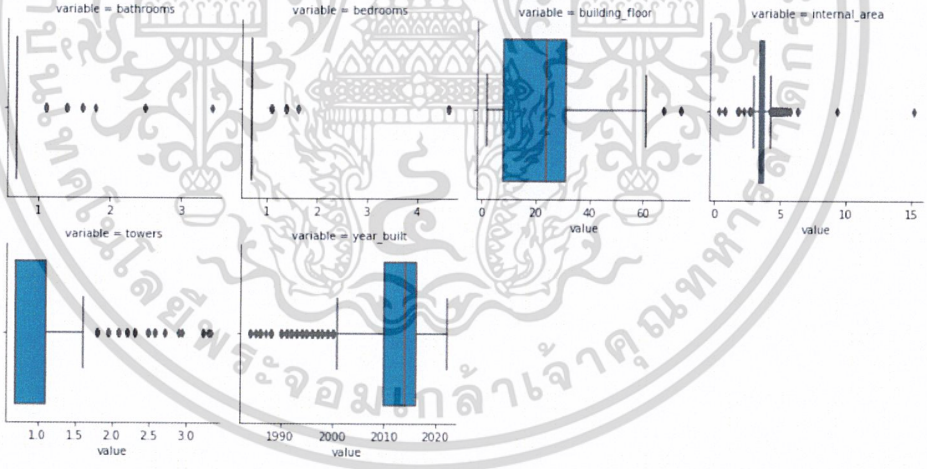
การตัด outliner ของข้อมูล จะทำให้โมเดลมีความแม่นยำมายิ่งขึ้น ตัดโดยวิธี interquartile range โดยดูจากความเบ้ของข้อมูล หากมีความเบ้มากกว่า 0.75 ทำการเทค log(1+X) เพื่อให้ข้อมูลมีการกระจายตัวแบบปกติก่อนนำไปตัด outliner นั่นเอง

```
ıt[20]:
```

	Skewed Features
internal_area	160.562505
bathrooms	36.562568
bedrooms	32.304043
towers	2.873440
building_floor	0.208336
latest_price_sale	-0.317384
year_built	-1.975118

รูปที่ 3.25 คอลัมน์ที่มีค่าความเบ้ ในช่วงราคา 1-5 ล้านบาท

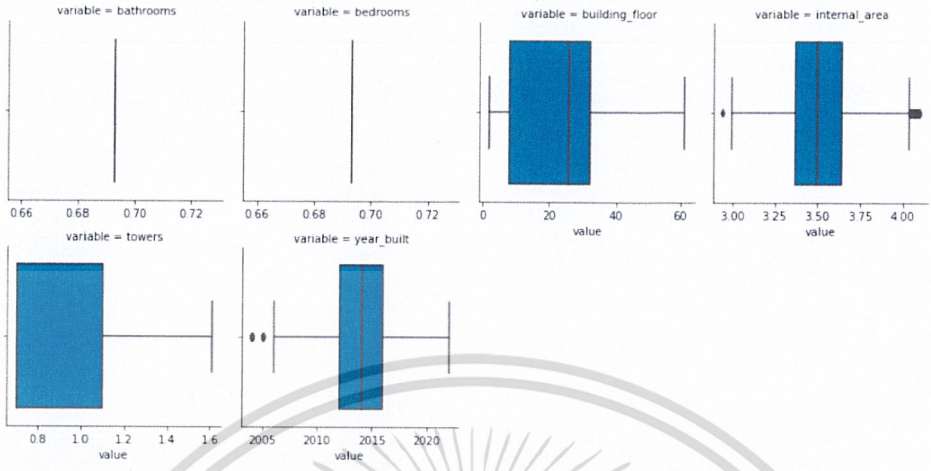
พล็อตกราฟ boxplot ก่อนตัด outliner ของข้อมูล bathrooms, bedrooms, building floor, internal area, towers



รูปที่ 3.26 ก่อนตัดค่าผิดปกติในช่วงราคา 1-5 ล้านบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พล็อตกราฟ boxplot หลังตัด outliner ของข้อมูล bathrooms, bedrooms, building floor, internal area, towers



รูปที่ 3.27 หลังตัดค่าผิดปกติในช่วงราคา 1-5 ล้านบาท

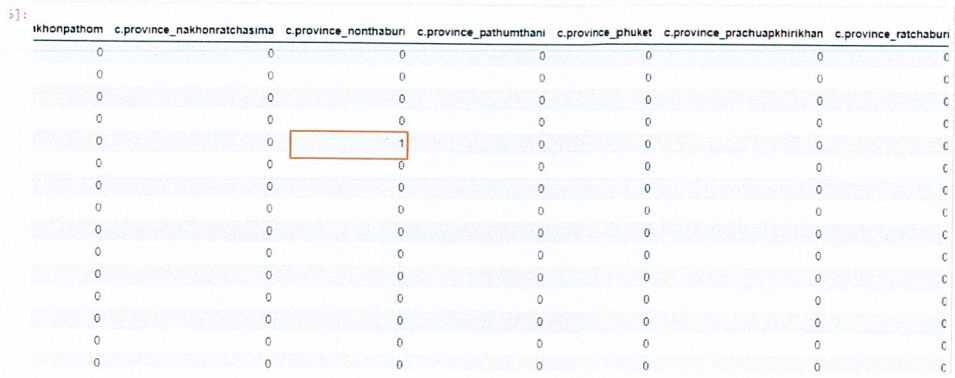
หลังจากนั้นนำตัวแปรออกจาก log(1+x)

การกำหนดชนิดของตัวแปร

- ตัวแปรห้องนอน ห้องน้ำ จำนวนอาคารทั้งหมดในโครงการ ตำแหน่งชั้นของคอนโดมิเนียม กำหนดชนิดของตัวแปรเป็น int
- ตัวแปรพื้นที่ภายใน ราขาขาย กำหนดชนิดของตัวแปรเป็น float
- ตัวแปรการอำนวนความสะดวกต่างๆ กำหนดชนิดของตัวแปรเป็น Boolean (True, False)

ในส่วนสุดท้ายก่อนนำข้อมูลเข้าโมเดลนั้นคือขั้นตอน Feature Engineering ในส่วนนี้จะมีการทำงาน 2 ส่วนคือ ข้อมูลที่เป็นหมวดหมู่เช่น เขต อำเภอ จังหวัด เราไม่สามารถนำข้อมูลดังกล่าวเข้าโมเดลได้จึงใช้เทคนิค one-hot encoding ในการแปลงคอลัมน์นั้นๆเป็นแถวต่างๆ และจะใส่ค่า 1 ในช่องที่มีในแถวนั้นและคอลัมน์อื่นๆจะใส่ค่า 0 นั่นเอง ใช้เทคนิค one-hot กับตัวแปร province, subdistrict, district

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ikhonpathom	c.province_nakhonratchasima	c.province_nonthaburi	c.province_pathumthani	c.province_phuket	c.province_prachuapkhirikhan	c.province_ratchaburi
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

รูปที่ 3.28 data farm การทำ one-hot encoding

ต่อมาคือการ Scaling data ของตัวแปรทั้งหมดที่เราจะนำไปพยากรณ์นั่นคือ เป็นการแปลงตัวเลขให้มี scale เท่ากันที่อยู่ในช่วง [0,1]

```
array([[0.        , 0.        , 0.08333333, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.54166667, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.40277778, ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.        , 0.        , 0.08333333, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.29166667, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.29166667, ..., 0.        , 0.        ,
        0.        ]])
```

รูปที่ 3.29 แสดงการทำ Scaling data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 Model Training ในช่วงราคา 1–5 ล้านบาท

เนื่องจากข้อมูลปัจจัยต่าง ๆ ที่นำมาใช้ สำหรับเป็นข้อมูลนำข้าโครงข่ายประสาทเทียมเพื่อสร้างโมเดลการพยากรณ์นั้นมีความแตกต่างกันใน เรื่องของชนิดข้อมูล หน่วยนับ และช่วงของข้อมูล ดังนั้นก่อนการนำข้อมูล เข้าสู่การสร้างโมเดลจึงจำเป็นต้องมีการจัดรูปแบบและปรับช่วงค่าของข้อมูลให้มีรูปแบบและ ค่าที่ใกล้เคียงกันโดยให้มีค่าอยู่ระหว่าง 0 -1 เพื่อให้ อยู่ในมาตรฐานเดียวกัน ซึ่งสมการที่ใช้สามารถแสดงได้ ดังสมการ

$$output = \frac{input - min}{max - min}$$

โดยที่ input คือ ข้อมูลที่นำเข้ามาคำนวณ

min คือ ค่าน้อยสุดของข้อมูลนั้นๆ

min คือ ค่ามากสุดของข้อมูลชุดนั้นๆ

ตารางที่ 3.6 แสดงปัจจัยและการเข้ารหัสของตัวแปร

ปัจจัย	ตัวแปร	การเข้ารหัส
จังหวัด	province	0,1
เขต	district	0,1
อำเภอ	subdistrict	0,1
การอำนวยความสะดวก	amenity	True,False
ห้องนอน	bathrooms	ข้อมูลเกิดจริง
ห้องน้ำ	bedrooms	ข้อมูลเกิดจริง
ตำแหน่งชั้นของคอนโดมิเนียม	building_floor	ข้อมูลเกิดจริง
พื้นที่ภายใน	internal_area	ข้อมูลเกิดจริง
อาคารทั้งหมดในโครงการ	towers	ข้อมูลเกิดจริง
ปีที่สร้าง	year_built	ข้อมูลเกิดจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการศึกษาโครงข่ายประสาทเทียม ซึ่งสามารถเขียนเป็น Script เป็น Python file เพื่อที่ออกแบบและเรียกใช้งานได้ตามต้องการ โดยจะอธิบายในส่วนของ Neural Network Training เพราะมีความสำคัญในการปรับค่าโครงข่าย (Rough-Tune) เพื่อที่จะได้โครงข่ายประสาทเทียมที่มีความเหมาะสมในการประมาณค่าฟังก์ชัน (Function Approximation) โดยมีส่วนที่น่าสนใจดังนี้

1. Algorithms ที่เกี่ยวข้องมีสามส่วน คือ

- Training คือ แนวความในการปรับค่าน้ำหนักและไปแอส เพื่อให้โครงข่ายสามรถเรียนรู้ความผิดพลาดและสามรถปรับตัวหาข้อมูลฝึกสอนได้
- Performance คือ ตัวชี้วัดของโครงข่ายขณะฝึกสอน ในที่นี้ใช้ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Squared Error หรือ MSE) เพื่อจะแยกแยะความสารถโครงข่ายได้ดียิ่งขึ้น
- Data division คือ การแบ่งข้อมูลในการฝึกสอน ทดสอบ และตรวจสอบความใช้ได้

2. Progress จะแสดงสถานะปัจจุบันของโครงข่าย โดยมีรายละเอียดดังนี้

- Epoch คือ จำนวนในการฝึกสอนสูงสุด ถ้าเกินค่านี้จะทำการหยุดฝึกสอน
- Batch Size คือ เวลาที่โครงข่ายประสาทเทียมใช้ในการฝึกสอนต่อ 1 epoch
- Performance คือ ตัวชี้วัดความสารถของโครงข่ายในการประมาณค่าฟังก์ชัน
- Gradient คือ ค่าความผิดพลาดในการหาคำตอบ โดยโครงข่ายประสาทเทียมจเปรียบเสมือนคนตาบอดที่เดินลงภูเขา (Error surface) โดยความสูง (Gradient) ของภูเขาจะหมายถึงความผิดพลาดในขณะนั้น ดังนั้น โครงข่ายประสาทเทียมจะหาคำตอบในทิศทางที่ลงจากภูเขา (ความชันลงมา) โดยเมื่อเจอความชันที่มากขึ้นจนไม่สามารถข้ามหรือเดินต่อไปได้ โครงข่ายประสาทเทียมจะหยุดการเรียนรู้



รูปที่ 3.30 การเจอคำตอบ Local ในโครงข่ายประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากการทำ Data preparation และ Feature Engineering มีข้อมูล 21,751 แถวและ 143 คอลัมน์ โดยเราจะกำหนด feature ของเรานั่นคือ ตัวแปรที่เราจะนำไปพยากรณ์ และ target คือราคาขาย จากนั่นทำการแบ่งข้อมูลออกเป็น 3 ส่วนคือ 1.สำหรับ train โมเดล 70 เปอร์เซ็นต์ 2.สำหรับ test โมเดล 15 เปอร์เซ็นต์ 3.สำหรับ validation โมเดล 15 เปอร์เซ็นต์ โดยโมเดลที่ใช้ในการ ทำนายคือ neural network กำหนด architecture ดังตารางดังต่อไปนี้

ตารางที่ 3.7 ตารางแสดง architecture ของโมเดลในช่วงราคา 1-5 ล้านบาท

	neurons	activation
Input Layer	128	tanh
Hidden layer	128	tanh
	256	tanh
	128	tanh
	128	ReLU
	64	ReLU
	32	ReLU
	4	ReLU
Output Layer	1	ReLU

โดยเลือก optimizer = adam, loss = mean_square_error, epoch = 300, batch size = 256 ในแต่ละ dense จะมี kernel_regularizer=regularizers.l2(0.01) ในทุกๆ ชั้นหลังจากนั้นทำการ train หลังจากนั้นฝึกสอนโมเดล (train model) ดังรูป

```
In [215]:  1 prediction_1 = Prediction('predict_price_model_class0_edit.pkl', preprocess_df_class0)
           2 prediction_1.train()

Train on 15225 samples, validate on 4568 samples
Epoch 1/300
15225/15225 [==============================] - 15s 967us/step - loss: 3480642.8333 - val_loss: 3454141.9536
Epoch 2/300
15225/15225 [==============================] - 1s 59us/step - loss: 3428270.3157 - val_loss: 3302191.7198
Epoch 3/300
15225/15225 [==============================] - 1s 56us/step - loss: 2989742.4848 - val_loss: 2395326.2785
Epoch 4/300
15225/15225 [==============================] - 1s 62us/step - loss: 1498529.8533 - val_loss: 819749.0316
Epoch 5/300
15225/15225 [==============================] - 1s 58us/step - loss: 831871.1763 - val_loss: 815920.9526
Epoch 6/300
15225/15225 [==============================] - 1s 57us/step - loss: 823610.3419 - val_loss: 813901.4933
Epoch 7/300
15225/15225 [==============================] - 1s 57us/step - loss: 819278.9891 - val_loss: 737901.9529
Epoch 8/300
15225/15225 [==============================] - 1s 61us/step - loss: 635036.4146 - val_loss: 587467.3381
Epoch 9/300
15225/15225 [==============================] - 1s 59us/step - loss: 554978.9098 - val_loss: 531647.3690
Epoch 10/300
```

รูปที่ 3.31 train โมเดลในช่วงราคา 1-5 ล้านบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 โมเดลการพยากรณ์ราคาประเมินในช่วงราคา 5-10 ล้านบาท

3.5.1 การทำความเข้าใจข้อมูล (Data Understanding)

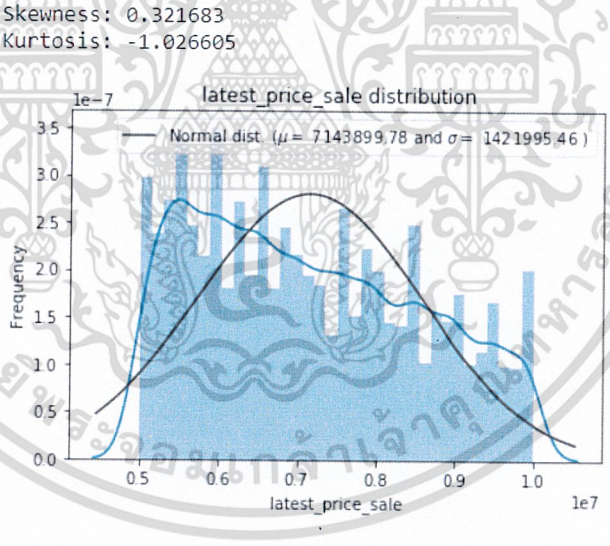
โดยเริ่มจาก Exploratory data analysis (EDA) เพื่อให้เราเข้าใจในลักษณะของข้อมูลในช่วงราคา 5-10 ล้านบาท ในช่วงราคานี้มีข้อมูลทั้งหมด 32,217 แถว และ 22 คอลัมน์ ดูค่าสถิติเบื้องต้นดังนี้

