การทำนายราคาที่พักบน Airbnb โดยการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณ

PREDICTING ACCOMMODATION PRICES ON AIRBNB USING ENTITY EMBEDDING

นิสิทรา บุญเรือง

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2566

การทำนายราคาที่พักบน Airbnb โดยการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณ

นิสิทรา บุญเรือง

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2566

PREDICTING ACCOMMODATION PRICES ON AIRBNB USING ENTITY EMBEDDING

NISITRA BUNRUANG

A Master's Project Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Data Science)

Faculty of Science, Srinakharinwirot University

2023

สารนิพนธ์

เรื่อง

การทำนายราคาที่พักบน Airbnb โดยการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณ

ของ

นิสิทรา บุญเรือง

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร.นภา แซ่เบ๊)

..... ประธาน

(อาจารย์ ดร.นิดา ชาติวัฒนศิริ)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิริสรรพ เหล่าหะเกียรติ)

ชื่อเรื่อง	การทำนายราคาที่พักบน Airbnb โดยการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณ
ผู้วิจัย	นิสิทรา บุญเรือง
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2566
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. นภา แซ่เบ๊

งานวิจัยนี้มีวัตถุประสงค์เพื่อสร้างแบบจำลองการทำนายราคาที่พักโดยใช้ชุดข้อมูลที่พัก Airbnb ในชุดข้อมูลของกรุงเทพมหานคร จำนวนข้อมูล 20,823 แถว 18 คอลัมน์ จากเว็บไซต์ http://insideairbnb.com// โดยศึกษาเปรียบเทียบการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณโดยใช้การเข้ารหัสแบบ Entity Embedding และ One-hot Encoding สำหรับตัวแปรเชิงกลุ่มที่มีความหลากหลายสูงผ่านแบบจำลอง 4 รูปแบบ ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors และ XGBoost ผลการทดลองแสดงให้เห็นว่าแบบจำลอง Random Forest ให้ผลการดำเนินการที่ดีที่สุดสำหรับการใช้ Entity Embedding โดยมีค่า RMSE 832.56, MAE 587.56 และ R-squared 0.25 ในขณะที่แบบจำลอง XGBoost ให้ผลลัพธ์ที่ดีที่สุดสำหรับการใช้ One-hot Encoding โดยมีค่า RMSE 787.22, MAE 544.42 และ R-squared 0.37 แม้ว่า One-hot Encoding จะให้ผลลัพธ์การทำนายที่ดีกว่าแต่ก็ยังมีค่าความคลาดเคลื่อนสูง อาจเป็นผลจากข้อมูลในชุดข้อมูลยังไม่เพียงพอที่จะสร้างแบบจำลองการทำนายราคาที่พักได้อย่างมีประสิทธิภาพ ดังนั้นการพิจารณาปัจจัยและตัวแปรอื่น เช่น สิ่งอำนวยความสะดวกในที่พัก หรือการออกแบบภายใน อาจช่วยเพิ่มประสิทธิภาพของแบบจำลองการทำนายราคา การวิจัยเพิ่มเติมในประเด็นเหล่านี้อาจนำไปสู่แบบจำลองการทำนายราคาที่พักที่แม่นยำและน่าเชื่อถือมากขึ้น นอกจากนี้ผลการทดลองโดยการนำข้อมูลที่ผ่านการเข้ารหัสโดยใช้เทคนิค Entity Embedding มาทำการแสดงผลยังแสดงให้เห็นถึงความสัมพันธ์ระหว่างกลุ่มต่าง ๆ ซึ่งเป็นอีกทางเลือกในการสำรวจและการวิเคราะห์ข้อมูลเพิ่มเติมได้

คำสำคัญ : Airbnb, Entity Embedding, One-hot Encoding

Title	PREDICTING ACCOMMODATION PRICES ON AIRBNB USING ENTITY EMBEDDING
Author	NISITRA BUNRUANG
Degree	MASTER OF SCIENCE
Academic Year	2023
Thesis Advisor	Assistant Professor Napa Sae-Bae , Ph.D.

This research aims to develop a predictive model for Airbnb accommodation prices using a dataset of 20,823 rows and 18 columns from the Bangkok metropolitan area, and obtained from the website http://insideairbnb.com//. The study compares the transformation of categorical data into numerical data using Entity Embedding and One-hot Encoding for high-diversity categorical variables across four models: Neural Network, Random Forest, K-Nearest Neighbors, and XGBoost. The experimental results demonstrated that the Random Forest with Entity Embedding achieved the most favorable performance metrics. It achieved RMSE of 832.56, MAE of 587.63, and R-squared of 0.25. Conversely, XGBoost demonstrated superior performance when utilizing One-hot Encoding. This model yielded RMSE of 787.22, MAE of 544.42, and R-squared of 0.37. Even though One-hot Encoding had slightly better predictions, it exhibited higher error rates and associated with this technique. This could be attributed to the insufficient data in the dataset to effectively build a predictive model for accommodation prices. Therefore, considering additional factors and variables such as accommodation amenities or interior design, could potentially enhance the performance of the price prediction model. Further research on these aspects promises accommodation price prediction models with higher accuracy and reliability. Moreover, the application of Entity Embedding visualization techniques reveals the relationship among various groups, opening up new avenues for data exploration and analysis.

Keyword : Airbnb, Entity Embedding, One-hot Encoding

กิตติกรรมประกาศ

สารนิพนธ์นี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์จาก ผศ.ดร.นภา แซ่เบ๊ อาจารย์ที่ปรึกษา ที่ให้คำปรึกษา คำแนะนำ ตลอดจนสนับสนุนข้อมูลทางวิชาการและข้อมูลสำหรับทำสารนิพนธ์นี้

ขอกราบขอบพระคุณคณะกรรมกรรมการสอบสารนิพนธ์ที่ได้ให้คำแนะและข้อเสนอแนะสำหรับการปรับปรุงสารนิพนธ์

ขอขอบพระคุณพ่อ แม่ ย่า และป้า ที่เป็นกำลังใจที่สำคัญ และสนับสนุนเรื่องการศึกษาในทุกๆ ด้านจนประสบความสำเร็จและลุล่วงมาจนถึงวันนี้

ขอขอบคุณพี่เอม พี่โบ๊ท พี่มิ้น ปุน เพียว นัน สำหรับกำลังแรง กำลังใจ และคอยให้การช่วยเหลือในเรื่องต่างๆ อย่างดีเสมอมา ขอขอบคุณไอซ์ ปอนด์ นิ พี่เกียร์ พี่บี พี่เอ พี่ต่าย พี่ฟลุ๊ก ที่มีส่วนช่วยเหลือทั้งในเรื่องการเรียน การจัดทำเอกสาร และคำแนะนำต่างๆ เพื่อให้สารนิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณทุกบทเพลงที่คอยปลอบประโลมจิตใจในยามที่ท้อ

สุดท้ายขอขอบคุณตัวข้าพเข้าในวัย 28 ปี ที่ได้มอบสารนิพนธ์ฉบับนี้ให้เป็นของขวัญแก่ข้าพเจ้าในวัย 29 ปี

นิสิทรา บุญเรือง

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

การท่องเที่ยวเป็นประสบการณ์ที่ทำให้แต่ละคนได้ออกเปิดโลกทัศน์ให้กว้างขึ้น ไปเรียนรู้วัฒนธรรมที่แตกต่าง สร้างความทรงจำใหม่ ๆ เปิดรับมุมมองใหม่ ๆ และค้นพบความสวยงามที่มีอยู่นอกเหนือจากสภาพแวดล้อมที่คุ้นเคย เมื่อทราบแล้วว่าจะเดินทางไปยังสถานที่ใด การวางแผนการท่องเที่ยวจึงถือเป็นสิ่งสำคัญโดยเฉพาะเรื่องของที่พักแรม

ปัจจุบันการหาที่พักแรมมีความสะดวกและรวดเร็วโดยใช้เว็บไซต์หรือแอปพลิเคชันการจองที่พักออนไลน์เพื่อค้นหาที่พักที่เป็นจุดหมายปลายทาง นอกจากนี้ยังสามารถกรองผลการค้นหาตามความต้องการเฉพาะ เช่น ราคา สิ่งอำนวยความสะดวก หรือคะแนนรีวิวจากผู้เข้าพักก่อนหน้า ทั้งยังกำหนดได้ว่าต้องการที่พักประเภทใด เช่น โรงแรม วิลล่า บ้านพักของผู้คนในท้องถิ่น หอพัก หรือคอนโดมิเนียม ซึ่งสามารถเลือกให้ตรงกับความต้องการและงบประมาณของผู้เข้าพัก แต่ในช่วงไม่กี่ปีที่ผ่านมาวิธีการเดินทางของผู้คนและประสบการณ์ที่ได้รับการจากการเข้าพักแรมถือเป็นการเปลี่ยนแปลงครั้งสำคัญเพราะโรงแรมและรีสอร์ทแบบดั้งเดิมไม่ได้เป็นทางเลือกเดียวสำหรับนักเดินทางอีกต่อไป

ในปี 2007 Airbnb ได้ถือกำเนิดขึ้น มีการพัฒนาอย่างต่อเนื่อง และขึ้นเป็นหนึ่งในแพลตฟอร์มท่องเที่ยวที่มีชื่อเสียงมากที่สุดในโลก Airbnb เป็นจุดเปลี่ยนครั้งสำคัญ เพราะเป็นจุดเริ่มต้นที่ให้เจ้าของบ้านประกาศปล่อยเช่าอะพาร์ตเมนต์ทั้งหลัง หรือห้องว่างภายในบ้านภายในแพลตฟอร์ม และหลังจากนั้นไม่นานก็เกิดแนวทางที่พักใหม่ ๆ เช่น บ้านต้นไม้ บ้านหลังเล็ก ซึ่งเป็นแนวคิดที่เริ่มต้นจากการมอบทางเลือกที่ประหยัดและไม่เหมือนใครให้กับนักเดินทาง เพราะที่พักของ Airbnb ต่างจากห้องพักในโรงแรมมาตรฐานในเรื่องรูปแบบของที่พัก ที่ตั้ง สิ่งอำนวยความสะดวก แม้กระทั่งการมีปฏิสัมพันธ์กับเจ้าของที่พักและสถานที่พัก ทำให้เกิดความรู้สึกถึงความเป็นตัวตนที่แท้จริงและการดื่มด่ำไปกับสถานที่ ซึ่งความสำเร็จของ Airbnb อยู่ที่ความสามารถในการเชื่อมโยงเจ้าของที่พักและผู้เข้าพักโดยมุ่งเน้นถึงประสบการณ์ที่ทุกคนรู้สึกเหมือนอยู่บ้านไม่ว่าจะไปที่ไหนก็ตาม Airbnb จึงมักเป็นตัวเลือกอันดับแรกที่นักเดินทางใช้จองที่พัก นอกจากนี้ยังเป็นแพลตฟอร์มออนไลน์ที่สามารถสร้างผลกำไรให้แก่ธุรกิจที่พักอีกด้วย จากสถิติของ Airbnb เผยว่าจำนวนคืนการจองที่พักบน Airbnb ทั่วประเทศไทยในปี 2021 เพิ่มขึ้นถึง 240% หากเทียบกับปี 2020 ที่เป็นช่วงระบาดของโควิด-19 โดยกรุงเทพมหานครเป็นจุดหมายปลายทางที่นิยมอันดับ 1 ของนักเดินทาง และในปี 2023 กรุงเทพมหานครก็ติดอันดับ 5 เมืองท่องเที่ยวยอดนิยมระดับโลก

บน Airbnb ด้วย ซึ่งการตัดสินใจในการเลือกที่พักต้องพิจารณาปัจจัยต่าง ๆ อย่างรอบคอบ เช่น สถานที่ตั้ง ราคา ขนาด ความปลอดภัย รีวิว ความชอบส่วนตัว โดยเฉพาะตัวแปรราคามีผลอย่างมากต่อการตัดสินใจเลือกจองของผู้เข้าพักเนื่องจากต้องดูองค์ประกอบอื่น ๆ ประกอบในการพิจารณาการจองด้วย เพราะถ้าหากมีราคาที่สูงแต่มีคะแนนรีวิวต่ำนั่นแสดงว่าที่พักนั้นอาจจะไม่ตอบสนองความต้องการของผู้เข้าพักเท่าที่ควร

สำหรับงานวิจัยนี้ได้นำข้อมูลมาจากเว็บไซต์ http://insideairbnb.com/get-the-data/ ที่เก็บรวบรวมข้อมูลต่าง ๆ เกี่ยวกับที่พักบน Airbnb ในหลาย ๆ เมือง หลาย ๆ ประเทศทั่วโลก เช่น พื้นที่ ประเภทที่พัก จำนวนคืนที่เข้าพัก คะแนนรีวิว ด้วยตัวแปรเหล่านี้เองผู้วิจัยจึงมีแนวคิดในการทำนายราคาที่พักบน Airbnb ในพื้นที่กรุงเทพมหานคร โดยการนำตัวแปรที่มีค่าความหลากหลายสูงมาแปลงให้อยู่ในรูปแบบของการเข้ารหัสด้วยวิธี Entity Embedding และ One-hot Encoding เพื่อทำนายตัวแปรราคาว่ามีความใกล้เคียงกับราคาที่ถูกตั้งไว้บน Airbnb หรือไม่