:

	bathrooms	bedrooms	building_floor	internal_area	latest_price_sale	towers	year_built
count	32217	32217	32217	32217	32217	32173	31279
mean	1	1	29	254	7143899	1	2012
std	0	1	14	36213	1422017	1	5
min	1	1	2	1	5000000	1	1981
25%	1	1	22	35	5900000	1	2010
50%	1	1	31	46	6990000	1	2014
75%	1	2	40	61	8300000	2	2017
max	52	99	74	6500000	10000000	20	2023

รูปที่ 3.32 แสดงค่าสถิติเบื้องต้นในช่วงราคา 5-10 ล้านบาท

วิเคราะห์ตัวแปร latest price sale

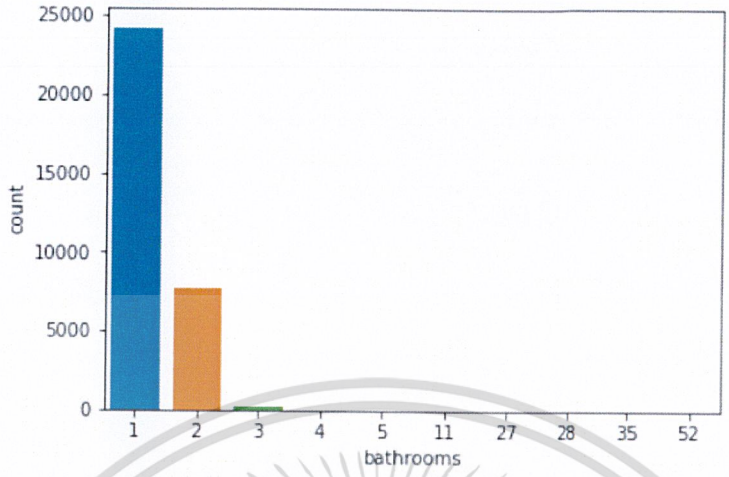


รูปที่ 3.33 แสดงการกระจายตัวของราคาในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่า ช่วงราคา 5-10 ล้านบาทมีการกระจายตัวใกล้เคียงการกระจายแบบปกติ นั่นทำให้เกิดข้อผิดพลาดในการพยากรณ์น้อยลง

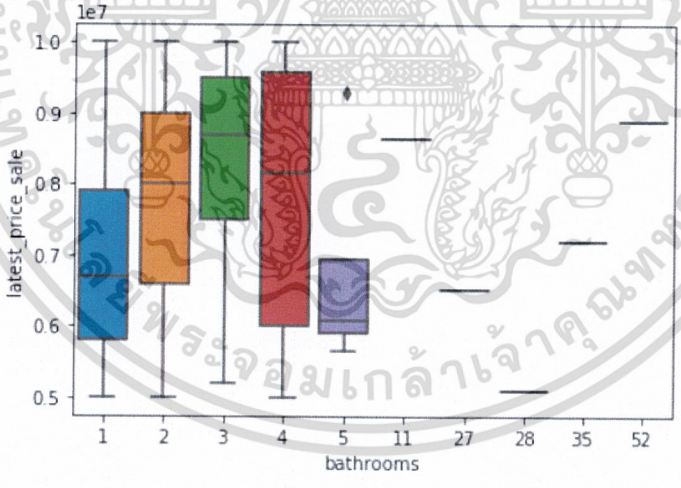
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร bathrooms



รูปที่ 3.34 แสดงจำนวนห้องอาบน้ำในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 5-10 ล้านบาท ที่ 1 ห้องอาบน้ำมีจำนวน 24,233 ห้อง, 2 ห้องอาบน้ำมีจำนวน 7705 ห้อง, 3 ห้องอาบน้ำมีจำนวน 252 ห้อง, 4 ห้องอาบน้ำมีจำนวน 28 ห้อง, 5 ห้องอาบน้ำมีจำนวน 4 ห้อง, 11 27 28 35 และ 52 ห้องอาบน้ำมีจำนวน 1 ห้อง

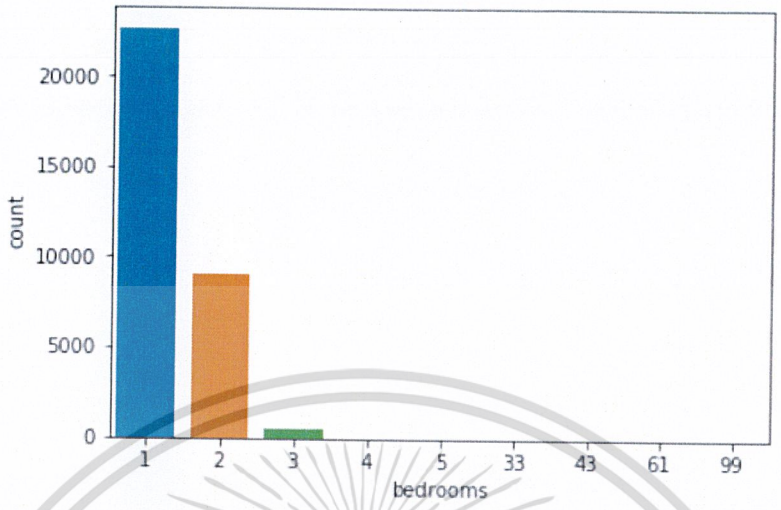


รูปที่ 3.35 แสดงจำนวนห้องนอนเมื่อเทียบกับราคาในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่าตัวแปรห้องน้ำเมื่อเทียบกับราคาพบว่าส่วนใหญ่ ช่วงราคา 5-10 ล้านบาท จะมี 1-5 ห้องน้ำ และที่มากกว่า 11 ห้องน้ำจะมีเพียงเล็กน้อย

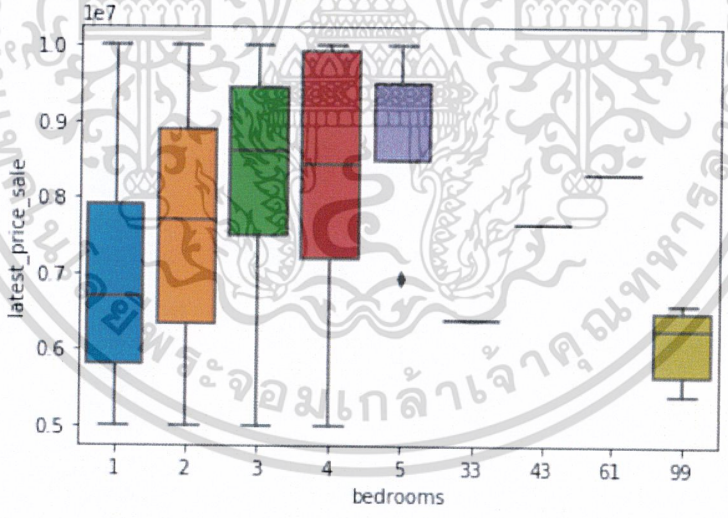
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร bedrooms



รูปที่ 3.36 แสดงจำนวนห้องนอนในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 5-10 ล้านบาท ที่ 1 ห้องนอนมีจำนวน 22,677 ห้อง, 2 ห้องนอนมีจำนวน 9,027 ห้อง, 3 ห้องนอนมีจำนวน 485 ห้อง, 4 ห้องนอนมีจำนวน 14 ห้อง

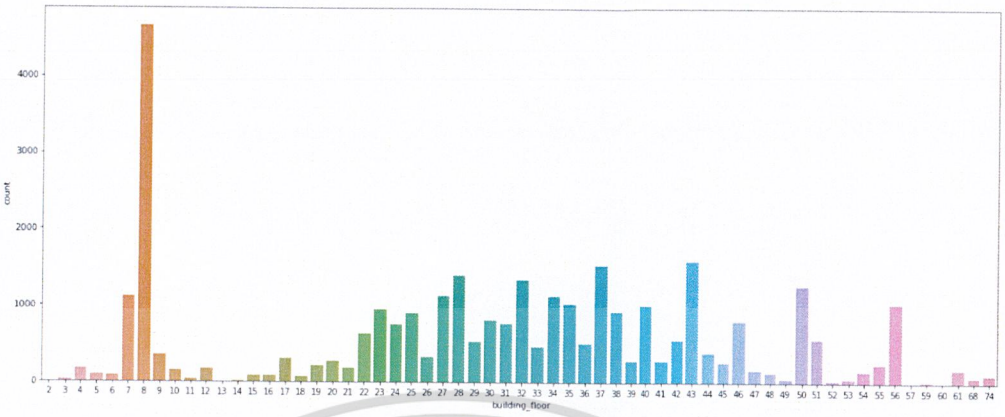


รูปที่ 3.37 แสดงจำนวนห้องนอนในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่าตัวแปรห้องนอนเมื่อเทียบกับราคาพบว่าส่วนใหญ่ ช่วงราคา 1-5 ล้านบาท จะมี 1-4 ห้องนอน และที่ 99 ห้องนอนเมื่อลองตรวจสอบแล้วพบว่าไม่มีจริง ส่วนนี้เราจะตัดค่าผิดปกติออกไป

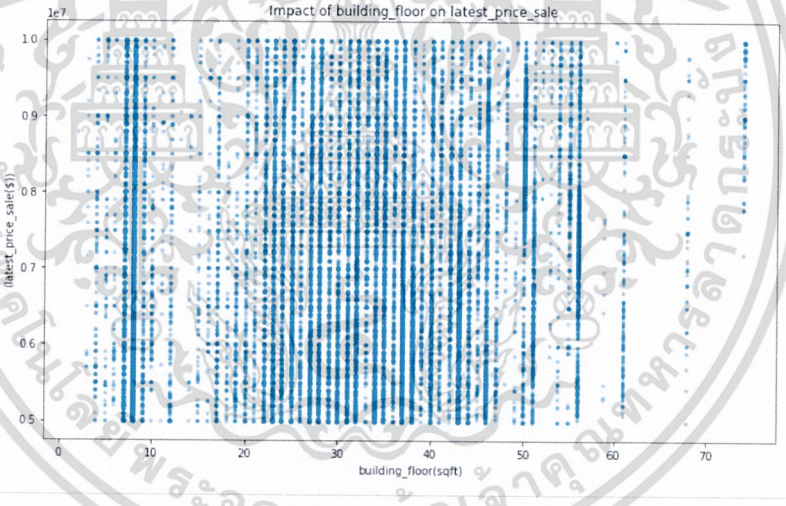
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร building floor



รูปที่ 3.38 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 5-10 ล้านบาท พบว่าชั้นที่ 8 มีจำนวนคอนโดมิเนียมมากสุดถึง 4,655 ห้องรองลงมาคือชั้นที่ 28, 27 ,32 จะอยู่ที่ประมาณ 1,300-1,500 ห้อง

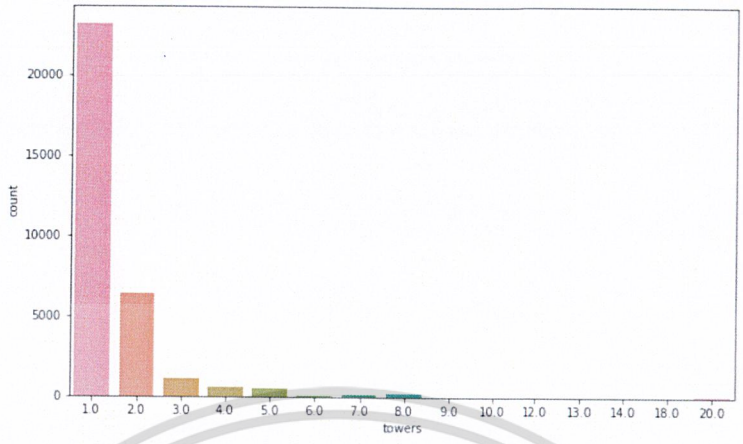


รูปที่ 3.39 แสดงจำนวนอาคารรวมของคอนโดมิเนียมเทียบกับราคาในช่วง 5-10 ล้านบาท

จากภาพการเปรียบเทียบระหว่าชั้นคอนโดมิเนียมและราคาขายในช่วงราคา 5-10 ล้านบาท พบว่าชั้นที่ 20-50 ชั้นจะอยู่ในช่วงราคา 5-10 ล้านบาท และชั้น 60-70 จะอยู่ในช่วงราคา 8 ล้านบาทขึ้นไป

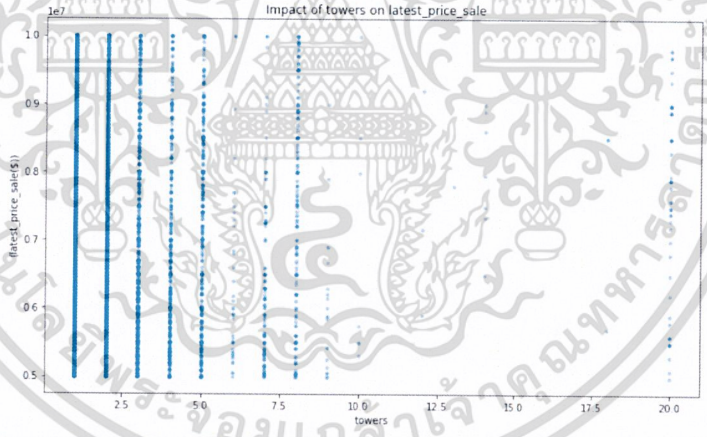
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร towers



รูปที่ 3.40 แสดงจำนวนชั้นของคอนโดมิเนียมในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 5-10 ล้านบาท อาคารรวมของคอนโดมิเนียมส่วนมากจะอยู่ที่ 1 -3 อาคารเป็นส่วนใหญ่โดยที่ 1 อาคารมีจำนวน 23,198 อาคาร, 2 อาคารมีจำนวน 6,403 อาคาร, 3 อาคารมีจำนวน 1,073 อาคาร

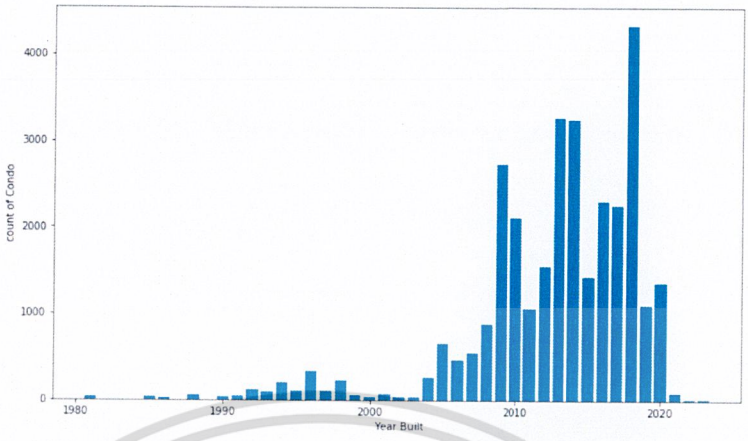


รูปที่ 3.41 แสดงจำนวนชั้นของคอนโดมิเนียมเทียบกับราคาในช่วง 5-10 ล้านบาท

จากภาพการเปรียบเทียบระหว่างอาคารรวมของคอนโดมิเนียมและราคาขายในช่วงราคา 5-10 ล้านบาท จะเห็นได้ว่า 1-5 อาคารรวมของคอนโดมิเนียมมีราคาตั้งแต่ 5-10 ล้านบาท และส่วนอาคารรวมของคอนโดมิเนียมที่มีมากกว่า 10-20 อาคาร

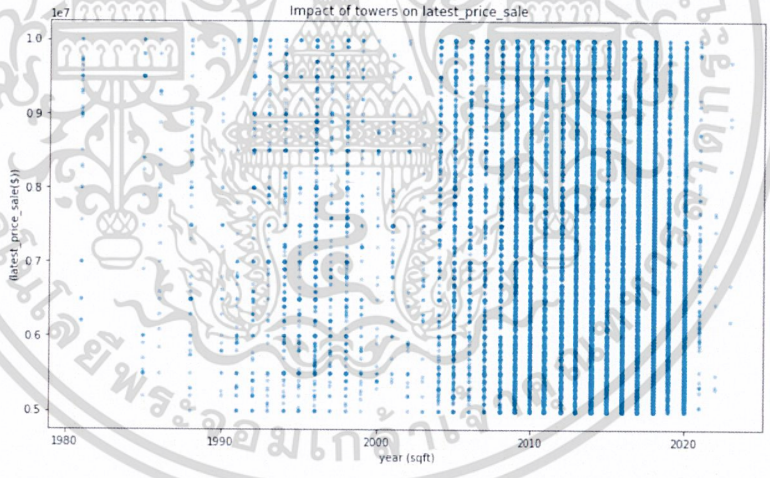
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร year built



รูปที่ 3.42 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมในช่วง 5-10 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 5-10 ล้านบาท ปีที่สร้างคอนโดมิเนียมมากที่สุดคือ 2018 จำนวน 4328 ห้องรองลงมาคือปี 2013 คือ 3269 ห้อง เมื่อดูแนวโน้มจะเห็นว่าจะอยู่ในช่วงระหว่างปี 2008-2018 และที่กำลังสร้างในอนาคต



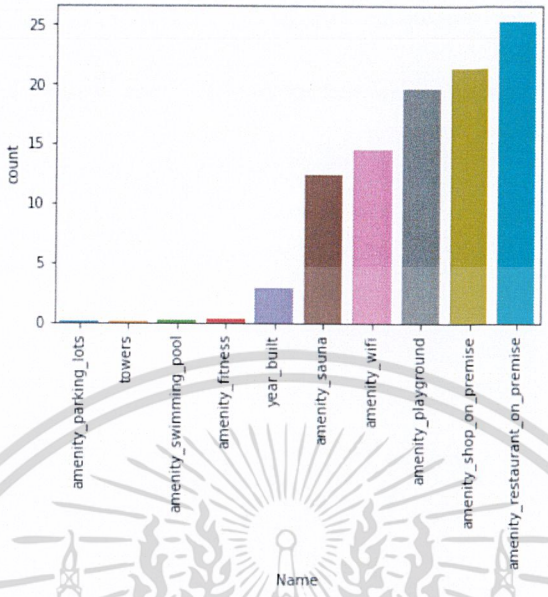
รูปที่ 3.43 แสดงจำนวนปีที่สร้างของคอนโดมิเนียมเทียบกับราคาในช่วง 5-10 ล้านบาท

จากภาพการเปรียบเทียบระหว่างปีที่สร้างคอนโดและราคาขายเห็นได้ว่าช่วงราคา 5-10 ล้านบาท พบว่าช่วงปี 2010-2010 ที่สร้างคอนโดมิเนียมมีช่วงราคาตั้งแต่ 5-10 ล้านบาทและนอกจากนั้นจะมีการการจายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 การเตรียมข้อมูล (Data Preparation)

วิเคราะห์จำนวนข้อมูลที่หายไป



รูปที่ 3.44 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลในช่วงราคา 5-10 ล้านบาท

	Total	Percent
amenity_restaurant_on_premise	8155	25.312723
amenity_shop_on_premise	6881	21.358289
amenity_playground	6351	19.713195
amenity_wifi	4674	14.507869
amenity_sauna	4028	12.502716
year_built	938	2.911506
amenity_fitness	118	0.366266
amenity_swimming_pool	64	0.198653
towers	44	0.136574
amenity_parking_lots	34	0.105534
amenity_bbq_area	14	0.043455
amenity_24_hours_security	7	0.021728
amenity_cctv	5	0.015520
amenity_elevator	4	0.012416
subdistrict	0	0.000000
province	0	0.000000
latest_price_sale	0	0.000000

รูปที่ 3.45 คอลัมน์และจำนวนข้อมูลที่หายไปในช่วงราคา 5-10 ล้านบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

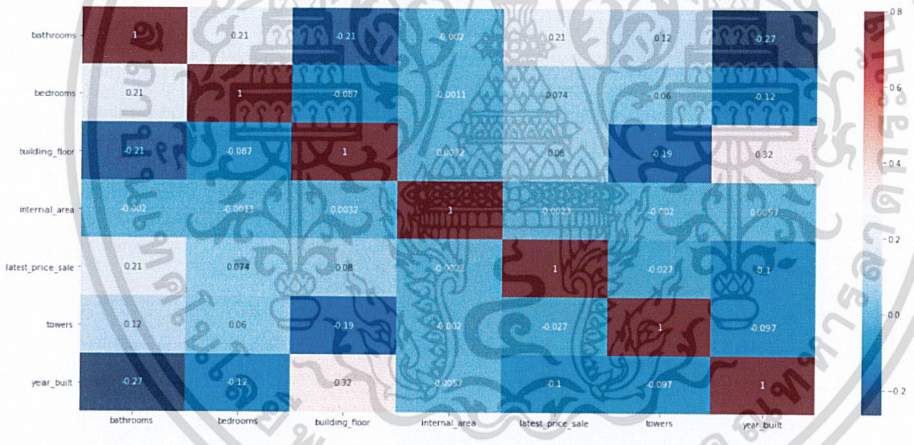
จากข้างต้นจะเห็นว่า year built มี 2.91 เปอร์เซ็นต์ และ towers มี 0.13 เปอร์เซ็นต์ของจำนวนที่หายไปดังนั้นจึงเลือกวิธีการลบคอลัมน์นั้นทิ้งไปเนื่องจากหากเติมค่านั้นอาจทำให้ข้อมูลผิดพลาดได้ และในส่วนของ amenity ทั้งหมดหารลบคอลัมน์ที่เป็น NaN ทิ้งไปอาจทำให้ข้อมูลนั้นไม่พียงพแต่อการพยากรณ์ดั้งนั้นจึงเลือกวิธีการเติมค่าที่หายไปด้วยค่า False

ดูค่า correlation

```
latest_price_sale     1.000000
bathrooms             0.207467
building_floor        0.079738
bedrooms              0.073729
internal_area        -0.002313
towers               -0.027435
year_built           -0.102635
Name: latest_price_sale, dtype: float64
```

รูปที่ 3.46 คอลัมน์ที่มีค่า correlation ในช่วงราคา 5-10 ล้านบาท

พบว่าในช่วงราคา 5-10 ล้าน ตัวห้องน้ำของคอนโดมิเนียมมีความสัมพันธ์กับราคาขาย มากที่สุด 0.2



รูปที่ 3.47 แสดงแผนภาพคอลัมน์ที่มีค่า correlation ในช่วงราคา 5-10 ล้านบาท

เมื่อตรวจสอบค่า collinearity ของตัวแปรทั้งหมดในช่วงราคา 5-10 ล้านบาทพบว่าไปมีคอลัมน์ใดที่สัมพันธ์กันเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัดข้อมูลที่ผิดปกติ (outlier)

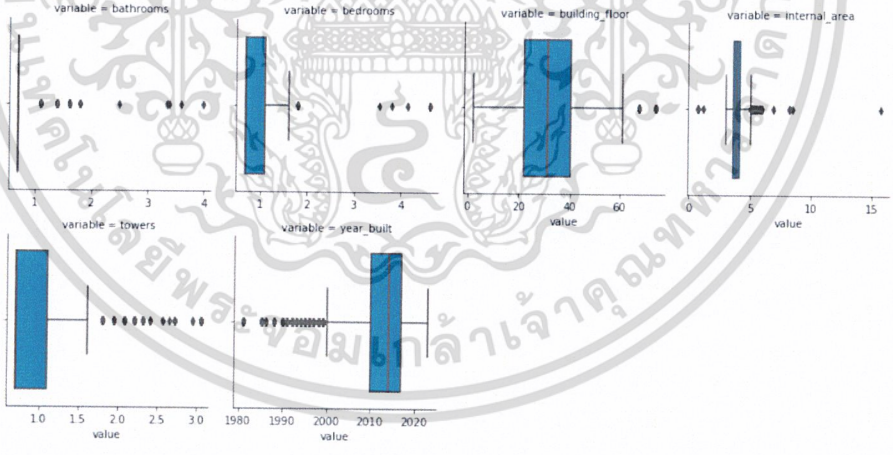
การตัด outlier ของข้อมูล จะทำให้โมเดลมีความแม่นยำมายิ่งขึ้น ตัดโดยวิธี interquartile range โดยดูจากความเบ้ของข้อมูล หากมีความเบ้มากกว่า 0.75 ทำการเทค log(1+X) เพื่อให้ข้อมูลมีการกระจายตัวแบบปกติก่อนนำไปตัด outliner นั่นเอง

|:

	Skewed Features
internal_area	179.482198
bedrooms	55.399719
bathrooms	28.537560
towers	6.958352
latest_price_sale	0.321668
building_floor	-0.054992
year_built	-1.671994

รูปที่ 3.48 คอลัมน์ที่มีค่าความเบ้ ในช่วงราคา 5-10 ล้านบาท

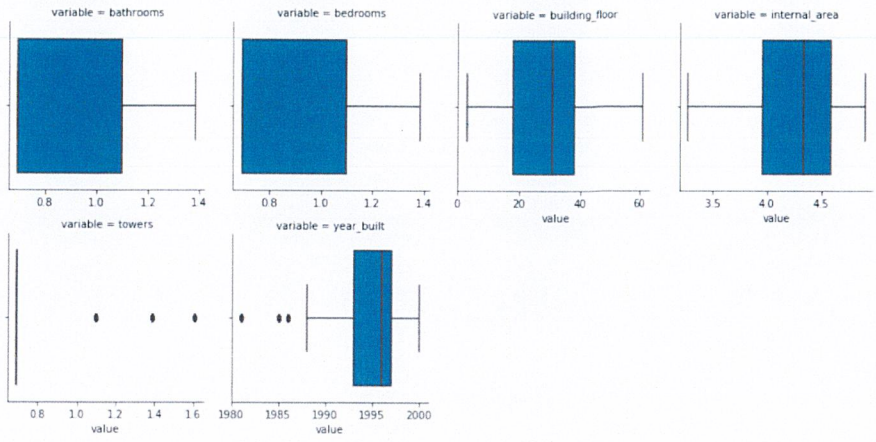
พล็อตกราฟ boxplot ก่อนตัด outlier ของข้อมูล bathrooms, bedrooms, building floor, internal area, towers



รูปที่ 3.49 ก่อนตัดค่าผิดปกติในช่วงราคา 5-10 ล้านบาท

พล็อตกราฟ boxplot หลังตัด outlier ของข้อมูล bathrooms, bedrooms, building floor, internal area, towers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.50 หลังตัดค่าผิดปกติในช่วงราคา 5-10 ล้านบาท

หลังจากนั้นนำตัวแปรออกจาก log(1+x)

การกำหนดชนิดของตัวแปร

- ตัวแปรห้องนอน ห้องน้ำ จำนวนอาคารทั้งหมดในโครงการ ตำแหน่งชั้นของคอนโดมิเนียม กำหนดชนิดของตัวแปรเป็น int
- ตัวแปรพื้นที่ภายใน ราขาขาย กำหนดชนิดของตัวแปรเป็น float
- ตัวแปรการอำนวนความสะดวกต่างๆ กำหนดชนิดของตัวแปรเป็น Boolean (True, False)

ในส่วนสุดท้ายก่อนนำข้อมูลเข้าโมเดลนั้นคือขั้นตอน Feature Engineering ในส่วนนี้จะมีการทำงาน 2 ส่วนคือ ข้อมูลที่เป็นหมวดหมู่เช่น เขต อำเภอ จังหวัด เราไม่สามารถนำข้อมูลดังกล่าวเข้าโมเดลได้จึงใช้เทคนิค one-hot encoding ในการแปลงคอลัมน์นั้นๆเป็นแถวต่างๆ และจะใส่ค่า 1 ในช่องที่มีในแถวนั้นและคอลัมน์อื่นๆจะใส่ค่า 0 นั่นเอง ใช้เทคนิค one-hot กับตัวแปร province, subdistrict, district ต่อมาคือการ Scaling data ของตัวแปรทั้งหมดที่เราจะนำไปพยากรณ์นั่นคือเป็นการแปลงตัวเลขให้มี scale เท่ากันที่อยู่ในช่วง [0,1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 Model Training ในช่วงราคา 5-10 ล้านบาท

หลังจากการทำ Data preparation และ Feature Engineering มีข้อมูล 16,228 แถวและ 93 คอลัมน์ โดยเราจะกำหนด feature ของเรานั่นคือ ตัวแปรที่เราจะนำไปพยากรณ์ และ target คือ ราคาขาย จากนั่นทำการแบ่งข้อมูลออกเป็น 3 ส่วนคือ 1.สำหรับ train โมเดล 70 เปอร์เซ็นต์ 2. สำหรับ test โมเดล 15 เปอร์เซ็นต์ 3.สำหรับ validation โมเดล 15 เปอร์เซ็นต์ โดยโมเดลที่ใช้ในการทำนายคือ neural network กำหนด architecture ดังตารางดังต่อไปนี้

ตารางที่ 3.8 ตารางแสดง architecture ของโมเดลในช่วงราคา 5-10 ล้านบาท

	neurons	activation
Input Layer	128	tanh
Hidden layer	128	tanh
	256	ReLU
	128	ReLU
	128	ReLU
	64	ReLU
	32	ReLU
	4	ReLU
Output Layer	1	ReLU

โดยเลือก optimizer = adam, loss = mean_square_error, epoch = 300, batch size = 256 ในแต่ละ dense จะมี kernel_regularizer=regularizers.l2(0.01) ในทุกๆ ชั้นหลังจากนั้นทำการ train หลังจากนั้นฝึกสอนโมเดล (train model) ดังรูป

```
In [135]:  1 prediction_2 = Prediction('predict_price_model_class1_edit.pkl', preprocess_df_class1)
           2 prediction_2.train_model2()

Train on 11359 samples, validate on 3408 samples
Epoch 1/300
11359/11359 [==============================] - 5s 478us/step - loss: 6982918.3313 - val_loss: 6959197.3897
Epoch 2/300
11359/11359 [==============================] - 0s 42us/step - loss: 6982139.3793 - val_loss: 6953534.5775
Epoch 3/300
11359/11359 [==============================] - 0s 39us/step - loss: 6835264.8330 - val_loss: 6249924.9272
Epoch 4/300
11359/11359 [==============================] - 0s 41us/step - loss: 3187094.3175 - val_loss: 1145853.5305
Epoch 5/300
11359/11359 [==============================] - 0s 41us/step - loss: 1117215.2136 - val_loss: 1113668.6549
Epoch 6/300
11359/11359 [==============================] - 0s 43us/step - loss: 1108085.1464 - val_loss: 1113100.2981
Epoch 7/300
11359/11359 [==============================] - 0s 43us/step - loss: 1108124.0993 - val_loss: 1114668.0951
Epoch 8/300
11359/11359 [==============================] - 0s 39us/step - loss: 1108245.1344 - val_loss: 1114175.5153
Epoch 9/300
11359/11359 [==============================] - 0s 37us/step - loss: 1108420.8387 - val_loss: 1113201.4765
Epoch 10/300
```

รูปที่ 3.51 train โมเดลในช่วงราคา 5-10 ล้านบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 โมเดลการพยากรณ์ราคาประเมินในช่วงราคา 10-20 ล้านบาท

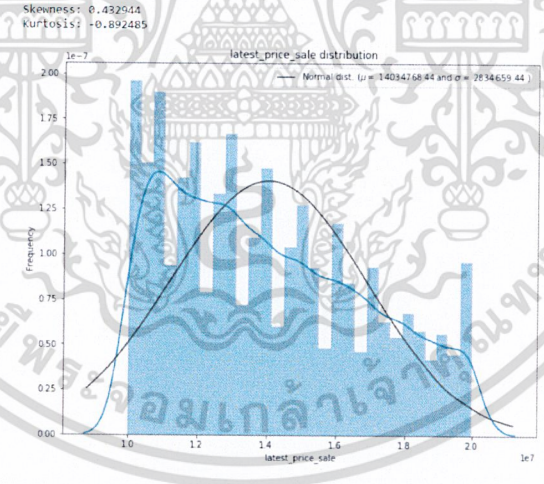
3.6.1 การทำความเข้าใจข้อมูล (Data Acquisition & Understanding)