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อสร้างแบบจำลองการทำนายราคาที่พักบน Airbnb โดยอาศัยตัวแปรที่เกี่ยวข้อง ได้แก่ neighborhood, room type, price, minimum nights, number of reviews, reviews per month, calculated host listings count และ availability 365 โดยศึกษาเปรียบเทียบกระบวนการปรับปรุงข้อมูล (transform data) ในตัวแปรตาม (dependent variable) และตัวแปรอิสระ (independent variable)

2. เพื่อจัดการกับตัวแปรที่มีค่าความหลากหลายสูง (high cardinality) ด้วยวิธีเข้ารหัสแบบ Entity Embedding และ One-hot Encoding

3. สร้างรูปแบบของแบบจำลองการทำนาย 4 รูปแบบ ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองแต่ละประเภท

4. เพื่อสร้างการแสดงภาพ (visualization) ให้เห็นความสัมพันธ์ของข้อมูลที่มีความหลากหลายสูง

1.3 ขอบเขตงานวิจัย

1. ข้อมูลที่ใช้ในงานวิจัยฉบับนี้เป็นชุดข้อมูลสาธารณะจาก http://insideairbnb.com ที่เก็บรวบรวมจาก Airbnb โดยเลือกชุดข้อมูลของกรุงเทพมหานครเพื่อนำมาสร้างแบบจำลองโดยตัวแปรที่นำมาใช้ในการศึกษา ได้แก่

- Neighborhood หรือชื่อของย่านหรือพื้นที่ที่รายการ Airbnb ตั้งอยู่
- Room type หรือประเภทของห้องพัก
- Price หรือราคารายวันที่ผู้เข้าพักต้องจ่ายในสกุลเงินบาทสำหรับการเข้าพักในกรุงเทพมหานคร
- Minimum nights หรือจำนวนขั้นต่ำของคืนที่ผู้เข้าพักต้องจองในรายการเข้าพัก
- Number of reviews หรือจำนวนรวมของรีวิวของรายการ Airbnb
- Reviews per month หรือจำนวนค่าเฉลี่ยของรีวิวต่อเดือน
- Calculated host listings count จำนวนรวมของรายการที่พักที่โฮสต์มีในกรุงเทพมหานคร
- Availability 365 หรือความพร้อมใช้งานของรายการที่พักใน 365 วันถัดไป

2. ในตัวแปรตาม (dependent variable) มีการปรับปรุงข้อมูลด้วยวิธี StandardScaler กับคอลัมน์ที่เป็นตัวแปรเชิงปริมาณทั้งหมด และในตัวแปรอิสระ (independent variable) มีการปรับการกระจายตัวของข้อมูล 2 วิธี คือ StandardScaler และ Logarithm ในคอลัมน์ price และเพื่อวัดประสิทธิภาพของแบบจำลอง โดยมีการทำการย้อนกลับของข้อมูล (inverse) ในตัวแปรตัวแปรอิสระเพื่อให้ข้อมูลกลับมาสู่รูปแบบเดิมก่อนที่จะนำไปใช้งานต่อ

3. จัดการข้อมูลตัวแปรอินพุต neighborhood ซึ่งเป็นตัวแปรที่มีค่าความหลากหลายสูง (high cardinality) โดยใช้วิธีการเข้ารหัสแบบ Entity Embedding และ One-hot Encoding

4. สร้างแบบจำลองการทำนายราคาที่พักบน Airbnb ที่มีการปรับข้อมูลโดยการทำ StandardScaler และ Logarithm 4 รูปแบบคือ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost และเปรียบเทียบประสิทธิภาพการทำนายโดยอาศัยตัวชี้วัดประสิทธิภาพสำหรับการทำนายแบบการวิเคราะห์เชิงถดถอย (Regression) ได้แก่ Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) และ R-squared

1.4 ขั้นตอนการดำเนินงานวิจัย

1. ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวกับการเข้ารหัสแบบ Entity Embedding และ One-hot Encoding ในการจัดการกับตัวแปรที่มีความหลากหลายสูง (high cardinality) และกรณีศึกษาต่าง ๆ ที่เกี่ยวข้องกับ Airbnb รวมถึงประวัติความเป็นมา และการเข้าใช้งานการจองที่พักผ่านแอปพลิเคชัน Airbnb เพื่อใช้ในการแก้ปัญหาในงานวิจัยนี้

2. ศึกษาการทำ StandardScaler และ Logarithm ซึ่งเป็นกระบวนการปรับข้อมูลที่มีการกระจายแบบไม่เหมาะสม เพื่อให้แบบจำลองมีประสิทธิภาพในการเรียนรู้และทำนายได้ดีขึ้น

3. ศึกษางานวิจัยที่เกี่ยวข้องกับโครงข่ายประสาทเทียม (neural network) ที่มีการจัดการกับตัวแปรที่มีความหลากหลายสูงด้วยการเข้ารหัสแบบ Entity Embedding พร้อมศึกษาโครงสร้างและแนวคิดที่ถูกใช้ในการพัฒนา

4. กำหนดขอบเขตของการวิจัยซึ่งประกอบด้วย

- ชุดข้อมูล คือ รวบรวมข้อมูล และเตรียมข้อมูล
- ในงานวิจัยนี้ใช้การจัดการข้อมูลที่มีการกระจายแบบไม่เหมาะสมโดยการทำ StandardScaler และ Logarithm
- ในการวิจัยนี้เลือกใช้การเข้ารหัสแบบ One-hot Encoding และ Entity Embedding เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลอง

5. นำเข้าข้อมูลชุดเดียวกันในแบบจำลอง คือ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost และเปรียบเทียบประสิทธิภาพระหว่างเข้ารหัสแบบ Entity Embedding และ One-hot Encoding

6. ปรับค่าพารามิเตอร์ต่าง ๆ ของแบบจำลองที่ให้ผลลัพธ์ดีที่สุด เพื่อหาค่าพารามิเตอร์ที่ทำให้แบบจำลองทำนายผิดพลาดน้อยที่สุด

7. เปรียบเทียบแบบจำลองชนิดต่าง ๆ ว่ามีประสิทธิภาพเป็นอย่างไร เพื่อหาแบบจำลองที่ดีที่สุดสำหรับการทำนายผล

8. วิเคราะห์ สรุป และอภิปรายผลงานวิจัย ถึงข้อดี ข้อเสีย และข้อจำกัดต่าง ๆ ของเทคนิคที่นำมาใช้ในแบบจำลอง

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1. แบบจำลองการทำนายราคาที่พักสามารถนำไปวิเคราะห์ภาพรวมของการทำธุรกิจบน Airbnb ว่าตัวแปรต่าง ๆ ส่งผลต่อราคาที่พักอย่างไร

2. รูปแบบการสร้างแบบจำลองสามารถนำไปใช้กับแพลตฟอร์มหรือชุดข้อมูลอื่นที่มีโครงสร้างของข้อมูลคล้ายกัน

3. การเข้ารหัสแบบ entity embedding สามารถนำไปใช้ในการวิเคราะห์ความสัมพันธ์ระหว่างพื้นที่ต่าง ๆ เพื่อกำหนดกลยุทธ์ด้านการทำธุรกิจที่พักบนแพลตฟอร์มได้ดียิ่งขึ้น

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

สำหรับงานวิจัยนี้ผู้วิจัยได้รวบรวมทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการเข้ารหัสด้วยวิธี Entity Embedding และ One-Hot Encoding เพื่อนำมาจัดการกับตัวแปรที่มีความหลากหลายสูง (high cardinality) และงานวิจัยที่เกี่ยวข้องกับหลักการทำงานของแบบจำลองที่ใช้สำหรับปัญหาแบบการวิเคราะห์เชิงถดถอย (regression)

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การจัดการข้อมูลที่ขาดหาย (Handling Missing Values)

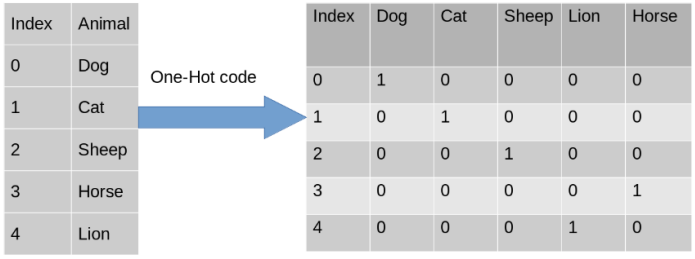
Handling Missing Values คือ การจัดการข้อมูลที่ขาดหาย ใช้สำหรับการตรวจจับและจัดการข้อมูลที่ขาดหาย เช่น การแทนค่าค่าที่ขาดหายด้วยค่าเฉลี่ย หรือการใช้เทคนิคทางสถิติในการจัดการข้อมูลที่ขาดหาย

2.1.2 กระบวนการเข้ารหัส (Encoding Method)

กระบวนการเข้ารหัสเป็นกระบวนการที่ใช้ในการแปลงข้อมูลจากรูปแบบหนึ่งไปยังรูปแบบหนึ่งเพื่อให้ข้อมูลมีรูปแบบที่เหมาะสมสำหรับการนำไปใช้ในแบบจำลองหรือการวิเคราะห์ข้อมูลอื่น ๆ แบ่งออกเป็นหลายประเภท ได้แก่

2.1.2.1 One-Hot Encoding

One-Hot Encoding คือ การแปลงข้อมูลหมวดหมู่ (categorical data) ที่ใช้ในการประมวลผลข้อมูลและสร้างแบบจำลองการเรียนรู้ของเครื่อง โดยจะเปลี่ยนข้อมูลหมวดหมู่ให้กลายเป็นตัวเลขที่สามารถนำไปใช้ในการคำนวณได้ โดยสร้างเวกเตอร์ (vector) ของตัวเลขที่มีองค์ประกอบที่ทุกองค์ประกอบแทนสถานะของค่าแต่ละค่า



ภาพประกอบ 1 การแปลงข้อมูลด้วยการเข้ารหัสแบบ One-hot Encoding

ที่มา : (Saxena, 2023)

```
import pandas as pd

# Sample dataset with a categorical column
data = {'Color': ['Red', 'Blue', 'Green', 'Red', 'Green']}
df = pd.DataFrame(data)

# Perform one-hot encoding using Pandas
one_hot_encoded = pd.get_dummies(df, columns=['Color'])
```

ภาพประกอบ 2 code แสดงการสร้าง One-hot Encoder

ที่มา : (Baruah, 2023)

แบบจำลองที่เหมาะกับการทำ One-Hot Encoding คือแบบจำลองที่ทำงานได้ดีกับข้อมูลประเภทตัวเลข ซึ่งแบบจำลองที่นิยม ได้แก่ Linear Regression, Logistic Regression, Decision Tree, Support Vector Machines โดยการเลือกแบบจำลองที่เหมาะสมในการทำ One-Hot Encoding นั้นควรพิจารณาจากประเภทของงานเป็นอันดับแรก ซึ่งข้อดีของการทำ One-Hot Encoding คือการแปลงข้อมูลให้คอมพิวเตอร์สามารถเรียนรู้และเข้าใจข้อมูลที่เป็นตัวเลขได้ ซึ่งรวมไปถึงการคำนวณทางคณิตศาสตร์ต่าง ๆ ข้อเสียคือการใช้ทรัพยากรของเครื่องที่เพิ่มขึ้นตามจำนวนข้อมูล เพราะข้อมูลจะถูกแปลงให้อยู่ในรูปแบบที่ประกอบไปด้วยเลข 0 เป็นจำนวนมาก หากข้อมูลหมวดหมู่มีหลายประเภทก็จะทำให้ต้องการหน่วยความจำในการทำงานเยอะขึ้น

2.1.2.2 Label Encoding

Label Encoding คือ การใช้ตัวเลขแทนค่าข้อมูลโดยการกำหนดตัวเลขที่ไม่ซ้ำกันให้กับแต่ละหมวดหมู่ ซึ่งไม่จำเป็นต้องสร้างเวกเตอร์แบบ One-Hot Encoding ทำให้ลดขนาดของข้อมูล

ตัวอย่าง ถ้ามีคอลัมน์ “color” ซึ่งมีค่าเป็น “red” “green” “blue” การใช้ Label Encoding จะแปลงข้อมูล ดังตาราง 1

ตาราง 1 การแปลงตัวแปรหมวดหมู่ (categorical data) ให้อยู่ในรูปแบบ Label Encoding

Color	Label Encoding
Red	0
Green	1
Blue	2

```
from sklearn.preprocessing import LabelEncoder

# Sample dataset with a categorical column
data = {'Size': ['Small', 'Medium', 'Large', 'Medium', 'Small']}
df = pd.DataFrame(data)

# Initialize the LabelEncoder
label_encoder = LabelEncoder()

# Fit and transform the 'Size' column
df['Size_encoded'] = label_encoder.fit_transform(df['Size'])
```

ภาพประกอบ 3 code แสดงการสร้าง Label Encoder

ที่มา : (Baruah, 2023)

2.1.2.3 Ordinal Encoding

Ordinal Encoding คือ การแปลงข้อมูลที่เป็นตัวแปรหมวดหมู่ให้เป็นตัวเลขตามลำดับหรือตามลำดับความสำคัญของข้อมูล โดยการแปลงนี้คือการกำหนดตัวเลขให้กับแต่ละหมวดหมู่โดยคำนึงถึงลำดับที่มีความหมาย ซึ่งสามารถใช้กับข้อมูลที่มีการเรียงลำดับ เช่น ระดับการศึกษา (เช่น มัธยมศึกษาตอนปลาย ปริญญาตรี ปริญญาโท ปริญญาเอก) หรือระดับความพึงพอใจ (เช่น ต่ำ ปาน-กลาง สูง) ซึ่งจะช่วยให้สามารถใช้ข้อมูลที่เป็นหมวดหมู่ในการฝึกแบบจำลองได้ง่ายขึ้น เนื่องจากแบบจำลองสามารถทำนายได้ตรงกับข้อมูลที่มีลำดับหรือความสำคัญของข้อมูล

ตัวอย่าง ถ้ามีคอลัมน์ “Educational Level” ที่มีค่าเป็น “Primary School” “Secondary School” “University” การใช้ Ordinal Encoding จะแปลงข้อมูลเป็นรหัสตัวเลข ดังนี้

ตาราง 2 การแปลงตัวแปรหมวดหมู่ (categorical data) ให้อยู่ในรูปแบบ Ordinal Encoding

Educational Level	Ordinal Encoding
Primary School	1
Secondary School	2
University	3

```python
import pandas as pd
import category_encoders as ce

# Sample dataset with an ordinal categorical column
data = {'Education_Level': ['High School', 'Bachelor\'s', 'Master\'s',
                            'Bachelor\'s', 'High School']}
df = pd.DataFrame(data)

# Define the order of categories
education_order = ['High School', 'Bachelor\'s', 'Master\'s']

# Initialize the OrdinalEncoder with specified order
ordinal_encoder = ce.OrdinalEncoder(mapping=[{'col': 'Education_Level', 'mapping':
 {level: index for index, level in enumerate(education_order)}}])

# Fit and transform the DataFrame
df_encoded = ordinal_encoder.fit_transform(df)

# Display the DataFrame with ordinal encoding
print(df_encoded)
```

ภาพประกอบ 4 code แสดงการสร้าง Ordinal Encoder

ที่มา : (Baruah, 2023)

2.1.2.4 Count Encoding

Count Encoding คือ การแปลงข้อมูลที่เป็นตัวแปรหมวดหมู่ให้กลายเป็นตัวเลขโดยการนับจำนวนครั้งที่แต่ละค่าหมวดหมู่ปรากฏในข้อมูล แล้วกำหนดตัวเลขนับนั้นให้กับแต่ละค่าหมวดหมู่ นั่นหมายความว่าค่าที่มีจำนวนครั้งมากจะได้รับตัวเลขนับที่สูงขึ้น ซึ่งมักใช้กับข้อมูลที่มีจำนวนหมวดหมู่มากและต้องการแปลงเป็นตัวเลขเพื่อใช้ในการฝึกแบบจำลอง โดยเฉพาะในกรณีที่ข้อมูลไม่ได้มีความเชื่อมโยงหรือลำดับที่เป็นรูปแบบชัดเจน

```python
import pandas as pd
from sklearn.model_selection import train_test_split
import category_encoders as ce

# Generate a dummy dataset with categorical variables
data = {
    'Color': ['Red', 'Blue', 'Green', 'Red', 'Red', 'Blue', 'Green'],
    'Size': ['Small', 'Medium', 'Large', 'Medium', 'Small', 'Small', 'Medium'],
    'Label': [1, 0, 1, 1, 0, 0, 1]
}

df = pd.DataFrame(data)

# Split the data into training and test sets
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)

# Initialize the CountEncoder
count_encoder = ce.CountEncoder()

# Fit the encoder on the training data
count_encoder.fit(train_df[['Color', 'Size']])

# Transform both the training and test datasets
train_encoded = count_encoder.transform(train_df[['Color', 'Size']])
test_encoded = count_encoder.transform(test_df[['Color', 'Size']])

# Display the encoded datasets
print("Training Data (After Count Encoding):\n", train_encoded)
print("\nTest Data (After Count Encoding):\n", test_encoded)
```

ภาพประกอบ 5 code แสดงการสร้าง Count Encoder

ที่มา : (Baruah, 2023)

การเข้ารหัสแบบ Count Encoding จะเป็นการช่วยลดจำนวนคอลัมน์ในข้อมูล โดยเฉพาะเมื่อมีตัวแปรหมวดหมู่มาก ๆ และจะช่วยรักษาความถี่ของข้อมูลในชุดข้อมูลต้นฉบับ ซึ่งเป็นข้อมูลที่มีความสำคัญสำหรับการวิเคราะห์และการฝึกแบบจำลอง แต่อาจทำให้สูญเสียข้อมูลที่มีความหมายหรือความสัมพันธ์อื่น ๆ ที่อาจมีอยู่ระหว่างหมวดหมู่

2.1.2.5 Target Encoding

Target Encoding คือ การแปลงข้อมูลที่เป็นตัวแปรแบบหมวดหมู่ให้กลายเป็นตัวเลขโดยใช้ข้อมูลจากตัวแปรเป้าหมาย (target variable) เป็นตัวชี้วัด โดยปกติจะใช้เมื่อตัวแปรเป้าหมายเป็นตัวแปรต่อเนื่องหรือตัวแปรสัมพัทธ์ เช่น การทำนายราคาของบ้านโดยใช้ข้อมูลตัวแปรอื่นเป็นตัวแปรอิสระ ซึ่งวิธีการเข้ารหัสแบบ Target Encoding จะทำงานโดยคำนึงถึงค่าเฉลี่ยของตัวแปรเป้าหมายสำหรับแต่ละหมวดหมู่ของตัวแปรหมวดหมู่ แล้วนำค่าเหล่านี้มาใช้แทนตัวแปรหมวดหมู่ในการฝึกแบบจำลอง

```python
import pandas as pd
from sklearn.model_selection import train_test_split
import category_encoders as ce

# Generate a dummy dataset with categorical variables
data = {
    'Color': ['Red', 'Blue', 'Green', 'Red', 'Red', 'Blue', 'Green'],
    'Size': ['Small', 'Medium', 'Large', 'Medium', 'Small', 'Small', 'Medium'],
    'Label': [1, 0, 1, 1, 0, 0, 1]
}

df = pd.DataFrame(data)

# Split the data into training and test sets
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)

# Initialize the MeanEncoder
mean_encoder = ce.TargetEncoder()

# Fit the encoder on the training data
mean_encoder.fit(train_df[['Color', 'Size']], train_df['Label'])

# Transform both the training and test datasets
train_encoded = mean_encoder.transform(train_df[['Color', 'Size']])
test_encoded = mean_encoder.transform(test_df[['Color', 'Size']])

# Display the encoded datasets
print("Training Data (After Mean Encoding):\n", train_encoded)
print("\nTest Data (After Mean Encoding):\n", test_encoded)
```

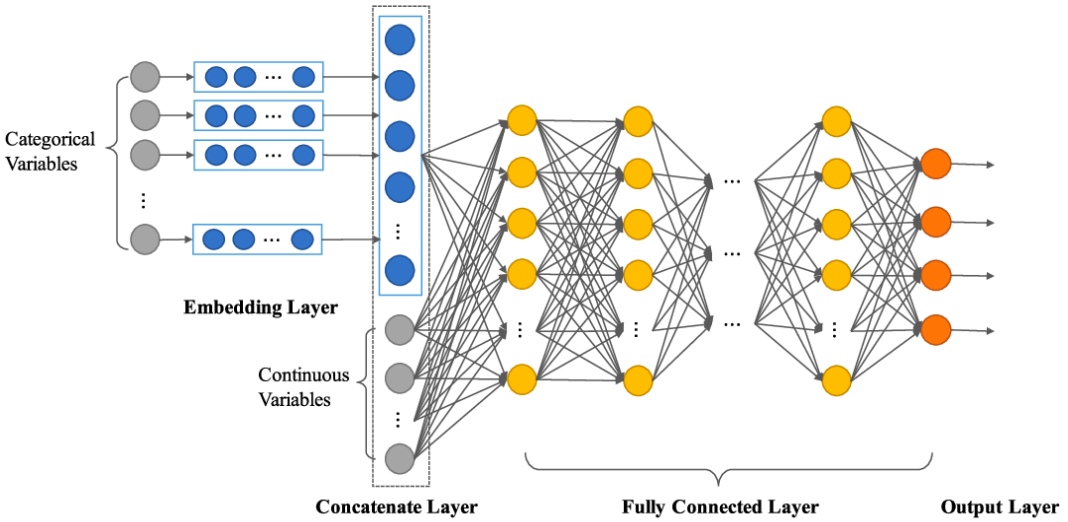
ภาพประกอบ 6 code แสดงการสร้าง Target Encoder

ที่มา : (Baruah, 2023)

2.1.2.6 Entity Embedding

Entity Embedding คือ การแปลงข้อมูลที่เป็นตัวแปรแบบหมวดหมู่ให้กลายเป็นตัวเลขโดยการเรียนรู้ลักษณะและความสัมพันธ์ของข้อมูลในขั้นตอนการฝึกแบบจำลอง (training process) ซึ่งเป็นเทคนิคที่ได้รับความนิยมมากในงานที่เกี่ยวกับการเรียนรู้เชิงลึก (deep learning) โดยเฉพาะในงานที่เกี่ยวข้องกับข้อมูลที่เป็นหมวดหมู่ เช่น การทำนายราคาของบ้านจากข้อมูลเชิงภูมิศาสตร์และลักษณะของที่อยู่ โดยการทำงานของ Entity Embedding จะใช้โครงสร้างโครงข่ายประสาทเทียม (neural network) ในการฝึกแบบจำลอง โดยมีการเรียนรู้ความสัมพันธ์ระหว่างตัวแปรหมวดหมู่และตัวแปรเป้าหมาย (target variable) ซึ่งมักจะเป็นตัวแปรต่อเนื่อง เช่น ราคาของบ้าน โดย Entity Embedding จะแปลงตัวแปรหมวดหมู่ให้อยู่ในรูปแบบของเวกเตอร์ที่มีมิติต่ำลง เพื่อให้แบบจำลองสามารถเรียนรู้และเข้าใจความสัมพันธ์ของข้อมูลได้ง่ายขึ้น

โครงข่ายประสาทเทียมที่ใช้การเข้ารหัสแบบ Entity Embedding มักถูกออกแบบให้มีโครงสร้างหลายชั้นเพื่อให้สามารถประมวลผลข้อมูลให้มีประสิทธิภาพมากขึ้น เนื่องจากการใช้ Entity Embedding เป็นการแปลงข้อมูลเชิงนามธรรม เช่น ชื่อบุคคล ชื่อสถานที่ หรือชื่อผลิตภัณฑ์ ให้เป็นข้อมูลเชิงตัวเลขในรูปแบบของเวกเตอร์ ซึ่งเวกเตอร์เหล่านี้มักมีมิติสูง เช่น 100 หรือ 1,000 มิติ ทำให้การประมวลผลข้อมูลด้วยโครงข่ายประสาทเทียมชั้นเดียวอาจไม่เพียงพอ



ภาพประกอบ 7 โครงข่ายประสาทเทียมขนาดลึกที่ใช้เทคนิค Entity Embeddingรวมถึงโครงสร้างที่ประกอบด้วยชั้น Embedding ชั้น Concatenate ชั้น Fully Connected และชั้น Output

ที่มา : (Ma & Zhang, 2020)

โครงสร้างหลายชั้นของโครงข่ายประสาทเทียมจะช่วยให้สามารถประมวลผลข้อมูลเชิงลึกที่ซ่อนอยู่ภายในเวกเตอร์ของ Entity Embedding ได้ดีขึ้น โดยในแต่ละชั้นของโครงข่ายประสาทเทียมจะทำหน้าที่ประมวลผลข้อมูลในลักษณะที่แตกต่างกัน โดยประกอบไปด้วย

1. Embedding Layer เป็นชั้นแรกที่รับข้อมูลแบบหมวดหมู่และแปลงค่าของข้อมูลหมวดหมู่ให้อยู่ในรูปแบบเวกเตอร์หลายมิติ โดยใช้การเข้ารหัสแบบ Entity Embedding เพื่อให้คอมพิวเตอร์สามารถจัดเก็บและเรียกใช้ข้อมูลในรูปแบบที่เหมาะสมได้

2. Concatenate Layer หลังจากที่ข้อมูลจากชั้น Embedding Layer ถูกแปลงเป็นเวกเตอร์แล้ว ชั้น Concatenate Layer จะรับเวกเตอร์เหล่านั้นและนำมารวมเข้าด้วยกัน โดยรวมข้อมูลจากหลายแหล่งเข้าด้วยกันเพื่อสร้างเวกเตอร์ข้อมูลที่ใหญ่ขึ้น