โดยเริ่มจากExploratory data analysis (EDA) เพื่อให้เราเข้าใจในลักษณะของข้อมูลในช่วงราคา 10-20 ล้านบาท ในช่วงราคานี้มีข้อมูลทั้งหมด 17,588 แถว และ 22 คอลัมน์ ดูค่าสถิติเบื้องต้นดังนี้

	bathrooms	bedrooms	building_floor	internal_area	latest_price_sale	towers	year_built
count	17588	17588	17588	17588	17588	17535	17246
mean	1	1	33	88	14034768	1	2010
std	0	1	15	44	2834740	1	6
min	1	1	3	1	10000000	1	1981
25%	1	1	24	60	11540000	1	2009
50%	2	2	33	76	13500000	1	2012
75%	2	2	43	104	16120000	2	2016
max	7	99	74	600	20000000	20	2023

รูปที่ 3.52 แสดงค่าสถิติเบื้องต้นในช่วงราคา 10-20 ล้านบาท

วิเคราะห์ตัวแปร latest price sale

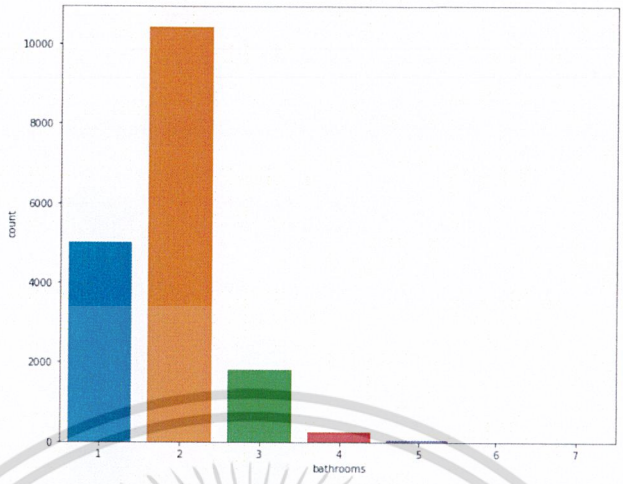


รูปที่ 3.53 แสดงการกระจายตัวของราคาในช่วง 10-20 ล้านบาท

จากภาพเห็นได้ว่า ช่วงราคา 10-20 ล้านบาทมีการกระจายตัวใกล้เคียงการกระจายแบบปกติ นั่นทำให้เกิดข้อผิดพลาดในการพยากรณ์น้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร bathrooms



รูปที่ 3.54 แสดงจำนวนห้องอาบน้ำในช่วง 10-20 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 10-20 ล้านบาท ที่ 1 ห้องอาบน้ำมีจำนวน 10,411 ห้อง, 2 ห้องอาบน้ำมีจำนวน 5,024 ห้อง, 3 ห้องอาบน้ำมีจำนวน 1,818 ห้อง, 4 ห้องอาบน้ำมีจำนวน 256 ห้อง, 5 ห้องอาบน้ำมีจำนวน 41 ห้อง

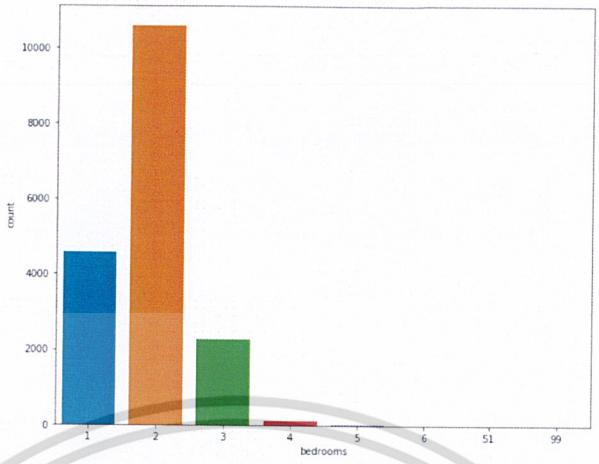


รูปที่ 3.55 แสดงจำนวนห้องอาบน้ำเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท

จากภาพเห็นได้ว่าตัวแปรห้องน้ำเมื่อเทียบกับราคาพบว่าส่วนใหญ่ ช่วงราคา 10-20 ล้านบาท จะมี 1-7 ห้องน้ำ

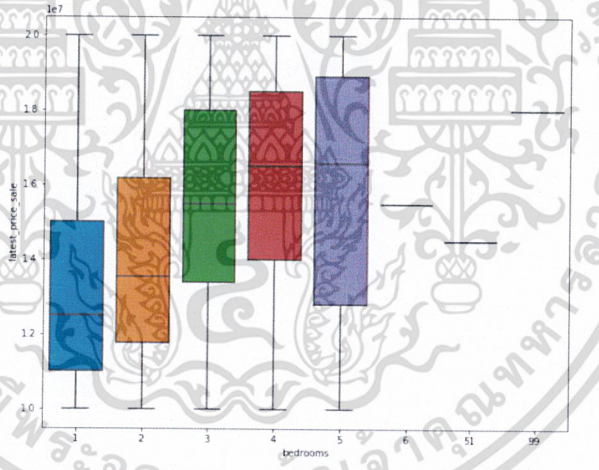
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร bedrooms



รูปที่ 3.56 แสดงจำนวนห้องนอนในช่วง 10-20 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 10-20 ล้านบาท ที่ 1 ห้องนอนมีจำนวน 10,593 ห้อง, 2 ห้องนอนมีจำนวน 4,584 ห้อง, 3 ห้องนอนมีจำนวน 3,378 ห้อง, 4 ห้องนอนมีจำนวน 121 ห้อง

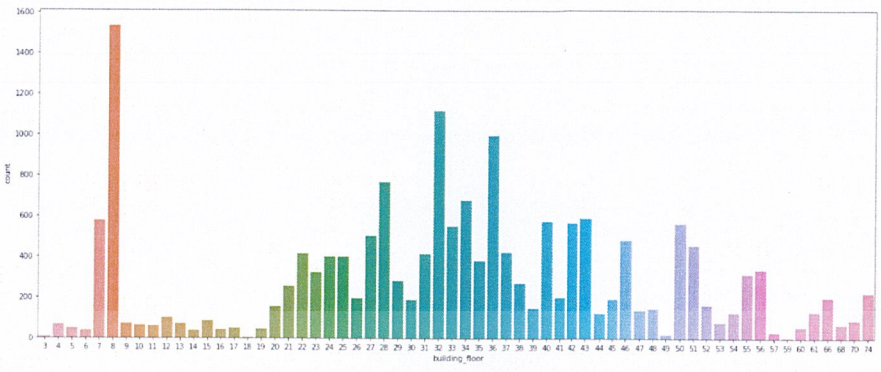


รูปที่ 3.57 แสดงจำนวนห้องนอนเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท

จากภาพเห็นได้ว่าตัวแปรห้องนอนเมื่อเทียบกับราคาพบว่าส่วนใหญ่ ช่วงราคา 10-20 ล้านบาท จะมี 1-5 ห้องนอน และที่ 51,99 ห้องนอนเมื่อลองตรวจสอบแล้วพบว่าไม่มีจริง ส่วนนี้เราจะตัดค่าผิดปกติออกไป

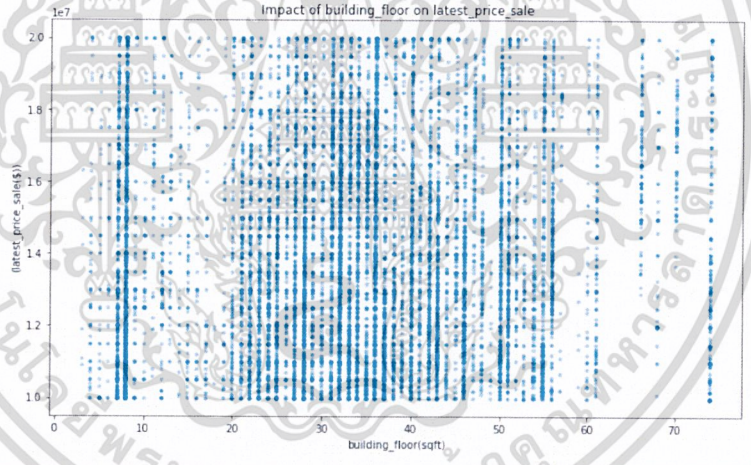
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร building floor



รูปที่ 3.58 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมในช่วงราคา 10-20 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 10-20 ล้านบาท พบว่าชั้นที่ 8 มีจำนวนคอนโดมิเนียมมากสุดถึง 1,534 ห้องรองลงมาคือชั้นที่ 28, 27, 32, 34 จะอยู่ที่ประมาณ 600-1,000 ห้อง

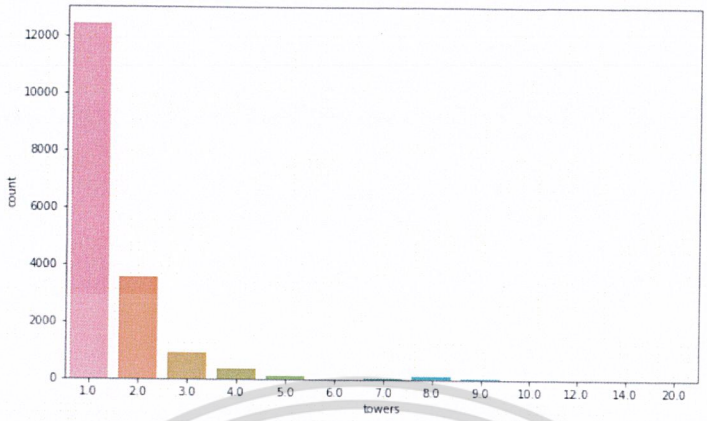


รูปที่ 3.59 แสดงจำนวนชั้นต่างๆของคอนโดมิเนียมเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท

จากภาพการเปรียบเทียบระหว่าชั้นคอนโดมิเนียมและราคาขายในช่วงราคา 10-20 ล้านบาท พบว่าชั้นที่ 20-60 ชั้นจะอยู่ในช่วงราคา 10-20 ล้านบาท และชั้น 60-70 จะอยู่ในช่วงราคา 12 ล้านบาทขึ้นไป

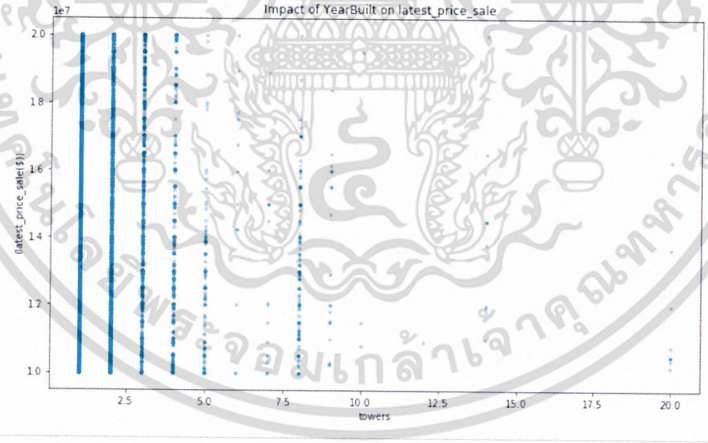
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร towers



รูปที่ 3.60 แสดงจำนวนอาคารรวมของคอนโดมิเนียมในช่วง 10-20 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 10-20 ล้านบาท อาคารรวมของคอนโดมิเนียมส่วนมากจะอยู่ที่ 1 -4 อาคารเป็นส่วนใหญ่โดยที่ 1 อาคารมีจำนวน 12,434 อาคาร, 2 อาคารมีจำนวน 3,546 อาคาร, 3 อาคารมีจำนวน 909 อาคาร

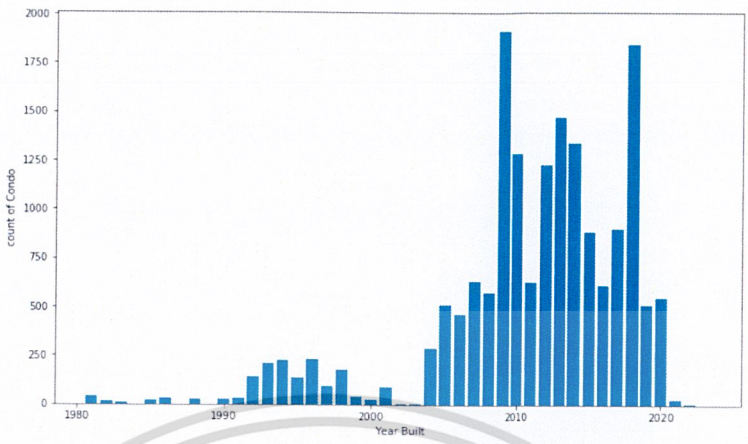


รูปที่ 3.61 แสดงจำนวนอาคารรวมของคอนโดมิเนียมเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท

จากภาพการเปรียบเทียบระหว่างอาคารรวมของคอนโดมิเนียมและราคาขายในช่วงราคา 10-20 ล้านบาท จะเห็นได้ว่า 1-5 อาคารวมของคอนโดมิเนียมมีราคาตั้งแต่ 10-20 ล้านบาท

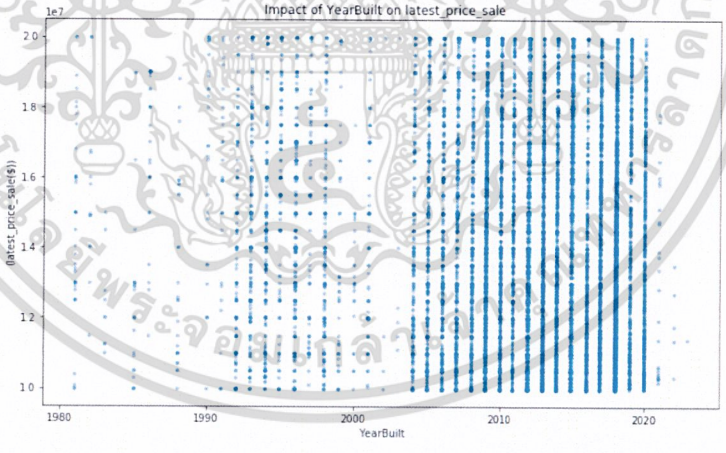
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ตัวแปร year built



รูปที่ 3.62 แสดงจำนวนปีที่สร้างคอนโดมิเนียมในช่วง 10-20 ล้านบาท

จากภาพเห็นได้ว่าช่วงราคา 10-20 ล้านบาท ปีที่สร้างคอนโดมิเนียมมากที่สุดคือ 2009 จำนวน 1,914 ห้องรองลงมาคือปี 2018 คือ 1,845 ห้อง เมื่อดูแนวโน้มจะเห็นว่าจะอยู่ในช่วงระหว่างปี 2010-2020 และที่กำลังสร้างในอนาคต



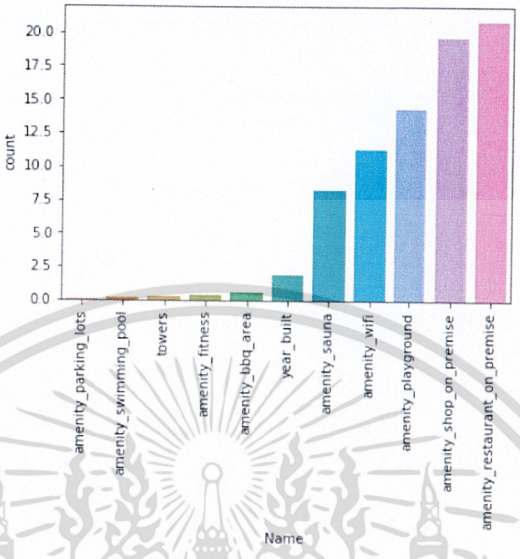
รูปที่ 3.63 แสดงจำนวนปีที่สร้างคอนโดมิเนียมเมื่อเทียบกับราคาในช่วง 10-20 ล้านบาท

จากภาพการเปรียบเทียบระหว่างปีที่สร้างคอนโดและราคาขายเห็นได้ว่าช่วงราคา 10-20 ล้านบาท พบว่าช่วงปี 2010-2010 ที่สร้างคอนโดมิเนียมมีช่วงราคาตั้งแต่ 10-20 ล้านบาทและนอกจากนั้นจะมีการการจายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 การเตรียมข้อมูล (Data Preparation)