3. Fully Connected Layer มีหน้าที่ประมวลผลข้อมูลที่ได้จากชั้น Concatenate Layer โดยการนำเอาข้อมูลมาผ่านการคูณเมตริกซ์และฟังก์ชันกิริยาเชิงเส้น (Linear Activation Function) เพื่อสร้างพฤติกรรมของโครงข่ายในการเรียนรู้และทำนายข้อมูล

4. Output Layer เป็นชั้นที่ได้รับข้อมูลมาจากชั้น Fully Connected Layer และใช้ในการสร้างผลลัพธ์ เช่น การทำนายค่าหรือการจำแนกหมวดหมู่ข้อมูล

2.1.3 **การปรับปรุงข้อมูล** (Transform Data)

กระบวนการที่ใช้เปลี่ยนแปลงหรือปรับปรุงข้อมูลให้เหมาะสมกับการใช้ในการวิเคราะห์หรือการประมวลผลต่อไป โดยมีวัตถุประสงค์หลักคือการทำให้ข้อมูลอยู่ในรูปแบบที่เข้าใจง่ายขึ้น หรือเพื่อให้แบบจำลองทำงานได้อย่างแม่นยำและมีประสิทธิภาพมากที่สุด โดยการปรับปรุงข้อมูลสามารถทำได้หลายวิธีตามลักษณะของข้อมูลและวัตถุประสงค์ต่าง ๆ ได้แก่

2.1.3.1 Scaling

Scaling คือ กระบวนการที่ใช้ในการปรับค่าของข้อมูลให้อยู่ในช่วงหรือเท่ากับของค่าที่กำหนด เป็นวิธีที่ช่วยให้ข้อมูลมีการกระจายที่เหมาะสมและช่วยลดความผิดพลาดในการวิเคราะห์หรือสร้างแบบจำลอง โดยวิธีที่พบบ่อยมีดังนี้

1. Min-Max Scaling คือ การปรับข้อมูลให้มีค่าอยู่ในช่วงระหว่างที่กำหนดไว้ เช่น ช่วงค่า 0 ถึง 1 หรือช่วงค่า -1 ถึง 1

2. Standardization (Z-score Scaling) คือ การปรับข้อมูลให้มีค่าเฉลี่ยเท่ากับ 0 และมีความแปรปรวนเท่ากับ 1

2.1.3.2 Log Transformation

Log Transformation คือ กระบวนการที่ใช้ในการเปลี่ยนแปลงค่าของข้อมูลโดยการใช้ฟังก์ชันลอการิทึม (logarithm) โดยที่ค่าของข้อมูลที่มีการกระจายแบบปกติหรือเชิงเส้นจะถูกแปลงให้มีการกระจายที่แปรปรวนน้อยลง

2.1.3.3 Binning or Bucketing

Binning or Bucketing คือ กระบวนการที่ใช้ในการแบ่งข้อมูลตามกลุ่มหรือช่วง (bins) เพื่อให้ข้อมูลมีโครงสร้างและมีความสัมพันธ์ที่ชัดเจน เช่น การแบ่งข้อมูลตามช่วงอายุ ราคาสินค้า ระดับความเสี่ยง โดยใช้ขอบเขตหรือเกณฑ์ที่ต่างกัน ได้แก่

1. Equal Width Binning คือ การแบ่งข้อมูลเป็นกลุ่มโดยการกำหนดช่วงที่มีความกว้างเท่า ๆ กัน โดยไม่คำนึงถึงการกระจายของข้อมูล เช่น แบ่งอายุเป็นกลุ่ม 1-10 ปี, 11-20 ปี หรือ 21-30 ปี เป็นต้น

2. Equal Frequency Binning คือ การแบ่งข้อมูลเป็นกลุ่มโดยการแบ่งตามจำนวนข้อมูลที่เท่ากันในแต่ละกลุ่มเพื่อให้มีการกระจายที่สมดุล เช่น แบ่งราคาสินค้าเป็นกลุ่มที่มีจำนวนสินค้าเท่า ๆ กันในแต่ละกลุ่ม

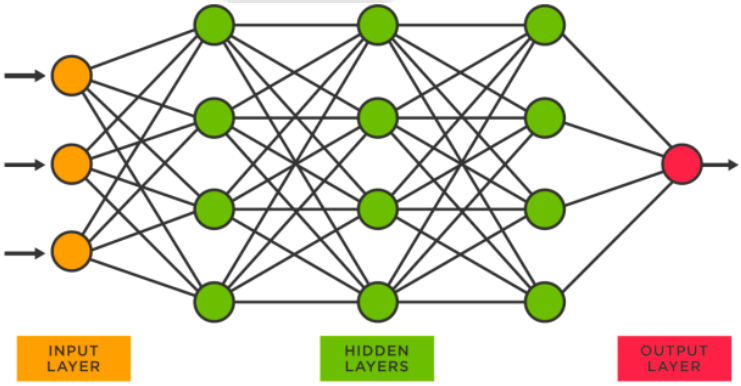
3. Custom Binning คือ การแบ่งกลุ่มข้อมูลในตัวแปรต่อเนื่องเป็นกลุ่ม ๆ ที่มีความหมายเฉพาะ เพื่อให้ง่ายต่อการวิเคราะห์และการจัดการข้อมูล เช่น กลุ่มเด็ก กลุ่มผู้ใหญ่ กลุ่มผู้สูงอายุ เป็นต้น

2.1.4 **แบบจำลองสำหรับการเรียนรู้ของเครื่อง** (Modeling)

ในงานวิจัยฉบับนี้เป็นปัญหาเชิงถดถอย (regression) ซึ่งเป็นประเภทหนึ่งของปัญหาทางสถิติที่เกี่ยวข้องกับการพยากรณ์หรือทำนายค่าตามของตัวแปรตาม (dependent variable) จากตัวแปรอิสระ (independent variable) ซึ่งในงานวิจัยนี้ได้มีการนำแบบจำลองที่นำมาใช้สำหรับการวัดประสิทธิภาพ ดังนี้

2.1.4.1 Neural Network

Neural Network เป็นแบบจำลองการเรียนรู้ของเครื่องที่ได้รับแรงบันดาลใจจากโครงสร้างของสมองมนุษย์ ประกอบด้วยหน่วยประมวลผลจำนวนมากที่เชื่อมต่อกันแบบเครือข่าย แต่ละหน่วยประมวลผลทำหน้าที่เป็นนิวรอนในสมองมนุษย์ โดยสามารถเรียนรู้ความสัมพันธ์ที่ซับซ้อนระหว่างข้อมูลต่าง ๆ ได้ เช่น ความสัมพันธ์ระหว่างภาพกับคำอธิบายภาพ หรือความสัมพันธ์ระหว่างคำกับคำอื่น ๆ ดังนั้น Neural Network จึงเป็นแบบจำลองที่มีประสิทธิภาพสูงสำหรับงานต่าง ๆ เช่น การจำแนกประเภท (classification) การถอดรหัส (decoding) และการสร้างแบบจำลอง (modeling)



ภาพประกอบ 8 โครงสร้างของ Neural Network

ที่มา : (Amzhao, 2023)

Neural Network ประกอบด้วยหน่วยประมวลผลจำนวนมากที่เชื่อมต่อกันแบบเครือข่าย แต่ละหน่วยประมวลผลรับข้อมูลจากหน่วยประมวลผลอื่น ๆ และส่งข้อมูลไปยังหน่วยประมวลผลอื่น ๆ ซึ่งโดยทั่วไปแล้ว Neural Network ประกอบด้วยอย่างน้อย 3 ส่วนหลัก ได้แก่

1. ชั้นนำเข้า (input layer) คือ จุดเริ่มต้นของโครงข่ายประสาท มีหน้าที่รับข้อมูลนำเข้า (input data) เข้าสู่แบบจำลอง โดยที่แต่ละโหนดในชั้นนำเข้าจะแทนค่าของแต่ละคุณสมบัติของข้อมูลนำเข้า โดยแต่ละโหนดจะเป็นตัวแทนของค่าที่มีความหมาย เช่น ค่าพิกเซลในรูปภาพ

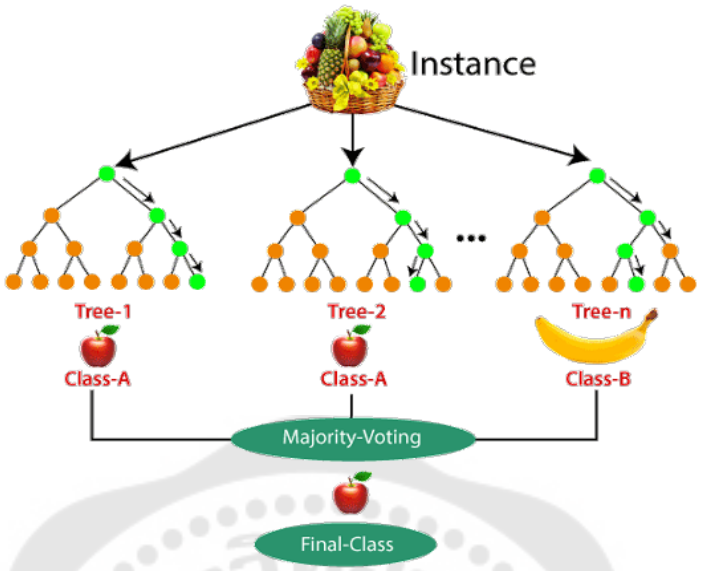
2. ชั้นซ่อน (hidden layer) เป็นส่วนที่มีหน้าที่ในการประมวลผลและเรียนรู้คุณลักษณะหลักของข้อมูล ซึ่งแต่ละโหนดในชั้นซ่อนจะประมวลผลข้อมูลนำเข้าด้วยการใช้ฟังก์ชันเชิงเส้นหรือฟังก์ชันทางคณิตศาสตร์เพื่อให้ได้ผลลัพธ์ที่เหมาะสม โดยชั้นซ่อนสามารถมีหลายชั้นได้ตามความซับซ้อนของแบบจำลอง และเป็นส่วนที่มีความสำคัญในการเรียนรู้คุณลักษณะที่ซับซ้อนของข้อมูล

3. ชั้นส่งออก (output layer) มีหน้าที่ในการส่งผลลัพธ์ที่ได้จากการประมวลผลข้อมูล โดยที่แต่ละโหนดในชั้นส่งออกจะแทนค่าผลลัพธ์ของแบบจำลอง ซึ่งอาจเป็นการแทนจำนวนหรือแทนค่าประเภทของข้อมูลตามงานที่กำหนด และชั้นส่งออกจะให้ผลลัพธ์ที่เป็นข้อมูลส่งออก (output data) ของแบบจำลองที่สามารถนำไปใช้ในงานได้ตามความต้องการของงานนั้น ๆ

ในการเรียนรู้ Neural Network ระบบจะใช้ข้อมูลที่มีค่าเป้าหมาย (target value) มาเปรียบเทียบกับผลลัพธ์ที่ได้จากชั้นส่งออก และใช้วิธีการย้อนกลับ (backpropagation) เพื่อปรับค่าพารามิเตอร์ภายในแบบจำลองเพื่อให้ผลลัพธ์ที่ได้มีความเข้ากันได้มากขึ้นกับค่าเป้าหมายที่ต้องการ

2.1.4.2 Random Forest

Random Forest เป็นแบบจำลองที่ใช้ในการเรียนรู้แบบมีการดูความสัมพันธ์ (supervised learning) โดยสร้างจากการรวมกันของต้นไม้ตัดสินใจหลายต้น (decision trees) ที่ถูกสร้างขึ้นจากการสุ่มตัวอย่างและคุณลักษณะ (features) จากชุดข้อมูลที่มีอยู่



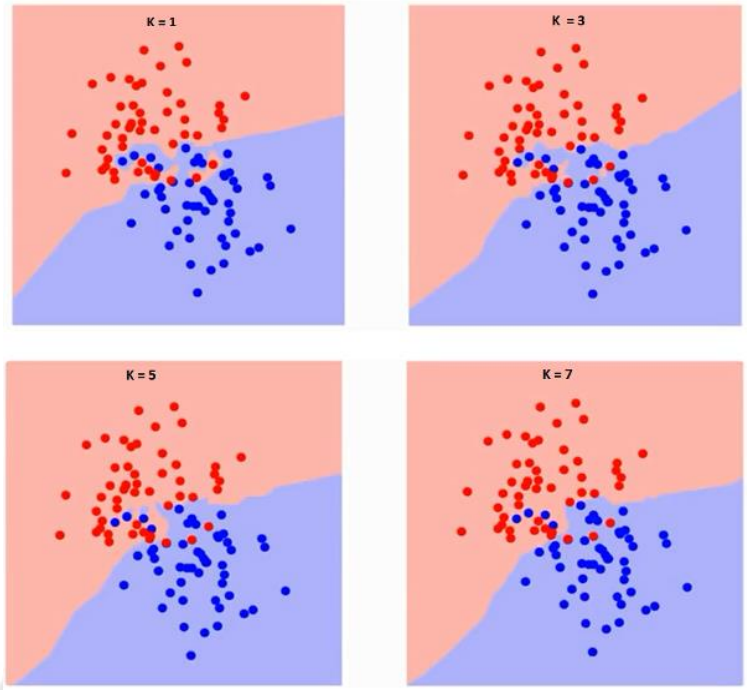
ภาพประกอบ 9 โครงสร้างของ Random Forest

ที่มา : (R, 2024)