วิเคราะห์จำนวนข้อมูลที่หายไป



รูปที่ 3.64 กราฟแท่งจำนวนคอลัมน์ที่หายไปของข้อมูลในช่วงราคา 10-20 ล้านบาท

]:

	Total	Percent
amenity_restaurant_on_premise	3677	20.906300
amenity_shop_on_premise	3453	19.632704
amenity_playground	2519	14.322265
amenity_wifi	1990	11.314533
amenity_sauna	1444	8.210143
year_built	342	1.944508
amenity_bbq_area	97	0.551512
amenity_fitness	67	0.380942
towers	53	0.301342
amenity_swimming_pool	27	0.153514
amenity_parking_lots	14	0.079600
amenity_24_hours_security	5	0.028428
amenity_cctv	5	0.028428
amenity_elevator	5	0.028428
subdistrict	0	0.000000
province	0	0.000000
latest_price_sale	0	0.000000

รูปที่ 3.65 คอลัมน์และจำนวนข้อมูลที่หายไปในช่วงราคา 10-20 ล้านบาท

จากข้างต้นจะเห็นว่า year built มี 1.94 เปอร์เซ็นต์ และ towers มี 0.3 เปอร์เซ็นต์ของจำนวนที่หายไปดังนั้นจึงเลือกวิธีการลบคอลัมน์นั้นทิ้งไปเนื่องจากหากเติมค่านั้นอาจทำให้ข้อมูลผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

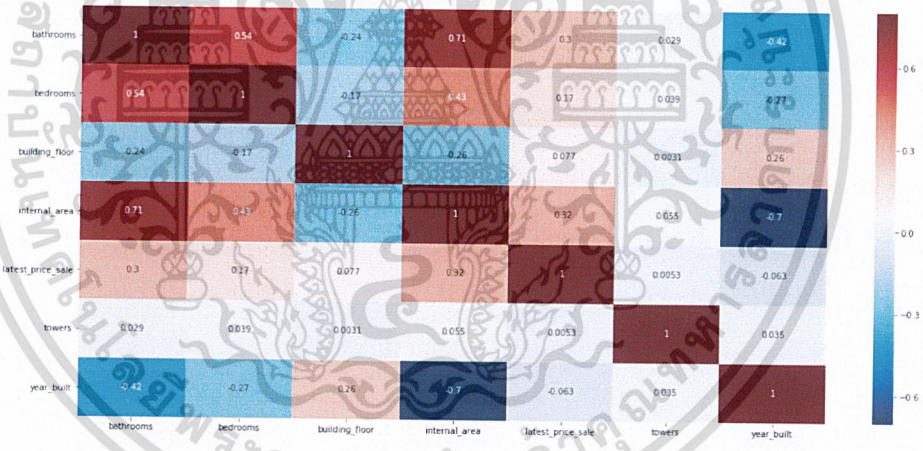
และในส่วนของ amenity ทั้งหมดหารลบคอลัมน์ที่เป็น NaN ทิ้งไปอาจทำให้ข้อมูลนั้นไม่พียงพอต่อการพยากรณ์ดั้งนั้นจึงเลือกวิธีการเติมค่าที่หายไปด้วยค่า False

ดูค่า correlation

```
latest_price_sale     1.000000
internal_area         0.322381
bathrooms             0.301477
bedrooms              0.166269
building_floor        0.077017
towers                0.005288
year_built           -0.063031
Name: latest_price_sale, dtype: float64
```

รูปที่ 3.66 คอลัมน์ที่มีค่า correlation ในช่วงราคา 10-20 ล้านบาท

พบว่าในช่วงราคา 10-20 ล้านบาท พื้นที่ภายในของคอนโดมิเนียมมีความสัมพันธ์กับราคาขาย มากที่สุด 0.32



รูปที่ 3.67 แสดงแผนภาพคอลัมน์ที่มีค่า correlation ในช่วงราคา 10-20 ล้านบาท

เมื่อตรวจสอบค่า collinearity ของตัวแปรทั้งหมดในช่วงราคา 10-20 ล้านบาทพบว่าไปมีคอลัมน์ใดที่สัมพันธ์กันเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตัดข้อมูลที่ผิดปกติ (outlier)

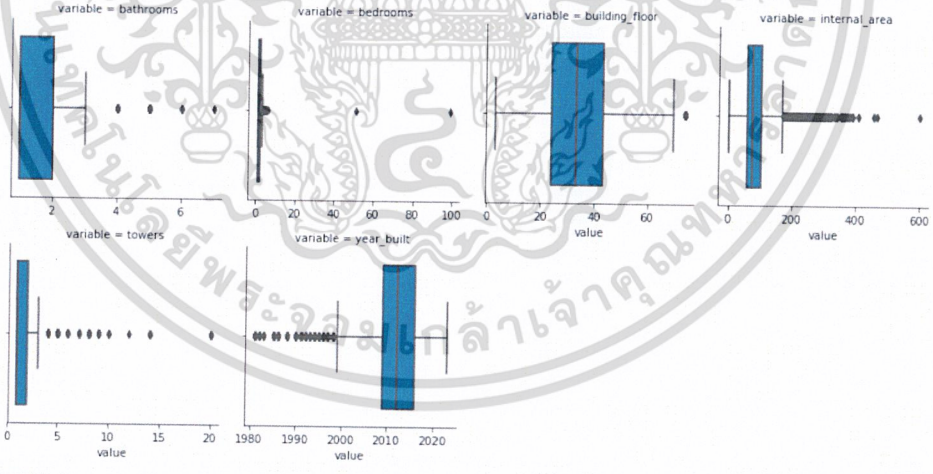
การตัด outlier ของข้อมูล จะทำให้โมเดลมีความแม่นยำมายิ่งขึ้น ตัดโดยวิธี interquartile range โดยดูจากความเบ้ของข้อมูล หากมีความเบ้มากกว่า 0.75 ทำการเทค log(1+X) เพื่อให้ข้อมูลมีการกระจายตัวแบบปกติก่อนนำไปตัด outliner นั่นเอง

|:

	Skewed Features
bedrooms	52.224455
towers	5.777721
internal_area	2.074569
bathrooms	0.758539
latest_price_sale	0.432907
building_floor	0.087644
year_built	-1.383745

รูปที่ 3.68 คอลัมน์ที่มีค่าความเบ้ในช่วงราคา 10-20 ล้านบาท

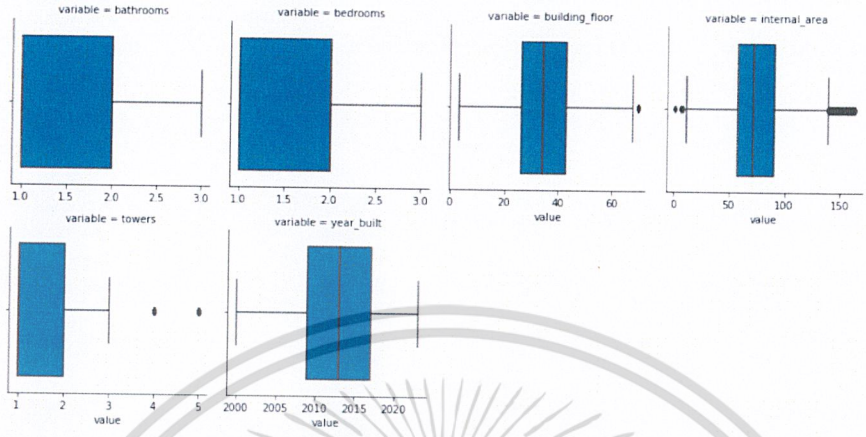
พล็อตกราฟ boxplot ก่อนตัด outlier ของข้อมูล bathrooms, bedrooms, building floor, internal area, towers



รูปที่ 3.69 ก่อนตัดค่าผิดปกติในช่วงราคา 10-20 ล้านบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พล็อตกราฟ boxplot หลังตัด outlier ของข้อมูล bathrooms, bedrooms, building floor, internal area, towers



รูปที่ 3.670 หลังตัดค่าผิดปกติในช่วงราคา 10-20 ล้านบาท

หลังจากนั้นนำตัวแปรออกจาก log(1+x)

การกำหนดชนิดของตัวแปร

- ตัวแปรห้องนอน ห้องน้ำ จำนวนอาคารทั้งหมดในโครงการ ตำแหน่งชั้นของคอนโดมิเนียม กำหนดชนิดของตัวแปรเป็น int
- ตัวแปรพื้นที่ภายใน ราขาขาย กำหนดชนิดของตัวแปรเป็น float
- ตัวแปรการอำนวนความสะดวกต่างๆ กำหนดชนิดของตัวแปรเป็น Boolean (True, False)

ในส่วนสุดท้ายก่อนนำข้อมูลเข้าโมเดลนั้นคือขั้นตอน Feature Engineering ในส่วนนี้จะมีการทำงาน 2 ส่วนคือ ข้อมูลที่เป็นหมวดหมู่เช่น เขต อำเภอ จังหวัด เราไม่สามารถนำข้อมูลดังกล่าวเข้าโมเดลได้จึงใช้เทคนิค one-hot encoding ในการแปลงคอลัมน์นั้นๆเป็นแถวต่างๆ และจะใส่ค่า 1 ในช่องที่มีในแถวนั้นและคอลัมน์อื่นๆจะใส่ค่า 0 นั่นเอง ใช้เทคนิค one-hot กับตัวแปร province, subdistrict, district ต่อมาคือการ Scaling data ของตัวแปรทั้งหมดที่เราจะนำไปพยากรณ์นั่นคือเป็นการแปลงตัวเลขให้มี scale เท่ากันที่อยู่ในช่วง [0,1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3 Model Training ในช่วงราคา 10-20 ล้านบาท

หลังจากการทำ Data preparation และ Feature Engineering มีข้อมูล 13,468 แถวและ 93คอลัมน์ โดยเราจะกำหนด feature ของเรานั่นคือ ตัวแปรที่เราจะนำไปพยากรณ์ และ target คือ ราคาขาย จากนั่นทำการแบ่งข้อมูลออกเป็น 3 ส่วนคือ 1.สำหรับ train โมเดล 70 เปอร์เซ็นต์ 2.สำหรับ test โมเดล 15 เปอร์เซ็นต์ 3.สำหรับ validation โมเดล 15 เปอร์เซ็นต์ โดยโมเดลที่ใช้ในการ ทำนายคือ neural network กำหนด architecture ดังตารางดังต่อไปนี้

ตารางที่ 3.9 ตารางแสดง architecture ของโมเดลในช่วงราคา 10-20 ล้านบาท

	neurons	activation
Input Layer	128	tanh
Hidden layer	128	tanh
	128	ReLU
	128	ReLU
	256	ReLU
	256	ReLU
	32	ReLU
	16	ReLU
	8	ReLU
Output Layer	1	ReLU