โดยทั่วไป แต่ละต้นไม้ในแบบจำลอง Random Forest จะถูกสร้างขึ้นมาโดยใช้วิธีการสุ่มทั้งข้อมูลและคุณลักษณะของข้อมูล เพื่อลดความเสี่ยงของการเกิดการเรียนรู้ที่มากเกินไป (overfitting) ซึ่งเป็นสถานการณ์ที่แบบจำลองเข้าใจข้อมูลที่ใช้สำหรับการฝึกอบรมได้ดีเกินไปจนไม่สามารถทำนายข้อมูลใหม่ได้อย่างแม่นยำ เมื่อมีข้อมูลใหม่เข้ามา Random Forest จะทำการทำนายผลลัพธ์โดยให้แต่ละต้นไม้ในป่าสร้างการทำนายขึ้นมา แล้วจะใช้วิธีการโหวต (voting) สำหรับงานการจำแนกประเภท (classification) หรือสำหรับงานการทำนายค่าต่อเนื่อง (regression) เพื่อให้ได้ผลลัพธ์สุดท้าย

2.1.4.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) คือ แบบจำลองการเรียนรู้ของเครื่องที่ใช้สำหรับงานการจำแนกประเภท (classification) และการทำนายค่าต่อเนื่อง (regression) โดยพิจารณาจากค่า k ของข้อมูลที่ใกล้ที่สุดในข้อมูลฝึก



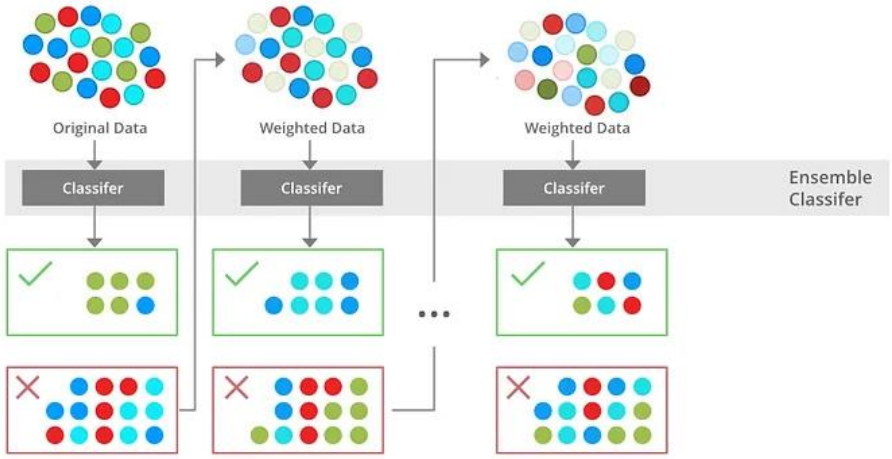
ภาพประกอบ 10 K-Nearest Neighbors (KNN)

ที่มา : (Srivastava, 2024)

การทำงานของแบบจำลองจะเลือกค่า k ที่ใกล้ที่สุดเพื่อพิจารณาผลของการทำนาย การเลือกค่า k ที่เหมาะสมเป็นสิ่งสำคัญเพราะถ้าค่า k ที่ต่ำเกินไปอาจนำไปสู่การทำนายที่มีความผันผวนสูง (high variance) ในขณะที่ค่า k ที่สูงเกินไปอาจทำให้ไม่สามารถจับรูปแบบในข้อมูลได้ (high bias)

2.1.4.4 XGBoost (Extreme Gradient Boosting)

XGBoost คือ แบบจำลองการเรียนรู้ของเครื่องที่ได้รับความนิยมสูง ซึ่งใช้งานได้ดีกับปัญหาการจำแนกประเภท (classification) และการทำนายค่าต่อเนื่อง (regression) แนวคิดหลักของแบบจำลอง XGBoost คือการใช้เทคนิคที่เรียกว่า Gradient Boosting ในการสร้างแบบจำลองซึ่งจะประกอบไปด้วยการรวมกันของหลาย ๆ ต้นไม้ตัดสินใจ (decision trees) โดยแต่ละต้นไม้จะพยายามแก้ไขข้อผิดพลาดที่เกิดขึ้นจากต้นไม้ก่อนหน้า เริ่มต้นจากต้นไม้ที่ง่ายแล้วค่อยเพิ่มความซับซ้อน เป็นการหาวิธีที่ดีที่สุดในการลดความผิดพลาดของการทำนายโดยการปรับค่าพารามิเตอร์ของต้นไม้ต้นต่อไปที่จะเพิ่มเข้ามา ซึ่งช่วยให้สามารถควบคุมการเกิดการเกิดการเรียนรู้ที่มากเกินไป (overfitting) และปรับปรุงประสิทธิภาพของแบบจำลองได้



ภาพประกอบ 11 XGBoost (Extreme Gradient Boosting)

ที่มา : (Verma, 2022)

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้มีการนำการเข้ารหัสแบบ Entity Embedding มาใช้ในการแปลงข้อมูลหมวดหมู่ที่มีค่าความหลากหลายสูง (high cardinality variables) ให้เป็นข้อมูลเชิงปริมาณ โดยการใช้การเข้ารหัสแบบ Entity Embedding ซึ่งไม่เพียงแต่ช่วยลดมิติของข้อมูลหมวดหมู่ที่มีค่าความหลากหลายสูง แต่ยังช่วยให้แบบจำลองสามารถเรียนรู้และทำความเข้าใจถึงความสัมพันธ์ที่ซับซ้อนในข้อมูลหมวดหมู่ได้ดีขึ้น ในงานวิจัยอื่นที่มีการใช้ Entity Embedding กับข้อมูลหมวดหมู่ที่มีค่าความหลากหลายสูง นักวิจัยมักใช้เทคนิคนี้เพื่อแปลงข้อมูลเหล่านั้นให้เข้ากับโครงสร้างของแบบจำลองการเรียนรู้ของเครื่อง ทำให้แบบจำลองสามารถเข้าถึงและใช้ประโยชน์จากข้อมูลได้อย่างมีประสิทธิภาพ ซึ่งการศึกษาเหล่านี้มักเน้นที่การสำรวจและทดลองว่าการเข้ารหัสด้วยวิธีนี้สามารถช่วยปรับปรุงความแม่นยำและประสิทธิภาพของแบบจำลองในงานที่เกี่ยวข้องกับข้อมูลประเภทหมวดหมู่ได้อย่างไร

สำหรับการนำ Entity Embedding ไปใช้กับข้อมูลสินค้าของศุลกากรของประเทศมาเลเซียเพื่อการตรวจจับการฉ้อโกงทางศุลกากร (Hooi et al., 2022) ในบทความวิจัยเรื่อง Feature Encoding For High Cardinality Categorical Variables Using Entity Embeddings : A Case Study in Customs Fraud Detection ได้ทำการศึกษาการตรวจจับการฉ้อโกงทางศุลกากร โดยการแปลงตัวแปรประเภทหมวดหมู่ที่มีค่าความหลากหลายสูงโดยใช้การเข้ารหัสแบบ Entity Embedding เพื่อลดมิติของข้อมูล ในงานวิจัยนี้ได้อธิบายถึงความท้าทายที่หน่วยงานศุลกากรต้องเผชิญ นั่นคือความท้าทายจากการเพิ่มขึ้นของระดับการค้าระหว่างประเทศและทรัพยากรที่ไม่

เพียงพอในการตรวจสอบ รัฐบาลจึงตัดสินใจใช้ทรัพยากรที่จำกัดเหล่านี้เพื่อมุ่งไปที่การตรวจจับการค้าที่น่าสงสัย ด้วยการใช้ชุดข้อมูลที่ถูกจัดเก็บไว้เพื่อตรวจจับการฉ้อโกงทางศุลกากร และจากการวิเคราะห์ชุดข้อมูล ปัญหาหลักที่พบคือปัญหาของค่าความหลากหลายสูง (high cardinality) จึงมีการนำการเข้ารหัสแบบ Entity Embedding มาใช้ในการแก้ไขปัญหานี้

Field Name	le	ohe	de	se	he	be	te	sr	woe	ee
k1_impcd	1	294	293	293	1	9	1	1	1	4
k1_expname	1	509	508	508	1	9	1	1	1	5
k1_agent	1	186	185	185	1	8	1	1	1	4
k1_contno	1	3388	3387	3387	1	12	1	1	1	8
k1_regoff	1	80	79	79	1	7	1	1	1	3
k1_pexoff	1	110	109	109	1	7	1	1	1	3
k1_aprvoff	1	113	112	112	1	7	1	1	1	3
k1_reloff	1	17	16	16	1	5	1	1	1	2

ภาพประกอบ 12 จำนวนฟีเจอร์หลังจากใช้วิธีการเข้ารหัส (encoding method) ที่แตกต่างกัน

ที่มา : (Hooi et al., 2022)

จากภาพประกอบ 13 เมื่อแสดงผลลัพธ์จากการวัดประสิทธิภาพของแบบจำลองด้วยวิธีการเข้ารหัสแบบ Entity Embedding ได้ค่า AUC-ROC เท่ากับ 85.44% และค่า F1-Score เท่ากับ 84.54% ซึ่งมีประสิทธิภาพที่สูงที่สุดเมื่อเทียบกับการเข้ารหัสด้วยวิธีอื่น

Encoding	Accuracy	Precision	Recall	AUC-ROC	F1-Score	Time
le	99.17%	3.70%	2.63%	51.14%	51.33%	2.16
ohe	**99.70%**	80.00%	52.63%	76.28%	81.67%	86.79
de	99.66%	71.43%	52.63%	76.26%	80.22%	22.76
se	99.69%	73.33%	57.89%	78.89%	82.27%	104.91
he	99.57%	**100.00%**	13.16%	56.58%	61.52%	6.74
be	99.67%	68.57%	63.16%	81.51%	82.79%	4.88
te	99.67%	74.07%	52.63%	76.27%	80.69%	1.51
sr	99.67%	74.07%	52.63%	76.27%	80.69%	**1.34**
woe	99.71%	80.77%	55.26%	77.60%	82.74%	3.63
ee	99.69%	67.50%	**71.05%**	**85.44%**	**84.54%**	1.54

ภาพประกอบ 13 ผลการทดสอบประสิทธิภาพของแบบจำลอง Neural Network เมื่อผ่านการเข้ารหัสต่างวิธี

ที่มา : (Hooi et al., 2022)

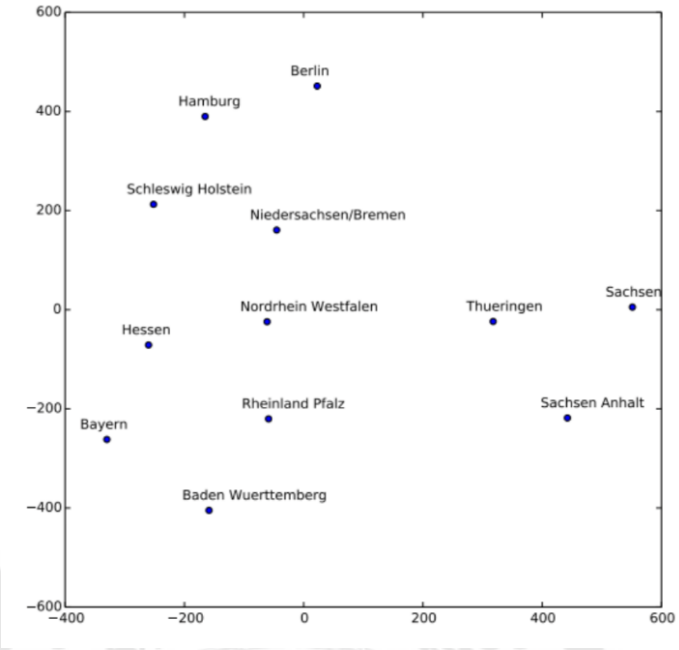
สำหรับการนำ Entity Embedding ไปใช้ในชุดข้อมูลจากการแข่งขัน Kaggle Rossmann Sale Prediction (Guo & Berkhahn, 2016) ยังได้ศึกษาการเข้ารหัสด้วยวิธีเดียวกันโดยผ่านบทความวิจัยเรื่อง Entity Embeddings of Categorical Variables เพื่อประเมินประสิทธิภาพของการเข้ารหัสแบบ Entity Embedding โดยในข้อมูลชุดนี้มีการระบุขนาดของเวกเตอร์ที่ใช้ในการฝังข้อมูล (embedding) สำหรับแต่ละลักษณะของข้อมูล การฝังข้อมูลเป็นการแปลงข้อมูลจากค่าที่เป็นหมวดหมู่หรือข้อมูลที่เป็นลำดับไปเป็นเวกเตอร์ในพื้นที่ต่อเนื่องให้มีมิติที่ต้องการ ซึ่งช่วยให้สามารถนำข้อมูลที่เป็นหมวดหมู่หรือลำดับไปใช้ในแบบจำลองที่ต้องการข้อมูลเป็นรูปแบบตัวเลข

feature	data type	number of values	EE dimension
store	nominal	1115	10
day of week	ordinal	7	6
day	ordinal	31	10
month	ordinal	12	6
year	ordinal	3 (2013-2015)	2
promotion	binary	2	1
state	nominal	12	6

ภาพประกอบ 14 จำนวนฟีเจอร์จากชุดข้อมูล Kaggle Rossmann ก่อนและหลังการทำ Entity Embedding

ที่มา : (Guo & Berkhahn, 2016)

โดยในงานวิจัยนี้มีการเปรียบเทียบประสิทธิภาพของแบบจำลอง คือ K-nearest neighbors (KNN), Random Forests, Gradient Boosted Trees และ Neural Network โดยวัดประสิทธิภาพของแบบจำลองโดยใช้ค่า MAPE พบว่า K-nearest neighbors (KNN) มีค่า MAPE เท่ากับ 0.099 Random Forests มีค่า MAPE เท่ากับ 0.089 Gradient Boosted Trees มีค่า MAPE เท่ากับ 0.071 และ Neural Network มีค่า MAPE เท่ากับ 0.070 ซึ่ง KNN ให้ผลลัพธ์ที่ดีกว่าเมื่อเทียบกับแบบจำลองประเภทอื่นในการคาดการณ์ยอดขาย

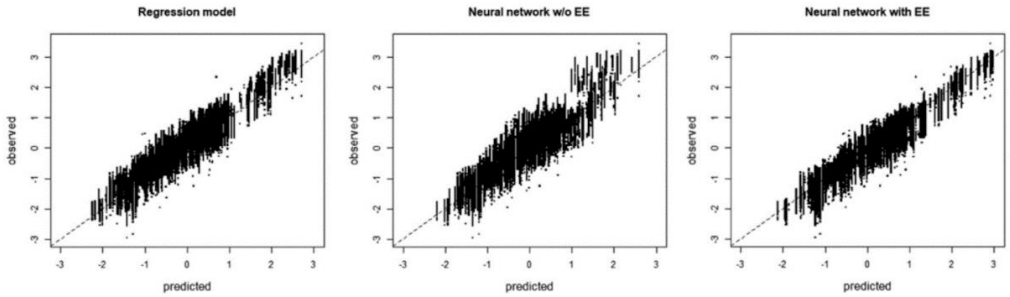


ภาพประกอบ 15 ผลลัพธ์ของการทำ embedded feature โดยใช้เทคนิคการลดมิติของข้อมูล (t-SNE)

ที่มา : (Guo & Berkhahn, 2016)

จากภาพประกอบ 15 แสดงผลลัพธ์ของการทำ embedded feature เพื่อแสดงข้อมูลให้อยู่บนพื้นที่ 2 มิติ โดยใช้เทคนิคการลดมิติของข้อมูล (t-SNE) จากงานวิจัยฉบับนี้แม้ว่าแบบจำลองจะไม่ทราบอะไรที่เกี่ยวกับภูมิศาสตร์ของประเทศเยอรมันแต่ตำแหน่งสัมพันธ์ของการทำ embedded feature ที่เรียนรู้เกี่ยวกับรัฐเยอรมันให้ผลลัพธ์คล้ายกับแผนที่ประเทศเยอรมันอย่างน่าแปลกใจ เหตุผลคือการทำ embedded feature จะรวมเอารัฐที่มีการกระจายที่คล้ายกันของลักษณะ เช่น เศรษฐกิจและสภาพวัฒนธรรมที่ใกล้เคียงกันให้มีระยะห่างใกล้เคียงกัน ในเวลาเดียวกันรัฐที่อยู่ใกล้กันทางภูมิศาสตร์ก็มีโอกาสที่จะมีเศรษฐกิจและวัฒนธรรมที่คล้ายกัน โดยเฉพาะรัฐที่อยู่ในกลุ่มด้านขวา คือ Sachsen, Thueringen และ Sachsen Anhalt ทั้งหมดมาจากเยอรมันตะวันออกในขณะที่รัฐในกลุ่มด้านซ้ายมาจากเยอรมันตะวันตก นี่จึงเป็นคุณลักษณะเด่นของการทำ embedded feature ว่าสามารถนำมาใช้ในการจัดกลุ่มข้อมูลแบบหมวดหมู่ได้อย่างมีประสิทธิภาพและเห็นความสัมพันธ์ของข้อมูล

ในการนำเครือข่ายประสาทเทียม (neural network) ประยุกต์ใช้ควบคู่กับการเข้ารหัสแบบ Entity Embedding เพื่อประเมินมูลค่าผลิตภัณฑ์ในกรณีศึกษาด้านอุตสาหกรรมยานยนต์ (Lee, 2023) ในงานวิจัยเรื่อง How can we use neural network with entity embedding for product valuations? A case study for the car industry ได้นำแบบจำลอง Neural Network มาใช้กับชุดข้อมูลเกี่ยวกับการจดทะเบียนรถยนต์เพื่อประเมินราคา และพยายามปรับปรุงประสิทธิภาพของแบบจำลองโดยใช้การเข้ารหัสแบบ Entity Embedding ในตัวแปรประเภทหมวดหมู่



ภาพประกอบ 16 การทดสอบประสิทธิภาพของแบบจำลองในชุดข้อมูลทดสอบ

ที่มา : (Lee, 2023)

ในงานวิจัยนี้ผู้วิจัยได้การประเมินผลประสิทธิภาพของแบบจำลองออกเป็น 3 ประเภท ได้แก่

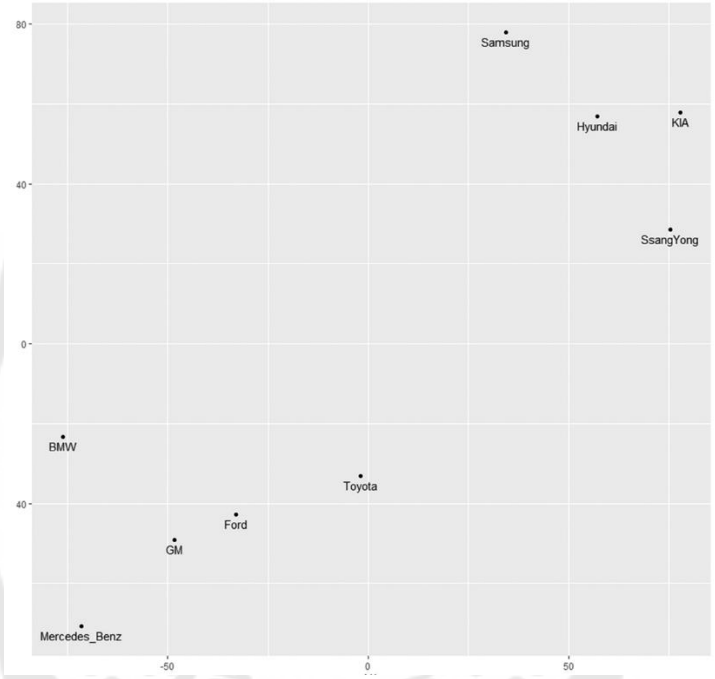
1. แบบจำลองเชิงถดถอย (regression model)
2. เครือข่ายที่ไม่มีการเข้ารหัสแบบ Entity Embedding (network without entity embedding)
3. เครือข่ายที่มีการเข้ารหัสแบบ Entity Embedding (network with entity embedding) ซึ่งผลลัพธ์ของการทดสอบประสิทธิภาพของแบบจำลอง ดังแสดงในตาราง 3

ตาราง 3 ผลลัพธ์ของการทดสอบประสิทธิภาพของแบบจำลอง

	Regression	Network without entity embedding	Network with entity embedding
No. of parameters	203	51,901	2,659,437
RMSE	0.32-0.34	0.33-0.38	0.18-0.19
MAPE	12.5-13.8	16.1-17.7	6.6-7.1

ที่มา : (Lee, 2023)

จากข้อมูลในตาราง 3 สังเกตได้ว่าเครือข่ายที่มีการเข้ารหัสแบบ Entity Embedding สามารถทำนายผลได้ดีที่สุด โดยมีค่า RMSE และ MAPE ที่ต่ำที่สุด ซึ่งบ่งชี้ถึงความแม่นยำที่สูงกว่าเมื่อเทียบกับเครือข่ายที่ไม่มีการฝังข้อมูลเอนทิตีและแบบจำลองเชิงถดถอย นอกจากนี้ยังสามารถใช้เทคนิคการลดมิติของข้อมูล (t-SNE) เพื่อลดขนาดของ Embedded Feature ลงมาให้สามารถแสดงผลในรูปแบบการแสดงภาพ (visualization) ได้โดยที่ยังสามารถรักษาความสัมพันธ์ของข้อมูลในการแสดงผลได้ดี



ภาพประกอบ 17 การแสดงภาพผู้ผลิตรถยนต์ที่ใช้เทคนิคการลดมิติของข้อมูล (t-SNE)

ที่มา : (Lee, 2023)

จากภาพประกอบ 17 เมื่อมีการลดมิติของข้อมูลด้วยเทคนิคการใช้ t-SNE จะพบว่ารถยนต์ที่มีผู้ผลิตที่อยู่ในภูมิภาคหรือประเทศคล้ายกันจะรวมกลุ่มอยู่ในพื้นที่ใกล้กัน ซึ่งการแสดงผลในลักษณะนี้จะช่วยให้เข้าใจโครงสร้างข้อมูลได้อย่างลึกซึ้ง และช่วยในการทำนายหรือการจัดกลุ่มข้อมูลได้ง่ายและมีประสิทธิภาพมากขึ้น ฉะนั้นแล้วตัวแปรหมวดหมู่จึงเป็นชนิดของข้อมูลที่สำคัญสำหรับการประเมินราคารถยนต์ โดยทั่วไปมักจะใช้การเข้ารหัสแบบ One-Hot Encoding ในการแปลงข้อมูลให้เป็นตัวเลข แต่ในงานวิจัยนี้ได้เลือกใช้การเข้ารหัสแบบ Entity Embedding เพื่อเพิ่มประสิทธิภาพให้กับแบบจำลอง

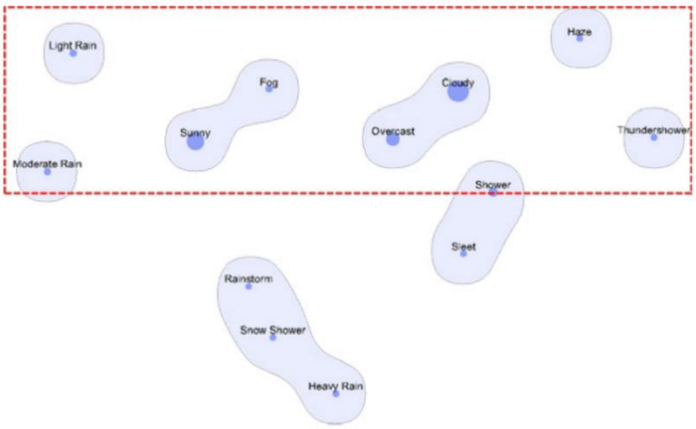
สำหรับการใช้การเข้ารหัสด้วยวิธี Entity Encoding ในชุดข้อมูลการจราจรของระบบจักรยานสาธารณะที่ใช้ร่วมกันในเมืองซูโจว ประเทศจีน เพื่อคาดการณ์ปริมาณการจราจร (Wang et al., 2019) ในบทความวิจัยเรื่อง Reveal the hidden layer via entity embedding in traffic prediction ได้มีการพิจารณาตัวแปรหลายประเภท ได้แก่ เวลา เทศกาล สภาพอากาศ และข้อมูลทั่วไปของพื้นที่นั้น ๆ และนำมาประยุกต์ใช้กับแบบจำลอง Neural Network โดยใช้ค่า MSE วัดประสิทธิภาพของแบบจำลอง ซึ่งคณะผู้วิจัยได้แบ่งข้อมูลออกเป็น 2 ชุด โดยแบ่งข้อมูลนำเข้าให้มีขนาดที่ต่างกัน ได้แก่ สถานี 100 อันดับแรก (Top 100 sites) ที่มีผู้ใช้บริการมากที่สุดในเมืองซูโจว และสถานีทั้งหมด 1,786 แห่ง (All 1,768 sites) ในเมืองซูโจว จากการทดสอบประสิทธิภาพของแบบจำลองได้ผลลัพธ์ ดังตาราง 4

ตาราง 4 ผลลัพธ์แสดงความแตกต่างของการแบ่งชุดข้อมูล

Input Size	Best Validation MSE	Traffic Flow (Usage) / site / day
Top 100 sites	31.768	403.701
All 1,786 sites	6.518	112.278

ที่มา : (Wang et al., 2019)