โดยเลือก optimizer = adam, loss = mean_square_error, epoch = 300, batch size = 256 ในแต่ละ dense จะมี kernel_regularizer=regularizers.l2(0.01) ในทุกๆ ชั้นหลังจากนั้นทำการ train หลังจากนั้นฝึกสอนโมเดล (train model) ดังรูป

```
In [219]:  1 prediction_3 = Prediction('predict_price_model_class2_edit.pkl', preprocess_df_class2)
           2 prediction_3.train_model3()
Epoch 295/300
9428/9428 [==============================] - 1s 59us/step - loss: 994434.5362 - val_loss: 1200465.5726
Epoch 296/300
9428/9428 [==============================] - 1s 61us/step - loss: 984784.5429 - val_loss: 1178520.5199
Epoch 297/300
9428/9428 [==============================] - 1s 62us/step - loss: 986608.6351 - val_loss: 1179389.9305
Epoch 298/300
9428/9428 [==============================] - 1s 60us/step - loss: 981080.3541 - val_loss: 1200971.2845
Epoch 299/300
9428/9428 [==============================] - 1s 56us/step - loss: 979482.0587 - val_loss: 1179527.2500
Epoch 300/300
9428/9428 [==============================] - 1s 59us/step - loss: 971915.6022 - val_loss: 1174359.0757
[[17271226.]
 [14704567.]
 [13638697.]
 ...
 [14724378.]
 [11418464.]
 [17845404.]]
Score (RMSE): 1581141.9326537312
r2_score : 0.6865463615581406
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.71 train โมเดลในช่วงราคา 10-20 ล้านบาท

3.7 การนำไปใช้งาน (Deployment)

เมื่อได้โมเดลที่เหมาะสมกับการพยากรณ์ราคาคอนโดมิเนียมแล้ว เพื่อให้ง่ายและสะดวกต่อการนำไปให้งานจริง จึงได้เขียนเป็นสคริปต์ไฟล์ โดยมีรูปแบบการเขียนเป็นโปรแกรมเชิงวัตถุ (Object oriented programming หรือ OOP) ในไพทอน แบ่งออเป็น 4 ไฟล์ดังนี้

- ไฟล์: Preprocessed.py
- ไฟล์: SelectionModel.py
- ไฟล์: PredictionModel.py
- ไฟล์: main.py

อธิบายรายละเอียดเพิ่มเติม

1. ไฟล์: Preprocessed.py คือ ขั้นตอนการเตรียมข้อมูลประกอบไปด้วย การพิจรณาข้อมูลที่ขาดหาย(Missing Value) การตัดข้อมูลที่มีค่าผิดปกติคือมีค่าน้อยกว่าหรือมากกว่าเกินความเป็นจริง (Outlier) การเปลี่ยนรูปข้อมูล (Data Transformation) เพื่อให้อยู่ในรูปของตัวเลขด้วยเทคนิค one-hot encoding รวมไปถึงการกำหนดชนิดของตัวแปร

2. ไฟล์: SelectionModel.py คือ โมเดล Random Forest เพื่อแยกช่วงราคาเป็น 3 ช่วงคือ 1-5 ล้านบาท, 5-10 ล้านบาท และ 10-20 ล้านบาท แบ่งออกเป็น 2 ฟังก์ชันคือ ฟังก์ชันสำหรับฝึกสอนโมเดล (train model) และฟังก์ชันเพื่อพยากรณ์ช่วงราคา

3. ไฟล์: PredictionModel.py คือ โมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมทั้ง 3 ช่วงราคา จะประกอบไปด้วย ฟังก์ชันสำหรับฝึกสอนโมเดล (train model) ทั้ง 3 ช่วงราคาและฟังก์ชันสำหรับพยากรณ์ราคา

4. ไฟล์: main.py คือ ไฟล์สำหรับเรียกใช้ทั้ง 3 ไฟล์ข้างต้น ผลลัพธ์ที่ได้เป็นราคาประเมินออกเป็นไฟล์ Excel

ซึ่งรายละเอียดการทั้งหมด จะเขียนเป็น README.md อธิบายฟังก์ชันอย่างละเอียดในภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัยและการอภิปรายผล

การทำโมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมในแต่ละช่วงราคามีกระบวนการดำเนินงานที่คล้ายกันดังนี้คือ การทำความเข้าใจเกี่ยวกับข้อมูล ทั้งนี้อาจจะพล็อตกราฟขั้นตอนนี้ทำให้เราเข้าใจในลักษณะของข้อมูลมากยิ่งขึ้น ต่อมาคือการเตรียมข้อมูล จากขั้นตอนข้างต้นเราจะเห็นลักษณะของข้อมูลว่าเป็นอย่างไร ข้อมูลบางตัวอาจมีขาดหายไป ดังนั้นเราอาจจะเติมค่าที่ขาดหายไปหรือการลบคอลัมน์นั้นทิ้งไป การตัดค่าที่แปลกเกินความเป็นจริงเพื่อให้โมเดลเรามีความถูกต้องมากยิ่งขึ้น และการแปลงชนิดของข้อมูล สุดท้ายคือการสร้างโมเดลและประเมินผลโมเดลด้วยค่าเฉลี่ยกำลังสอง และ ค่าสัมประสิทธิ์แสดงการตัดสินใจ

ประเมินผลโมเดลจากการพล็อตกราฟ

- Performance คือ การพล็อตค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Squared Error หรือ MSE) ของการฝึกสอน การทดสอบ และความใช้ได้ โดยการพล็อตกราฟที่ดีควรมีค่า MSE ขณะทดสอบ ขณะฝึกสอน ขณะ Validation ใกล้เคียงกัน ซึ่งจะแสดงถึงความเป็น Generalization หรือไม่เกิดการ Over-fitting
- Regression คือการพล็อตข้อมูลเปรียบเทียบระหว่างค่าจริง (Actural value) และค่าทำนายจากโครงข่าย (Forecasting value) ของข้อมูล Training, Validation, Testing
  1. Training ใช้ในการฝึกสอนเพื่อให้ความผิดพลาดของโครงข่ายมีค่าน้อยที่สุด
  2. Validation ใช้ในการวัดความเป็น Generalization
  3. Testing ใช้ในการวัดความสามารถหลังการฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.การการพยากรณ์ราคาประสินทรัพย์คอนโดมิเนียมในช่วงราคา 1-5 ล้านบาท

หลังจากการทำ Data preparation และ Feature Engineering มีข้อมูล 21,751 แถวและ 143 คอลัมน์ โดยเราจะกำหนด feature ของเรานั่นคือ ตัวแปรที่เราจะนำไปพยากรณ์ และ target คือราคาขาย จากนั่นทำการแบ่งข้อมูลออกเป็น 3 ส่วนคือ

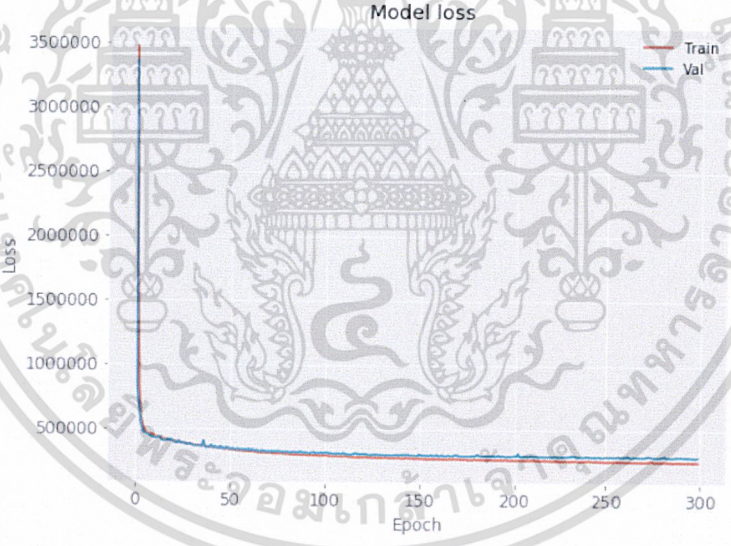
1. สำหรับ train โมเดล 70 เปอร์เซ็นต์

2. สำหรับ test โมเดล 15 เปอร์เซ็นต์

3. สำหรับ validation โมเดล 15 เปอร์เซ็นต์

โดยโมเดลที่ใช้ในการทำนาย คือ Artificial Neural Network

- Performance



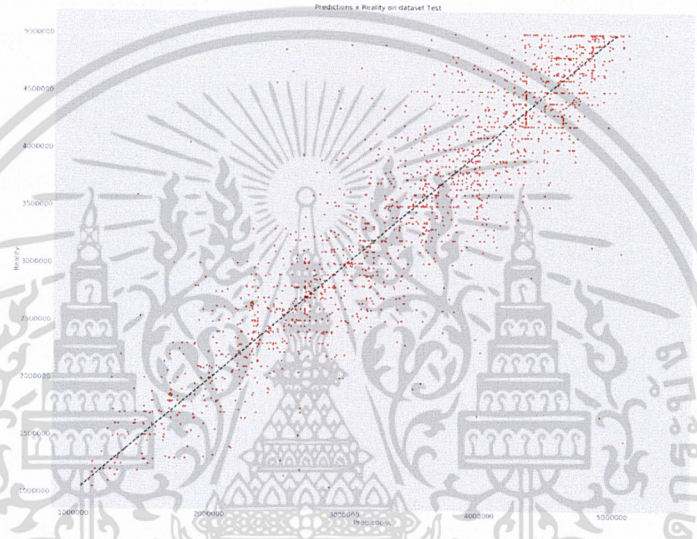
รูปที่ 4.1 แสดงค่า loss และ val loss ของโมเดลในช่วงราคา 1-5 ล้านบาท

- Loss — loss ที่เกิดจากการคำนวณจากข้อมูลที่ใช้สอนโมเดล
- Val Loss — loss ที่เกิดจากการคำนวณจากข้อมูลที่โมเดลไม่เคยเห็นมาก่อน ใช้เพื่อทดสอบโมเดลในกรณีที่เราเอาไปใช้จริง แล้วโมเดลต้องเจอกับข้อมูลที่ไม่เคยเห็นมาก่อน แล้วจะยังสามารถทำอย่างที่เราต้องการได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟแสดงให้เห็นว่า ค่า train และค่า val ทั้งสองเส้นแนบกันจนสนิท โมเดลจึงเรียนรู้ได้ดี กราฟมีค่า MSE ขณะฝึกสอน และขณะ Validation ใกล้เคียงกัน ซึ่งจะแสดงถึงความเป็น Generalization นั่นคือ โครงข่ายสามารถพยากรณ์ค่าช่วงนอกข้อมูลที่ฝึกสอนได้ แม้ไม่เคยเห็นมาก่อน

- Regression คือการพล็อตข้อมูลเปรียบเทียบระหว่างค่าจริง (Actual value) และค่าทำนายจากโครงข่าย (Forecasting value) ของข้อมูล



รูปที่ 4.2 ข้อมูลเปรียบเทียบระหว่างค่าจริงและค่าทำนายในช่วงราคา 1-5 ล้านบาท

เมื่อประเมินผลโมเดลด้วยค่าเฉลี่ยกำลังสองมีค่าเท่ากับ 428,248 ค่าสัมประสิทธิ์แสดงการตัดสินใจเท่ากับ 0.81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.การการพยาการณ์ราคาประสินทรัพย์คอนโดมิเนียมในช่วงราคา 5-10 ล้านบาท

หลังจากการทำ Data preparation และ Feature Engineering มีข้อมูล 16,228 แถวและ 93 คอลัมน์ โดยเราจะกำหนด feature ของเรานั่นคือ ตัวแปรที่เราจะนำไปพยากรณ์ และ target คือ ราคาขาย จากนั่นทำการแบ่งข้อมูลออกเป็น 3 ส่วนคือ

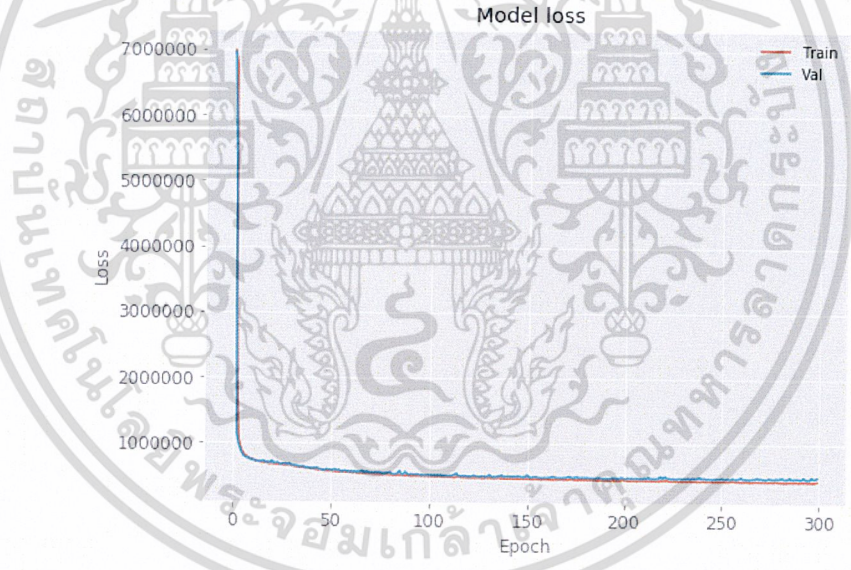
1.สำหรับ train โมเดล 70 เปอร์เซ็นต์

2.สำหรับ test โมเดล 15 เปอร์เซ็นต์

3.สำหรับ validation โมเดล 15 เปอร์เซ็นต์

โดยโมเดลที่ใช้ในการทำนาย คือ Artificial Neural Network

- Performance



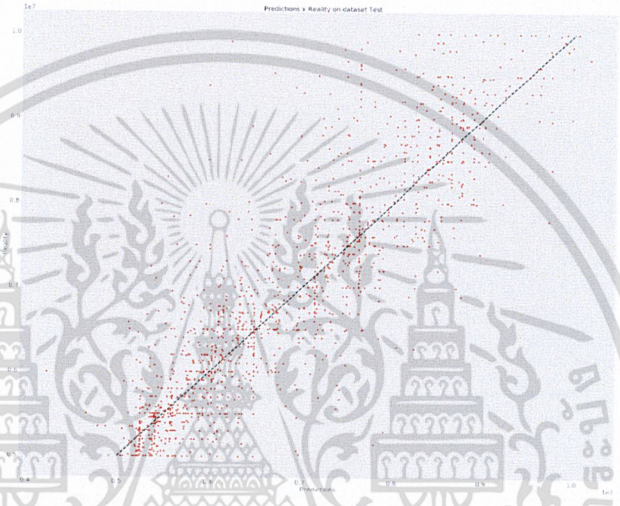
รูปที่ 4.3 แสดงค่า loss และ val loss ของโมเดลในช่วงราคา 5-10 ล้านบาท

- Loss — loss ที่เกิดจากการคำนวณจากข้อมูลที่ใช้สอนโมเดล
- Val Loss — loss ที่เกิดจากการคำนวณจากข้อมูลที่โมเดลไม่เคยเห็นมาก่อน ใช้เพื่อทดสอบโมเดลในกรณีที่เราเอาไปใช้จริง แล้วโมเดลต้องเจอกับข้อมูลที่ไม่เคยเห็นมาก่อนแล้วจะยังสามารถทำอย่างที่เราต้องการได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟแสดงให้เห็นว่า ค่า train และค่า val ทั้งสองเส้นแนบกันจนสนิท โมเดลจึงเรียนรู้ได้ดี กราฟมีค่า MSE ขณะฝึกสอน และขณะ Validation ใกล้เคียงกัน ซึ่งจะแสดงถึงความเป็น Generalization นั่นคือ โครงข่ายสามารถพยากรณ์ค่าช่วงนอกข้อมูลที่ฝึกสอนได้ แม้ไม่เคยเห็นมาก่อน

- Regression คือการพล็อตข้อมูลเปรียบเทียบระหว่างค่าจริง (Actual value) และค่าทำนายจากโครงข่าย (Forecasting value) ของข้อมูล



รูปที่ 4.4 ข้อมูลเปรียบเทียบระหว่างค่าจริงและค่าทำนายในช่วงราคา 5-10 ล้านบาท

เมื่อประเมินผลโมเดลด้วยค่าเฉลี่ยกำลังสองมีค่าเท่ากับ 669,998 ค่าสัมประสิทธิ์แสดงการตัดสินใจเท่ากับ 0.75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การการพยากาณ์ราคาประสินทรัพย์คอนโดมิเนียมในช่วงราคา 10-20 ล้านบาท

หลังจากการทำ Data preparation และ Feature Engineering มีข้อมูล 13,468 แถวและ 93 คอลัมน์ โดยเราจะกำหนด feature ของเรานั่นคือ ตัวแปรที่เราจะนำไปพยากรณ์ และ target คือ ราคาขาย จากนั่นทำการแบ่งข้อมูลออกเป็น 3 ส่วนคือ

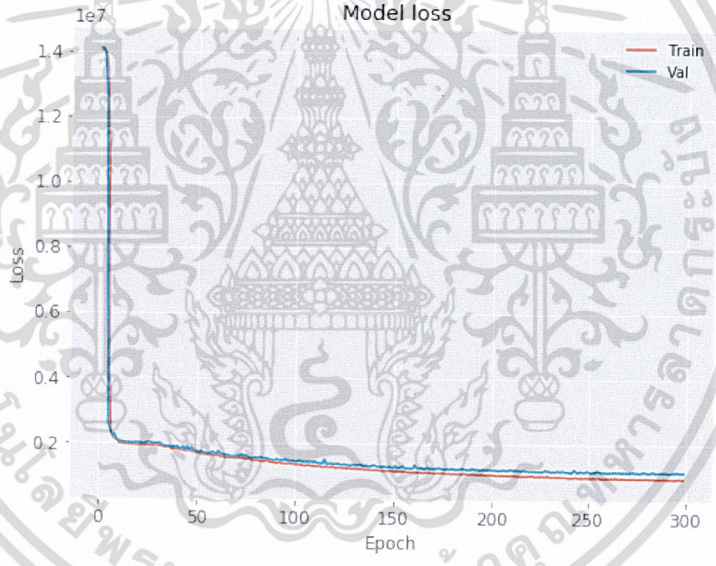
1. สำหรับ train โมเดล 70 เปอร์เซ็นต์

2. สำหรับ test โมเดล 15 เปอร์เซ็นต์

3. สำหรับ validation โมเดล 15 เปอร์เซ็นต์

โดยโมเดลที่ใช้ในการทำนาย คือ Artificial Neural Network

- Performance



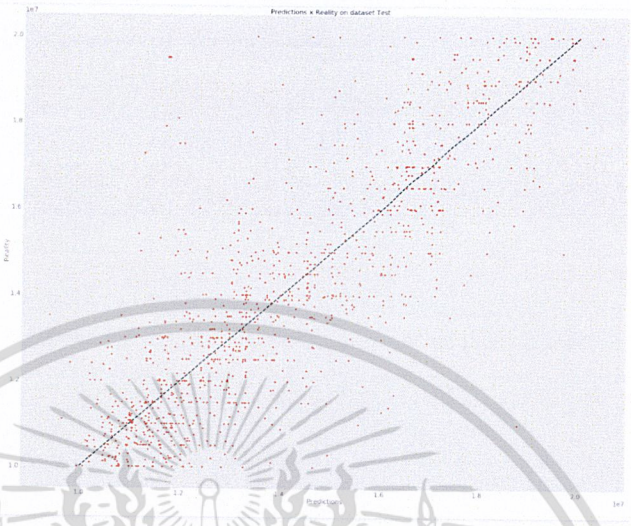
รูปที่ 4.5 แสดงค่า loss และ val loss ของโมเดลในช่วงราคา 10-20 ล้านบาท

- Loss — loss ที่เกิดจากการคำนวณจากข้อมูลที่ใช้สอนโมเดล
- Val Loss — loss ที่เกิดจากการคำนวณจากข้อมูลที่โมเดลไม่เคยเห็นมาก่อน ใช้เพื่อทดสอบโมเดลในกรณีที่เราเอาไปใช้จริง แล้วโมเดลต้องเจอกับข้อมูลที่ไม่เคยเห็นมาก่อน แล้วจะยังสามารถทำอย่างที่เราต้องการได้หรือไม่

จากกราฟแสดงให้เห็นว่า ค่า train และค่า val ทั้งสองเส้นแนบกันจนสนิท โมเดลจึงเรียนรู้ได้ดี กราฟมีค่า MSE ขณะฝึกสอน และขณะ Validation ใกล้เคียงกัน ซึ่งจะแสดงถึงความเป็น Generalization นั่นคือ โครงข่ายสามารถพยากรณ์ค่าช่วงนอกข้อมูลที่ฝึกสอนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Regression คือการพล็อตข้อมูลเปรียบเทียบระหว่างค่าจริง (Actual value) และค่าทำนายจากโครงข่าย (Forecasting value) ของข้อมูล



รูปที่ 4.6 ข้อมูลเปรียบเทียบระหว่างค่าจริงและค่าทำนายในช่วงราคา 5-10 ล้านบาท

เมื่อประเมินผลโมเดลด้วยค่าเฉลี่ยกำลังสองมีค่าเท่ากับ 1,497,060 ค่าสัมประสิทธิ์แสดงการตัดสินใจเท่ากับ 0.7

4. โมเดลเพื่อแยกช่วงราคาเป็น 3 ช่วงคือ 1-5 ล้านบาท, 5-10 ล้านบาท และ 10-20 ล้านบาท

หลังจากการทำ Data preparation จะเพิ่มขั้นตอนแบ่งกลุ่มช่วงราคาออกเป็น 3 กลุ่มคือ ช่วง 1-5 ล้านบาท, 5-10 ล้านบาท และ 10-20 ล้านบาท จากนั้นทำการ group จะได้ทั้งหมด 3 class และทำการลบคอมลัมน์ของราคาขายทิ้งไป ต่อจากนั้นทำการ Feature Engineering ต่อไป มีข้อมูลทั้งหมด 46,055 แถวและ 147 คอลัมน์ ก่อนนำไปเข้าโมเดลแบ่งข้อมูลออกเป็น 2 ส่วนคือ 1. ข้อมูลสำหรับฝึกสอนโมเดล 70 เปอร์เซ็นต์ 2.ข้อมูลสำหรับทดสอบโมเดล 30 เปอร์เซ็นต์ เลือกโมเดลที่ใช้คือ Random Forest พบว่ามีความถูกต้องของโมเดล คือ 89.51 เปอร์เซ็นต์

Accuracy: 89.51%

	precision	recall	f1-score	support
0	0.94	0.95	0.94	7663
1	0.82	0.81	0.81	3937
2	0.89	0.87	0.88	2629
micro avg	0.90	0.90	0.90	14229
macro avg	0.88	0.88	0.88	14229
weighted avg	0.89	0.90	0.89	14229

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 แสดงค่าความถูกต้องของโมเดลการพยากรณ์เพื่อแบ่งช่วงราคา

บทที่ 5
สรุปผลวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การการพยาการณ์ราคาประสินทรัพย์คอนโดมิเนียมในช่วงราคา 1-5 ล้านบาท ประเมินผลโมเดลด้วยค่าเฉลี่ยกำลังสองมีค่าเท่ากับ 428,248 ค่าสัมประสิทธิ์แสดงการตัดสินใจเท่ากับ 0.81 การการพยาการณ์ราคาประสินทรัพย์คอนโดมิเนียมในช่วงราคา 10-20 ล้านบาท ประเมินผลโมเดลด้วยค่าเฉลี่ยกำลังสองมีค่าเท่ากับ 669,998 ค่าสัมประสิทธิ์แสดงการตัดสินใจเท่ากับ 0.75 การพยาการณ์ราคาประสินทรัพย์คอนโดมิเนียมในช่วงราคา 10-20 ล้านประเมินผลโมเดลด้วยค่าเฉลี่ยกำลังสองมีค่าเท่ากับ 1,497,060 ค่าสัมประสิทธิ์แสดงการตัดสินใจเท่ากับ 0.7 การพยากรณ์ราคาประสินทรัพย์คอนโดมิเนียมทั้ง 3 ช่วงราคาด้วยโมเดล neural network และโมเดลการพยากรณ์ราคาประเมินสินทรัพย์คอนโดมิเนียมออกเป็น 3 ช่วง มีค่าความถูกต้องเท่ากับ 89.51 เปอร์เซ็นต์ ด้วยโมเดล Random Forest

5.2 ข้อเสนอแนะ

1. การการพยาการณ์ราคาประสินทรัพย์คอนโดมิเนียมอาจแบ่งช่วงให้มีช่วงราคาที่สั้นลง อาจทำให้การประเมินผลด้วยค่าเฉลี่ยกำลังสองลดลง

2. โมเดล neural network อาจปรับเปลี่ยนฟังก์ชันกระตุ้นและจำนวนนิวรอนในชั้นเพื่อให้ได้ค่าความผิดพลาด เวลาที่น้อยในการฝึกสอน บางโครงข่ายที่ซับซ้อนมากๆ อาจจะหยุดการฝึกสอนเนื่องจากครบจำนวนรอบ (Epoch) และคำตอบที่ได้เป็นเพียง (Best found) เพื่อเพิ่มสมรรถนะของโครงข่ายประสาทเทียม

3. ปรับเงื่อนไขในการหยุดสอน (Stopping criteria) เพื่อหาคำตอบที่ดีกว่า ใช้ในกรณีที่เงื่อนไขการฝึกสอนเข้มงวดเกินไป (Tighten criteria) เพื่อเพิ่มสมรรถนะของโครงข่ายประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

ปัญญา หันตุลาม. 2556 . โครงสร้างเครือข่ายประสาทเทียม (Neural Network). [Online].

Available: http://panya-electronicelab.blogspot.com/2013/09/neural-network.html

อำภา สาระศิริ. 2559 . เทคนิคการเรียนรู้พื้นฐานโครงข่ายประสาทเทียม. [Online].

Available: http://www.mut.ac.th/research-detail-92

Engkarat Techapanurak. 2018 . Artificial Neural Network โครงข่ายประสาทเทียม. [Online].

Available: https://medium.com/@dopplerz/bayesian-neural-network

Mr. P L. Optimization & Activation Function. [Online].

Available: https://medium.com/@pingloaf/deep-learning

ธนาวุฒิ ประกอบผล. 2552 . โครงข่ายประสาทเทียม Artificial Neural Networks. [Online].

Available: file:///C:/Users/user/Downloads/Documents/5.%20โครงข่าย%2073-87.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ไฟล์: README.md

FORECASTING THE PRICE OF CONDOMINIUM ASSETS

Condominium real estate assets is that gained popularity in the present continuous. Whether it is for investment or residential. This is trading a condominium appraised requires to cover price in every co-operative education has prepared a model assets condominium price forecast. By made by relying on basic information and factors that affect estimated price of condominiums by using Random Forest models split into 3 price range the range is 1-5 million baht, 5-10. Million and 10-20 million and using artificial neural network models calculate the estimated price from the database of the group such

a price range, the price of an asset of a condominium.

Requirement

This project requires **Python 3** and the following Python libraries installed

Install : Install using pip

- [NumPy](http://www.numpy.org/)

- [Pandas](http://pandas.pydata.org/)

- [matplotlib](http://matplotlib.org/)

- [scikit-learn](http://scikit-learn.org/stable/)

- [Tensorflow](https://www.tensorflow.org/)

Dataset

file name : post_data_for_Aun.json

Usage

Folder structure

- python_script : all python files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Preprocessed.py : Data preparation before modeling

- SelectionModel.py : Forecasting model for the selection and Feature to divide into 3 condominium appraised asset range using Random Forest models.

- PredictionModel.py : The artificial neural network forecasting model in forecasting asset appraised condominium

- main.py : file run for train and predict model

1. Preprocessed.py

Function

- filter_columns : Select Feature for the model.

- Feature : bathrooms, bedrooms, building_floor, district, internal_area, latest_price_sale, province, subdistrict, towers, year_built, amenity

- take_log : Logarithimic distribution of the target variable : latest_price_sale

- drop_null : Searching for predictors which have at least 30% of its values missing, deleted rows (percentage of nulls less than 5%), replace median (percentage of nulls less than 5%)

- fill_na : replace nan values with 0

- Outlier_Discrete : Delete the abnormal value of feature in the bedroom and bathroom.

- log_skewed : check which features also have some skewness that can affect the results and log those values. Taking the logarithm from our target variable, we obtain something similar to a Gaussian distribution

- drop_outliers : Delete data with outliers. To make the model more precise cutting method, interquartile range range

- natural_log

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- fill_amenity : Replace the missing value of amenity value false

- onehot_encoding : For this categorical feature, converted to a number format using one-hot encoding

    - Feature : district, subdistrict, province

- label_class : For the preparation of information to gain access to selected model.

- drop_columns : Delete columns district, subdistrict, province

Usage Library For Model

```python
import numpy as np
import pandas as pd
from scipy.stats import skew
pd.options.mode.chained_assignment = None
```

Declare Preprocessed

```python
def preprocess(df,mode,schema=None,onehot_dict=None):
```

- **df** *(data frame)

    -- load the dataset into a Pandas data frame

- **mode**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **train** : Data Preparation for the artificial neural network forecasting model in forecasting asset appraised condominium

- **predict_class** : Data Preparation forecasting model for the selection and Feature to divide into 3 condominium appraised asset range using Random Forest models.

- **predict_price** : For data predict price

- **schema** (list) default: None

-- list of all column for prediction

- **onehot_dict** *(dict) default: None*

-- dict of onehot encoding (district, subdistrict, province)

```python
def preprocess_prediction(df,price_column,group,selection_column,onehot_dict):
```

train model artificial neural network : Data Preparation for the artificial neural network forecasting model in forecasting asset appraised condominium

- **df** (data frame)

-- load the dataset into a Pandas data frame

- **price_column** (str)

-- target or variable to prediction : latest_price_sale

- **group**

- **group==1** : price range 1-5 m

- **group==2** : price range 5-10 m

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **group==3** : price range 10-20 m

- **selection_column** (list) default: None

    -- list of all column for prediction

- **onehot_dict** *(dict) default: None*

    -- dict of onehot encoding (district, subdistrict, province)

2. SelectionModel.py

Functions

1. preprocess train data

2. preprocess test data

3. train model

    - train model with *```RandomForestClassifier```*, find the class

    - load trained model

4. predict class

Usage Library For Model

```python

from Preprocessed import preprocess

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification_report,accuracy_score

from sklearn.ensemble import RandomForestClassifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
import numpy as np

import pandas as pd

import pickle

import json
```

```

#### Declare SelectionModel

```python

def __init__(self, model_path, df=None, load_model=False):

```

- **model_path** *(str)*

        -- directory path to save model

- **df** *(data frame)*   default: None

        -- dataset into a Pandas data frame after data preprocessing (mode == predict_class)

- **load_model** *(boolean) default: False*

 - load trained model or not

        - if `True` : load trained model from `model_path` and do not train new model

preprocess train data

preprocessing for train Selection Model before prediction class

```
` self._train_df = preprocess(self._df,'predict_class')`

```python
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

def preprocess_train(self):

```

### preprocess test  data

preprocessing for test Selection Model before prediction class

```python

def preprocess_test(self, test_data) :

```

- **test_data** *(data frame)*

        -- data to prediction class and predict price each a artificial neural network model

        `self._preprocess_df_test = preprocess(test_data,'predict_price',selection_column,onehot_dict)`

train model

Will be the model for prediction class

```python

def train(self):

```

predict class

```python

def predict(self, predict_data):

```

- **predict_data** *(data frame)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

-- data to prediction by RandomForest model

3. PredictionModel.py

Functions

1. model for prediction

2. train model

- train model with *```artificial neural network```*, find the predict price condominium

- load trained model

3. predict price condominium

Usage Library For Model

```python

import json

import pandas as pd

import pickle

import numpy as np

from math import sqrt

from keras import regularizers

from sklearn.preprocessing import MinMaxScaler,StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error,r2_score

from keras.models import Sequential

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

from keras.layers.core import Dense, Activation

from keras.callbacks import ModelCheckpoint

from Preprocessed import preprocess_prediction

from SelectionModel import SelectionModel

from sklearn import preprocessing

```

#### Declare PredictionModel

```python

def __init__(self, df=None, load_model=False):

```

- **df** *(data frame)* default: None

        -- dataset into a Pandas data frame after data preprocessing (mode == train)

- **load_model** *(boolean) default: False*

 - load trained model or not

        - if `True` : load trained model from `model_path` and do not train new model

model for prediction

Modeling to predict the price each price range would have a different model.

```python

def generate_nn_sequential_model(model,group,input_dim)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
```

- **group**

        - **group==1** : price range 1-5 m

        - **group==2** : price range 5-10 m

        - **group==3** : price range 10-20 m

- **input_dim**(int) : shape of feature

train model

- train model 1 : train model price range 1-5 m

- train model 2 : train model price range 5-10 m

- train model 3 : train model price range 10-20 m

```python

def train_model(self,model_path,df,group):

 ```

  model_path *(str)*

        -- directory path to save model

- **df** *(data frame)* default: None

        -- dataset into a Pandas data frame after data preprocess

- **group**   (int) : range price for train model

        - **group==1** : price range 1-5 m

        - **group==2** : price range 5-10 m

        - **group==3** : price range 10-20 m

predict price condominium
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```python
def predict(self, predict_data):
```

- **predict_data** *(data frame)*

    -- data to prediction by artificial neural network model

4. main.py

Functions

1. train model selection
2. train model prediction
3. prediction price

Usage Library For Model

```python
from sklearn.model_selection import train_test_split
from SelectionModel import SelectionModel
from PredictionModel import PredictionModel
import json
import pandas as pd
```

Load data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```python
with open('../../dataset/post_data_for_Aun.json', 'rb') as json_data:
    data = json.load(json_data)
df = pd.DataFrame(data)
train_df, predict_df = train_test_split(df, test_size=0.3)
```

train_df : for train selection model

predict_df : for test selection model and predict price by artificial neural network model

Train and Export Selection Model

```python
model_selection_train_object = SelectionModel('../model/predict_class_model.pkl', df=train_df)
model_selection_train_object.train()
```

output :

Trained model will be saved name is `predict_class_model.pkl`file. directory as **`../model`** data input (data frame) is `train_df`

Will be train model : accuracy is 89.18 %

Example : Train Model

```python
prediction_1 = PredictionModel(df)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **df** *(data frame)* : input data for train

1. Train Model 1 : train model price range 1-5 m

```python
prediction_1.train_model('../model/predict_price_model_class0.pkl',df,group=0)
```

output :

Trained model will be saved name is `predict_price_model_class0.pkl`file. directory as **`../model`**

Data input (data frame) is df , group=0 is range price 1-5 m

Will be train model : r_2 = 0.81

2. Train Model 2 : train model price range 5-10 m

```python
prediction_1.train_model('../model/predict_price_model_class1.pkl',df,group=1)
```

output :

Trained model will be saved name is `predict_price_model_class1.pkl`file. directory as **`../model`**

Data input (data frame) is df , group=1 is range price 5-10 m

Will be train model : r_2 = 0.71

3. Train Model 3 : train model price range 10-20 m

```python

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

prediction_1 = PredictionModel(df)

prediction_1.train_model('../model/predict_price_model_class2.pkl',df,group=2)

```

Trained model will be saved name is `predict_price_model_class2.pkl`file. directory as **`../model`**

Data input (data frame) is df , group=2 is range price 10-20 m

Will be train model : r_2 = 0.65

#### Predict and Submit

```python

model_selection_predict_object = PredictionModel(load_model=True)

predicted_result = model_selection_predict_object.predict(predict_df)

predicted_result.to_csv('../result/predicted_result_all_prediction.csv')

```

output :

Trained model will be loaded.

The predited price will be given. The output price depend on what model was trained.