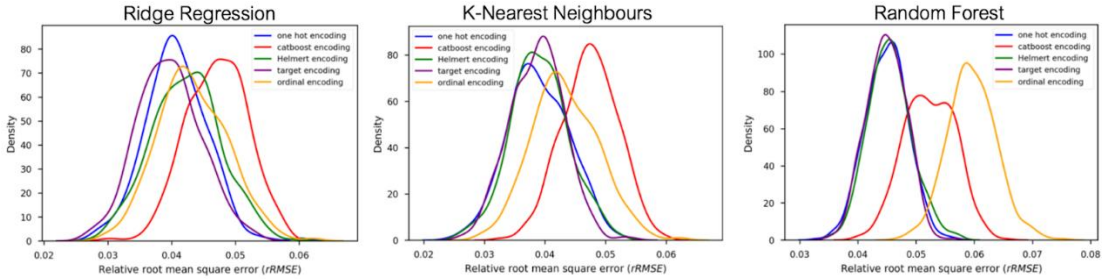
จากตาราง 4 พบว่า สถานี 100 อันดับแรก ที่มีผู้ใช้บริการมากที่สุดในเมืองซูโจว ได้ค่า MSE เท่ากับ 31.768 ส่วนสถานีทั้งหมด 1,786 แห่ง ในเมืองซูโจว ได้ค่า MSE เท่ากับ 6.518 ซึ่งแสดงให้เห็นว่าแบบจำลองที่มีข้อมูลนำเข้าขนาดใหญ่มีประสิทธิภาพมากกว่าแบบจำลองที่มีข้อมูลนำเข้าขนาดเล็ก และเนื่องจากในงานวิจัยฉบับนี้มีการใช้การเข้ารหัสแบบ Entity Embedding จึงสามารถมองเห็นความสัมพันธ์ของตัวแปรโดยใช้เทคนิคการลดมิติของข้อมูลด้วยการใช้ t-SNE ในตัวแปรสภาพอากาศ พบว่าในวันที่มีแดดหรือมีเมฆมากเป็นสภาพอากาศที่เหมาะสำหรับการปั่นจักรยาน ค่อนข้างเหมาะสำหรับการปั่นจักรยาน ส่วนสภาพอากาศอื่น เช่น มีฝนเล็กน้อยหรือมีหมอก จะเห็นว่ารวมกลุ่มกันอยู่ห่างจากวันที่มีสภาพอากาศที่มีแดดหรือมีเมฆมาก ดังภาพประกอบ 18



ภาพประกอบ 18 การแสดงภาพ 2 มิติ ในตัวแปรสภาพอากาศ

ที่มา : (Wang et al., 2019)

การใช้ Entity Embedding สำหรับการประเมินมูลค่าทรัพย์สิน (Gnat, 2021) ในบทความวิจัยเรื่อง Impact of Categorical Variables Encoding on Property Mass Valuation ได้นำเสนอผลกระทบของการเข้ารหัสตัวแปรประเภทหมวดหมู่ต่อการประเมินมูลค่าทรัพย์สิน ซึ่งต้องมีการเข้ารหัสตัวแปรอย่างเหมาะสมเพื่อให้ได้ผลลัพธ์ที่แม่นยำที่สุด งานวิจัยนี้จึงได้ทดสอบการเข้ารหัสตัวแปรด้วยเทคนิคต่าง ๆ ในขั้นตอนการเตรียมข้อมูลเพื่อประเมินว่าการเลือกวิธีการเข้ารหัสมีผลต่อผลลัพธ์ของการประเมินมูลค่าหรือไม่โดยใช้แบบจำลอง ได้แก่ Ridge Regression, K-nearest Neighbours Regression และ Random Forest Regression โดยแต่ละแบบจำลองมีตัวแปรอธิบายหมวดหมู่ที่ใช้การเข้ารหัส ได้แก่ One-hot Encoding, Catboost Encoding, Helmert Encoding, Target Encoding และ Ordinal Encoding ผลลัพธ์พบว่าผลการประเมินมูลค่าทรัพย์สินแบบจำนวนมากมีความแตกต่างกันไปตามวิธีการเข้ารหัสตัวแปรหมวดหมู่ และแต่ละแบบจำลองที่ใช้ในการวิจัยนี้ตอบสนองต่อเทคนิคการเข้ารหัสได้แตกต่างกัน



ภาพประกอบ 19 Kernel density estimation of relational root mean square errors

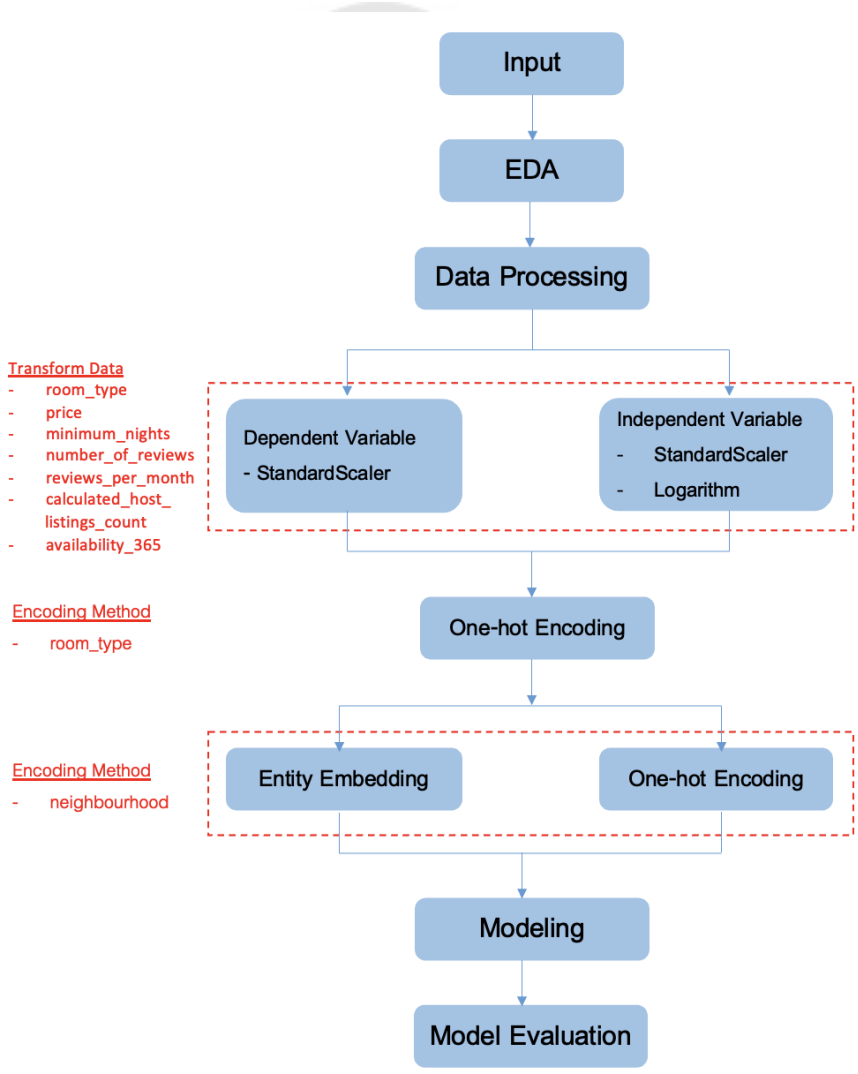
ที่มา : (Gnat, 2021)

จากภาพประกอบ 19 แสดงให้เห็นถึงผลลัพธ์ของประสิทธิภาพของการเข้ารหัสแบบต่าง ๆ ในแบบจำลองที่ต่างกัน ซึ่งพบว่าการเข้ารหัสแบบ One-hot Encoding ถูกพิสูจน์แล้วว่าเป็นตัวเลือกที่ดีที่สุด

จากงานวิจัยที่ยกตัวอย่างมานั้นได้พูดถึงการเข้ารหัสแบบ Entity embedding ว่าเป็นเทคนิคที่ใช้สำหรับการแปลงข้อมูลที่มีค่าความหลากหลายสูงให้กลายเป็นเวกเตอร์ที่มีมิติต่ำลง ทำให้เหมาะสมที่จะนำไปใช้ในการทดสอบประสิทธิภาพของแบบจำลองให้มีประสิทธิภาพมากขึ้น นอกจากนี้ยังสามารถใช้ Embedded Feature ที่ได้มามาทำการแสดงข้อมูลในรูปแบบต่าง ๆ โดยใช้เทคนิคการลดมิติของข้อมูล เช่น t-SNE (t-Distributed Stochastic Neighbor Embedding) หรือ (PCA : Principal Component Analysis) ส่วน One-hot encoding นั้นเป็นเทคนิคที่ใช้ในการแปลงข้อมูลที่มีลักษณะเป็นข้อความหรือหมวดหมู่เป็นเวกเตอร์ที่ประกอบด้วยค่า 0 และ 1 เพื่อให้สามารถนำเข้าได้กับแบบจำลองที่รับข้อมูลเป็นตัวเลขได้ โดยในเวกเตอร์นี้มีค่า 1 อยู่ในตำแหน่งที่สอดคล้องกับหมวดหมู่ของข้อมูลที่เป็นไปได้ และค่า 0 อยู่ที่ตำแหน่งที่ไม่สอดคล้องกับหมวดหมู่ที่เป็นไปได้ ซึ่งเหมาะสำหรับใช้กับแบบจำลองที่ไม่สามารถจัดการกับข้อมูลที่มีลักษณะเป็นข้อความหรือหมวดหมู่โดยตรง เช่น Decision Trees, Support Vector Machines (SVM), Neural Networks หรือแบบจำลองที่ใช้งานได้กับข้อมูลที่เป็นตัวเลขเท่านั้น เช่น Logistic Regression และ Linear Regression โดยจำเป็นต้องพิจารณาถึงความเหมาะสมของข้อมูลที่จะนำมาใช้ในการเข้ารหัสแบบต่าง ๆ เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดจากแบบจำลองที่เลือกใช้การเข้ารหัสนั้น ๆ

บทที่ 3
แนวคิดและวิธีวิจัย

งานวิจัยนี้มุ่งเน้นการค้นหาเทคนิคการเข้ารหัส (encoding method) ที่มีประสิทธิภาพเพื่อใช้สำหรับการทำนายราคาที่พักบน Airbnb โดยเทคนิคที่เลือกใช้ในงานวิจัยนี้ ได้แก่ Entity Embedding และ One-hot Encoding และนำมาทดสอบประสิทธิภาพโดยประยุกต์ใช้กับแบบจำลองต่างชนิด ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost



ภาพประกอบ 20 กระบวนการสร้างแบบจำลอง

กระบวนการสร้างแบบจำลองเริ่มจากการนำเข้าข้อมูล (input) และการสำรวจข้อมูล (Exploratory Data Analysis : EDA) ซึ่งเป็นการวิเคราะห์ข้อมูลเบื้องต้นเพื่อทำความเข้าใจข้อมูลและหาความสัมพันธ์ระหว่างตัวแปรต่าง ๆ ต่อมาทำการจัดการข้อมูล (data processing) โดยมีการจัดการกับข้อมูลที่หายไป (missing data) ข้อมูลที่มีรายการซ้ำ (duplicate data) และข้อมูลส่วนเกิน (outlier) หลังจากนั้นจะเริ่มกระบวนการปรับปรุงข้อมูล (transform data) เพื่อแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม โดยในตัวแปรตาม (dependent variable) มีการปรับปรุงข้อมูลด้วยวิธี StandardScaler กับคอลัมน์ที่เป็นตัวแปรเชิงปริมาณทั้งหมด และในตัวแปรอิสระ (independent variable) มีการปรับการกระจายตัวของข้อมูล 2 วิธี คือ StandardScaler และ Logarithm ซึ่งหลังจากที่ข้อมูลอยู่ในรูปแบบการกระจายตัวที่เหมาะสมแล้วจะถูกนำไปเข้ารหัสแบบ Entity Embedding และ One-hot Encoding ในตัวแปรที่มีค่าความหลากหลายสูง (high cardinality) นั่นคือตัวแปรในคอลัมน์ neighbourhood ถัดมาจะเป็นขั้นตอนการสร้างแบบจำลอง (modeling) เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองชนิดต่าง ๆ ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost และสุดท้ายเพื่อทดสอบและประเมินประสิทธิภาพของแบบจำลองผ่านข้อมูลทดสอบ (test data) โดยพิจารณาค่าความคลาดเคลื่อนด้วยค่า Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) และ R-squared

3.1 **การเก็บรวบรวมข้อมูล** (Data Collection)

ชุดข้อมูลที่ใช้ในงานวิจัยฉบับนี้เป็นข้อมูลสาธารณะจากเว็บไซต์ http://insideairbnb.com/get-the-data/ ที่เก็บรวบรวมข้อมูลต่าง ๆ เกี่ยวกับที่พักบน Airbnb ในหลาย ๆ เมือง หลาย ๆ ประเทศทั่วโลก เช่น พื้นที่ ประเภทที่พัก จำนวนคืนที่เข้าพัก คะแนนรีวิว โดยผู้วิจัยได้เลือกชุดข้อมูลของ Airbnb ในพื้นที่กรุงเทพมหานคร ประกอบด้วย 20,823 แถว 18 คอลัมน์