Result of price prediction submission to csv : saved name is `predicted_result_all_prediction.csv` file. directory as **`../result`**

but feature of prediction value is nan, can't prediction so submission to csv : saved name is `not_prediction_have_nan.csv` file. directory as **`../result`**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

1. ไฟล์: Preprocessed.py

```
import numpy as np
import pandas as pd
from scipy.stats import skew
pd.options.mode.chained_assignment = None


def filter_columns(df):
    amenity = []
    for x in list(df.columns):
        if 'amenity' in x:
            amenity.append(x)
    FEATURES = ['bathrooms', 'bedrooms', 'building_floor', 'district', 'internal_area',
                'latest_price_sale', 'province', 'subdistrict', 'towers', 'year_built']
    for value in amenity:
        FEATURES.append(value)
    df = df[FEATURES]
    return df
def take_log(df,column):
    df[column] = np.log1p(df[column])
    return df
def drop_null(df):
    null_values = df.isnull().sum()/df.shape[0]*100
    missing_value = null_values[null_values >= 30]
    missing_value.sort_values(inplace=True)
    # deleted columns missing data morn than 30%
    deleted_col = []
    for key, value in missing_value.items():
        if bool(value) is True:
            deleted_col.append(key)
            print('deleted_col : ', len(deleted_col))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for col in deleted_col:
        del df[col]
    # deleted rows (percentage of nulls less than 5%)
    numeric_data = df[
        ['bathrooms', 'bedrooms', 'building_floor', 'internal_area', 'latest_price_sale',
'towers', 'year_built']]
    null_values = numeric_data.isnull().sum() / numeric_data.shape[0] * 100
    missing_value = null_values[null_values <= 5]
    for key, value in missing_value.items():
        if  0 <= value <= 5:
            df.dropna(subset=[key], inplace=True)
    # repace median (percentage of nulls less than 5%)
    numeric_data = df[
        ['bathrooms', 'bedrooms', 'building_floor', 'internal_area', 'latest_price_sale',
'towers', 'year_built']]
    null_values = numeric_data.isnull().sum() / numeric_data.shape[0] * 100
    med_value = null_values[null_values >= 5]
    for key, value in med_value.items():
        if value >= 5:
            med = df[key].median()
            df[key] = df[key].fillna(med)
    return df
def fill_na(df):
    df = df.fillna(0)
    return df


def  discrete_outlier(df,*columns):
    # df.drop(df[df['bathrooms'] > 5 ].index, inplace=True)
    # df.drop(df[df['bedrooms'] > 5].index, inplace=True)
    # return df
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for column in columns:
        df = df.drop(df[df[column] > 5].index)
    return df
def log_skewed(df):
    numeric_feats = df.dtypes[df.dtypes != "object"].index
    skewed_feats = df[numeric_feats].apply(lambda x: pd.DataFrame.skew(x))  #
compute skewness
    skewed_feats = skewed_feats[skewed_feats > 0.1]
    skewed_feats = skewed_feats.index
    df = take_log(df,skewed_feats)
    return df, skewed_feats
def drop_outliers(df):
    data_outliner = df[['internal_area','building_floor','towers','year_built']]
    for field_name in data_outliner :
        distance = 1.5 * (np.percentile(df[field_name], 75) -
np.percentile(df[field_name], 25))
        df = df.drop(df[df[field_name] > distance + np.percentile(df[field_name],
75)].index)
        df = df.drop(df[df[field_name] < np.percentile(df[field_name], 25) -
distance].index)
    return df
def natural_log(df, skewed_feats,columns):
    df[columns] = np.expm1(df[columns])
    df[skewed_feats] = np.expm1(df[skewed_feats])
    return df
def fill_amenity(df):
    amenity_columns = []
    for dataset in list(df.columns):
        if 'amenity' in dataset :
            amenity_columns.append(dataset)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for amenity_column in amenity_columns :
        df[amenity_column].fillna(False,inplace=True)
    return df
def onehot_encoding (df,columns):
    for x in df[columns]:
        df[x] = pd.Categorical(df[x])
        dfDummies = pd.get_dummies(df[x], prefix = x)
        df = pd.concat([df, dfDummies], axis=1)
    return df
def label_class(df):
    labels = ["low_price", "medium_price", "hight_price"]
    df['group'] = pd.cut(x=df.latest_price_sale ,
bins=[1_000_000,5_000_000,10_000_000,20_000_000], labels=labels)
    df.dropna(subset=['group'],inplace=True)
    del(df['latest_price_sale'])
    df.group = df.group.astype('category')
    df.group = df.group.cat.codes
    return df
def drop_columns(df,*columns):
    for column in columns:
        del(df[column])
def preprocess(df, mode, schema=None, onehot_dict=None):
    if mode == 'train':
        if schema is not None and onehot_dict is not None:
            for key, value in onehot_dict.items():
                df = df[df[key].isin(value)]
            schema.remove('group')
            schema.append('latest_price_sale')
        df = filter_columns(df)
        df = take_log(df,'latest_price_sale')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        df = drop_null(df)
        df = discrete_outlier(df,'bathrooms','bedrooms')
        df, skewed_feats = log_skewed(df)
        df = drop_outliers(df)
        df = natural_log(df, skewed_feats,'latest_price_sale')
        df = fill_amenity(df)
        df = onehot_encoding(df,['district','subdistrict','province'])
        if schema is not None:
            for column in schema:
                if column not in list(df.columns):
                    df[column] = 0
            df = df[schema]
#           ordered_data = []
#           for index, row in df.iterrows():
#               new_row = []
#               for column in schema:
#                   new_row.append(row[column])
#               ordered_data.append(new_row)
#           df = pd.DataFrame(ordered_data, columns=schema)
    elif mode == 'predict_class':
        if onehot_dict is not None:
            for key, value in onehot_dict.items():
                df = df[df[key].isin(value)]
        df = filter_columns(df)
        df = take_log(df,'latest_price_sale')
        df = drop_null(df)
        # df = fill_na(df)
        df = discrete_outlier(df,'bathrooms','bedrooms')
        df, skewed_feats = log_skewed(df)
        df = drop_outliers(df)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        df = natural_log(df, skewed_feats,'latest_price_sale')
        df = fill_amenity(df)
        df = onehot_encoding(df, ['district', 'subdistrict', 'province'])
        if schema is not None:
            for column in schema:
                if column not in list(df.columns):
                    df[column] = 0
        df = label_class(df)
        drop_columns(df,'province','district','subdistrict')
        if schema is not None:
            for column in schema:
                if column not in list(df.columns):
                    df[column] = 0
            df = df[schema]
#           ordered_data = []
#           for index, row in df.iterrows():
#               new_row = []
#               for column in template_schema:
#                   new_row.append(row[column])
#               ordered_data.append(new_row)
#           df = pd.DataFrame(ordered_data, columns=template_schema)
    elif mode == 'predict_price':
        if onehot_dict is not None:
            for key, value in onehot_dict.items():
                df = df[df[key].isin(value)]
        df = filter_columns(df)
        df = fill_amenity(df)
        df = onehot_encoding(df, ['district', 'subdistrict', 'province'])
        if schema is not None:
            for column in schema:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if column not in list(df.columns):
                df[column] = 0
        df = label_class(df)
        drop_columns(df,'province','district','subdistrict')
        if schema is not None:
            for column in schema:
                if column not in list(df.columns):
                    df[column] = 0
            df = df[schema]

    return df
def preprocess_prediction(df,price_column,group,selection_column,onehot_dict):
    price_range = None
    if group == 0:
        price_range = (1_000_000,5_000_000)
    elif group == 1:
        price_range = (5_000_000, 10_000_000)
    elif group == 2:
        price_range = (10_000_000, 20_000_000)
    df = df[(df[price_column] > price_range[0]) & (df[price_column] < price_range[1])]
    df = preprocess(df, 'train', selection_column, onehot_dict)
    return df
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ไฟล์: SelectionModel.py

```
from Preprocessed import preprocess
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report,accuracy_score
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import pandas as pd
import pickle
import json
class SelectionModel:
        def __init__(self, model_path, df=None, load_model=False):
        self._model_path = model_path
        self._df = df
        if load_model:
            self._model = pickle.load(open(model_path, 'rb'))
        self._train_df = None
        self._test_df = None
        self._preprocess_df_test = None
        self._filter_df = None
        self._preprocess_df_test_nans = None
        self._preprocess_df_test_not_nans =None
    def preprocess_train(self):
        ### Data preprocessing ###
        subdistrict_train_list = self._df['subdistrict']
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        district_train_list = self._df['district']
        province_train_list = self._df['province']
        subdistrict_train_list = list(set(subdistrict_train_list))
        district_train_list = list(set(district_train_list))
        province_train_list = list(set(province_train_list))
        onehot_dict =
{'subdistrict':subdistrict_train_list,'district':district_train_list,'province':province_train_list}
        with open('../model/onehot_dict.json', 'w') as f:
            json.dump(onehot_dict, f, indent=4)
        self._train_df = preprocess(self._df,'predict_class')

    def preprocess_test(self, test_data):
        ### preprocessing and predict model selection ###
        with open('../model/selection_column.json', 'r') as f:
            selection_column = json.load(f)
        with open('../model/onehot_dict.json', 'r') as f:
            onehot_dict = json.load(f)
        self._preprocess_df_test =
preprocess(test_data,'predict_price',selection_column,onehot_dict)
        self._preprocess_df_test_nans =
self._preprocess_df_test.loc[self._preprocess_df_test.isnull().any(axis=1)]
        self._preprocess_df_test_not_nans = self._preprocess_df_test
        self._preprocess_df_test_not_nans.dropna(inplace=True)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Y_test = self._preprocess_df_test_not_nans.group

        X_test = self._preprocess_df_test_not_nans.drop(columns = ['group'])

        scalar = StandardScaler()

        self._test_df = scalar.fit_transform(X_test)


    def save_model(self,model):

        fout = open(self._model_path, 'wb')

        pickle.dump(model, fout)

        fout.close()


    def train(self):

        self.preprocess_train()

        self._train_df.dropna(inplace=True)

        X = self._train_df.drop(columns=['group'])

        Y = self._train_df['group']

        X_train, X_validation, Y_train, Y_validation = train_test_split(X, Y, test_size=0.3,
stratify=Y)

        scalar = StandardScaler()

        X_train = scalar.fit_transform(X_train)

        X_validation = scalar.transform(X_validation)

        rForest = RandomForestClassifier()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        rForest.fit(X_train, Y_train)
        prediction = rForest.predict(X_validation)
        self.save_model(rForest)
        # evaluate predictions
        accuracy = accuracy_score(Y_validation, prediction)
        print("Accuracy: %.2f%%" % (accuracy * 100.0))
        print(classification_report(Y_validation, prediction))
        with open('../model/selection_column.json', 'w') as f:
            json.dump(list(self._train_df.columns), f, indent=4)


    def predict(self, predict_data):
        self.preprocess_test(predict_data)
        prediction = self._model.predict(self._test_df)
#         print("predict class : ", prediction)
        self._preprocess_df_test['group'] = prediction
        self._preprocess_df_test_nans.to_csv('../result/not_prediction_have_nan.csv')
        print(prediction)
        return self._preprocess_df_test
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ไฟล์: PredictionModel.py

```
import json
import pandas as pd
import pickle
import numpy as np
from math import sqrt
from keras import regularizers
from sklearn.preprocessing import MinMaxScaler,StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error,r2_score
from keras.models import Sequential
from keras.layers.core import Dense, Activation
from keras.callbacks import ModelCheckpoint
from Preprocessed import preprocess_prediction
from SelectionModel import SelectionModel
from sklearn import preprocessing


class PredictionModel:


    def __init__(self, df=None, load_model=False):
        self._model = None
        self._df = df
        if load_model:
            self._model1 = pickle.load(open('../model/predict_price_model_class0.pkl',
'rb'))
            self._model2 = pickle.load(open('../model/predict_price_model_class1.pkl',
'rb'))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            self._model3 = pickle.load(open('../model/predict_price_model_class2.pkl',
'rb'))

        self._preprocess_df_class0 = None

        self._preprocess_df_class1 = None

        self._preprocess_df_class2 = None

        self._selection_column = None

        self._onehot_dict = None


    def load_column(self):

        with open('../model/selection_column.json', 'r') as f:

            self._selection_column = json.load(f)

        with open('../model/onehot_dict.json', 'r') as f:

            self._onehot_dict = json.load(f)


    def predict(self, predict_data):

            model_selection_predict_object =
SelectionModel('../model/predict_class_model.pkl', load_model=True)

            predict_df = model_selection_predict_object.predict(predict_data)

    #         predict_df = model_selection_predict_object.predict(predict_data,
preprocess = True)

            g1 = predict_df[predict_df['group'] == 0]

            g2 = predict_df[predict_df['group'] == 1]

            g3 = predict_df[predict_df['group'] == 2]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            del(g1['group'])
            del(g2['group'])
            del(g3['group'])
            min_max_scaler = MinMaxScaler()
            g1_data = min_max_scaler.fit_transform(g1)
            g1_pred = self._model1.predict(g1_data)
            g2_data = min_max_scaler.fit_transform(g2)
            g2_pred = self._model2.predict(g2_data)
            g3_data = min_max_scaler.fit_transform(g3)
            g3_pred = self._model3.predict(g3_data)
            g1['predicted_price'] = g1_pred
            g2['predicted_price'] = g2_pred
            g3['predicted_price'] = g3_pred
            result = pd.concat([g1,g2,g3])
        return result
    @staticmethod
    def generate_nn_sequential_model(model,group,input_dim):
        if group == 0:
            model.add(Dense(128, input_dim=input_dim, activation='tanh',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))
            model.add(Dense(128, activation='tanh',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            model.add(Dense(128, activation='tanh',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))

            model.add(Dense(128, activation='tanh',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))

            model.add(Dense(256, activation='relu',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))

            model.add(Dense(256, activation='relu',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))

            model.add(Dense(64, activation='relu',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))

            model.add(Dense(1, activation='relu',
kernel_regularizer=regularizers.l2(0.01),use_bias=True))
        elif group == 1:

            model.add(Dense(512, input_dim=input_dim, activation='tanh',
kernel_regularizer=regularizers.l2(0.01)))

            model.add(Dense(256, activation='tanh',
kernel_regularizer=regularizers.l2(0.01)))

            model.add(Dense(128, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))

            model.add(Dense(128, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))

            model.add(Dense(64, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))

            model.add(Dense(64, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            model.add(Dense(32, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(32, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(16, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(8, activation='relu', kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(4, activation='relu', kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(1, kernel_initializer='normal', activation='linear'))
        elif group == 2:
            model.add(Dense(128, input_dim=input_dim, activation='tanh',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(128, activation='tanh',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(128, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(128, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(256, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(256, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(256, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            model.add(Dense(32, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(16, activation='relu',
kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(8, activation='relu', kernel_regularizer=regularizers.l2(0.01)))
            model.add(Dense(1, kernel_regularizer=regularizers.l2(0.01),
activation='linear'))
        model.compile(loss='mean_absolute_error', optimizer='adam')
        return model


    def train_model(self,model_path,df,group):
        self.load_column()
        self._df = preprocess_prediction(df=df,price_column='latest_price_sale',
                                group=group,selection_column=self._selection_column
                                ,onehot_dict = self._onehot_dict)
        self._df.dropna(inplace=True)
        feature = self._df.drop(columns=['latest_price_sale'])
        target = self._df['latest_price_sale']


        min_max_scaler = preprocessing.MinMaxScaler()
        feature = min_max_scaler.fit_transform(feature)


        # Split into train/test
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        x_train, x_val_and_test, y_train, y_val_and_test = train_test_split(feature, target,
test_size=0.3, random_state=42)

        x_val, x_test, y_val, y_test = train_test_split(x_val_and_test, y_val_and_test,
test_size=0.3, random_state=42)

model = Sequential()

        model =
PredictionModel.generate_nn_sequential_model(model,group,feature.shape[1])

        checkpointer = ModelCheckpoint(filepath="../model/best_weights.hdf5",

        verbose=0, save_best_only=False)  # save best model

        model.fit(x_train, y_train, validation_data=(x_val, y_val),
callbacks=[checkpointer], verbose=1, epochs=500, batch_size=256)

        model.load_weights('../model/best_weights.hdf5')  # load weights from best
model

        pickle.dump(model, open(model_path, 'wb'))

        self._model = model

        pred = (self._model.predict(x_test))

        score = np.sqrt((mean_squared_error(y_test, pred)))

        print("Score (RMSE): {}".format(score))

        r2 = r2_score(y_test, pred)

        print('r2_score :', r2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ไฟล์: main.py

```
from sklearn.model_selection import train_test_split
from SelectionModel import SelectionModel
from PredictionModel import PredictionModel
import json
import pandas as pd
with open('../../dataset/post_data_for_Aun.json', 'rb') as json_data:
    data = json.load(json_data)
df = pd.DataFrame(data)
train_df, predict_df = train_test_split(df, test_size=0.3)
model_selection_train_object = SelectionModel('../model/predict_class_model.pkl',
df=train_df)
model_selection_train_object.train()
# # load model and predict class
model_selection_predict_object =
SelectionModel('../model/predict_class_model.pkl', df=train_df, load_model=True)
predict_df = model_selection_predict_object.predict(predict_df)
# prediction_1 = PredictionModel(df)
prediction_1.train_model('../model/predict_price_model_class0.pkl',df,group=0)
prediction_1.train_model('../model/predict_price_model_class1.pkl',df,group=1)
prediction_1.train_model('../model/predict_price_model_class2.pkl',df,group=2)
# # load model and predict price
model_selection_predict_object = PredictionModel(load_model=True)
predicted_result = model_selection_predict_object.predict(predict_df)
predicted_result.to_csv('../result/predicted_result_all_prediction.csv')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้