ตาราง 5 แสดงคอลัมน์ของข้อมูล

ลำดับ	ชื่อแอททริบิวต์	ประเภทของข้อมูล	คำอธิบาย
1	id	integer	หมายเลยเลขประจำตัวที่ไม่ซ้ำในรายการ
2	name	text	ชื่อที่พัก
3	host_id	integer	หมายเลขเลขประจำตัวที่ไม่ซ้ำสำหรับเจ้าของที่พักหรือผู้ใช้
4	host_name	text	ชื่อของเจ้าของที่พัก
5	neighbourhood_group	text	กลุ่มย่านที่ตั้งของที่พัก
6	neighborhood	text	ย่านที่พัก
7	latitude	numeric	ละติจูดตามพิกัดทางภูมิศาสตร์
8	longitude	numeric	ลองติจูดตามพิกัดทางภูมิศาสตร์
9	room_type	text	ประเภทของที่พัก
10	price	currency	ราคาของที่พัก
11	minimum_nights	integer	จำนวนคืนขั้นต่ำสำหรับการเข้าพัก
12	number_of_reviews	numeric	จำนวนรีวิวทั้งหมด
13	last_review	date	วันที่ล่าสุดที่ผู้เข้าพักรีวิวให้กับที่พัก
14	reviews_per_month	numeric	จำนวนรีวิวเฉลี่ยต่อเดือน
15	calculated_host_listing_count	integer	จำนวนรายการที่เจ้าของที่พักมีทั้งหมด
16	availability_365	integer	ความพร้อมในการจองสำหรับ 365 วันถัดไป
17	number_of_reviews_ltm	integer	จำนวนรีวิวใน 12 เดือนที่ผ่านมา
18	license	text	หมายเลขใบอนุญาต

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20823 entries, 0 to 20822
Data columns (total 8 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   neighbourhood                   20823 non-null  object
 1   room_type                       20823 non-null  object
 2   price                           20823 non-null  int64
 3   minimum_nights                  20823 non-null  int64
 4   number_of_reviews               20823 non-null  int64
 5   reviews_per_month               13360 non-null  float64
 6   calculated_host_listings_count  20823 non-null  int64
 7   availability_365                20823 non-null  int64
dtypes: float64(1), int64(5), object(2)
memory usage: 1.3+ MB
```

ภาพประกอบ 21 การสำรวจข้อมูลเบื้องต้น

จากภาพประกอบ 21 จะเห็นว่าข้อมูลของ neighbourhood และ room_type เป็นข้อมูลประเภทหมวดหมู่ (categorical variables) และข้อมูลใน reviews_per_month มีข้อมูลที่หายไปอยู่เป็นจำนวนมาก ซึ่งมีความจำเป็นต้องหาข้อมูลมาเติมเต็มในส่วนที่ขาดหายไป

3.2 การจัดการข้อมูล (Data Processing)

3.2.1 การจัดการกับข้อมูลที่หายไป (missing data) และข้อมูลที่มีรายการซ้ำ (duplicate data)

จากการสำรวจข้อมูลพบข้อมูลที่หายไปในคอลัมน์ reviews_per_month จำนวน 6,596 รายการ สำหรับค่าที่หายไปผู้วิจัยได้ตั้งสมมติฐานว่าอาจจะเกิดขึ้นเนื่องจากไม่มีรีวิวสำหรับรายการนั้น ๆ และเพื่อยืนยันสมมติฐานได้มีการกรองข้อมูลใน DataFrame และเก็บไว้เฉพาะรายการที่มีค่าที่หายไปและนำมาตรวจสอบกับคอลัมน์ number_of_reviews พบว่ามีค่าเป็น 0 นั่นแสดงว่ารายการทั้งหมดที่มีข้อมูลที่หายไปไม่มีการรีวิวจากผู้เข้าพัก ดังนั้นจึงต้องเติมข้อมูลที่หายไปให้มีค่าเท่ากับ 0 ส่วนข้อมูลที่มีการรายการซ้ำใน DataFrame พบว่ามีจำนวน 925 รายการ จึงได้ทำการลบข้อมูลที่มีรายการซ้ำทั้งหมดออกไป ดังนั้นในชุดข้อมูลจะมีรายการเหลือเพียง 19,898 รายการ 8 คอลัมน์

```
<class 'pandas.core.frame.DataFrame'>
Index: 19898 entries, 0 to 20822
Data columns (total 8 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   neighbourhood                   19898 non-null  object
 1   room_type                       19898 non-null  object
 2   price                           19898 non-null  int64
 3   minimum_nights                  19898 non-null  int64
 4   number_of_reviews               19898 non-null  int64
 5   reviews_per_month               13302 non-null  float64
 6   calculated_host_listings_count  19898 non-null  int64
 7   availability_365                19898 non-null  int64
dtypes: float64(1), int64(5), object(2)
memory usage: 1.4+ MB
```

ภาพประกอบ 22 ข้อมูลหลังจากจัดการกับข้อมูลที่หายไปและข้อมูลที่มีรายการซ้ำ

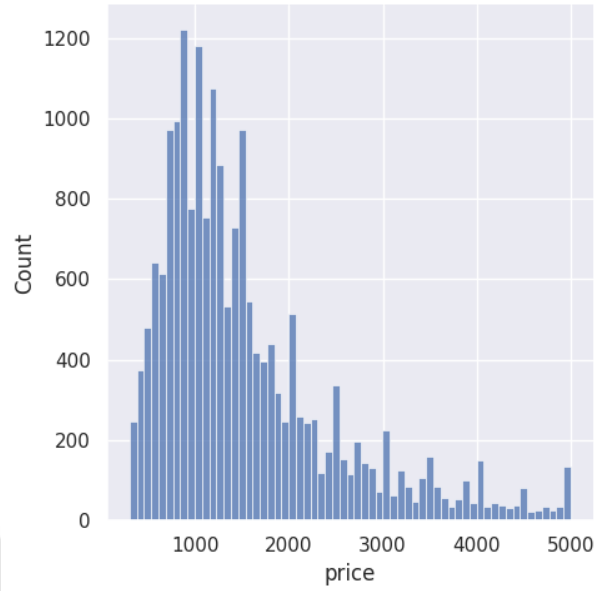
3.2.2 **การจัดการกับข้อมูลส่วนเกิน** (outliers)

จากการสำรวจข้อมูลในคอลัมน์ price พบว่าช่วงราคาของที่พักอยู่ระหว่าง 40-1,000,000 บาท มีดังนี้

ตาราง 6 ตารางแจกแจงความถี่ของคอลัมน์ price

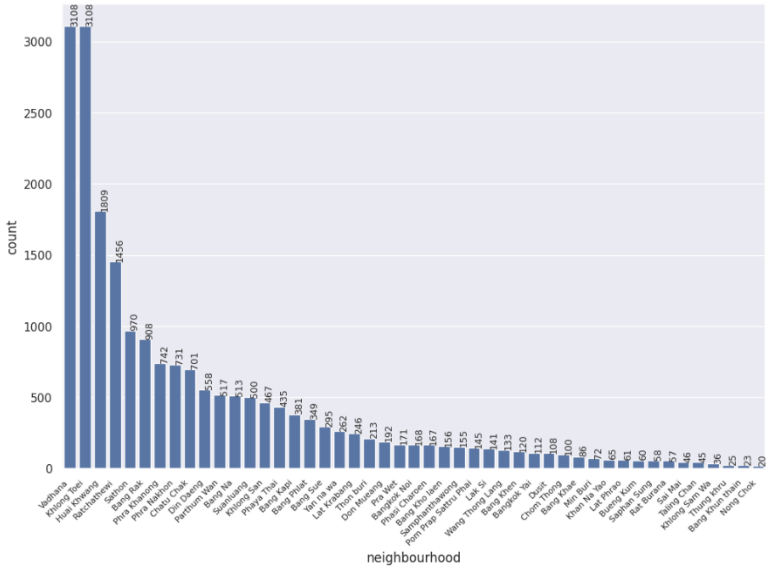
Price Range	Frequency
40 - 299	18
300 - 5,000	18,598
5,001 - 10,000	872
10,001 - 100,000	388
100,001 - 500,000	17
500,001 - 1,000,000	5

จากตาราง 6 จะเห็นว่าข้อมูลราคามีการกระจายตัวอยู่ในช่วงกว้างมากและหากมีการเก็บค่าทุกค่าในคอลัมน์ price ไว้จะทำให้แบบจำลองที่ได้มีจำนวนของข้อมูลส่วนเกินอยู่เป็นจำนวนมาก ดังนั้นในงานวิจัยนี้ จึงได้มีตัดข้อมูลบางส่วนที่ต่ำและสูงเกินความเป็นจริงออกโดยเลือกเก็บไว้เฉพาะช่วงราคาของที่พักที่อยู่ระหว่าง 300-5,000 บาทเท่านั้น ทำให้มีจำนวนคงเหลือของรายการทั้งหมด 18,598 รายการ



ภาพประกอบ 23 แสดงการกระจายตัวของคอลัมน์ price ที่อยู่ในช่วง 300-5,000 บาท

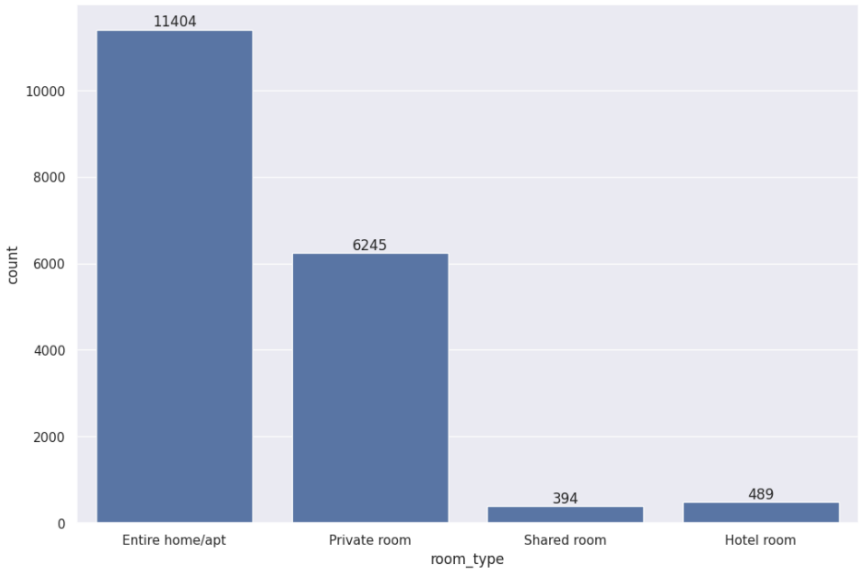
ถัดมาเป็นการสำรวจข้อมูลในคอลัมน์ neighbourhood พบว่ามีค่าที่ไม่ซ้ำกันในคอลัมน์ (unique values) จำนวน 50 ย่าน ซึ่งมีบางย่านที่มีค่าที่น้อยเกินไปที่อาจทำให้เกิดเป็นข้อมูลส่วนเกินได้ จึงทำการลบย่านที่มีค่าที่นับได้น้อยว่า 20 ออกไปจำนวน 3 ย่าน ได้แก่ หนองจอก หนองแขม และบางบอน เป็นจำนวนรวมทั้งหมด 66 รายการ และพบว่าค่าที่ไม่ซ้ำกันในคอลัมน์ neighbourhood มีจำนวนคงเหลือ 47 ย่าน ทำให้มีจำนวนคงเหลือของรายการทั้งหมด 18,532 รายการ



ภาพประกอบ 24 แสดงจำนวนรายการในแต่ละ neighbourhood

3.2.3 การจัดการกับตัวแปรประเภทหมวดหมู่ (categorical variables)

ข้อมูลประเภทหมวดหมู่ในคอลัมน์ room_type เก็บค่าของข้อมูลไว้ 4 ประเภท ได้แก่ Private room, Entire home/apt, Shared room และ Hotel room



ภาพประกอบ 25 แสดงจำนวนรายการในแต่ละ room_type

ในขั้นตอนนี้มีการเปลี่ยนตัวแปรหมวดหมู่ให้เป็นตัวแปรเชิงปริมาณของข้อมูลในคอลัมน์ room_type โดยการแทนข้อมูลในรูปแบบของตัวเลข (dummy variables) โดยใช้ตัวเลข 0 และ 1 หลังจากสร้างตัวแปรเชิงปริมาณทำให้มีจำนวนคงเหลือของรายการทั้งหมด 18,532 รายการ 11 คอลัมน์ ส่วนคอลัมน์ neighbourhood เป็นตัวแปรที่มีค่าความหลากหลายสูงซึ่งมีจำนวน 47 ค่าที่ไม่ซ้ำกัน ในขั้นตอนถัดไปผู้วิจัยจะได้นำเทคนิคการเข้ารหัสแบบ Entity Embedding และ One-hot Encoding มาใช้ในการจัดการกับคอลัมน์นี้

3.2.4 การเข้ารหัสแบบ Entity Embedding และ One-hot Encoding กับตัวแปรที่มีค่าความหลากหลายสูง (High Cardinality)

จากการสำรวจข้อมูลพบว่าคอลัมน์ neighbourhood มีจำนวนค่าที่ไม่ซ้ำกันในคอลัมน์ 47 รายการ จึงต้องมีการจัดการกับตัวแปรนี้ให้เป็นตัวเลขที่แสดงความหมายของค่า neighborhood นั้น ๆ ก่อนที่จะนำไปใช้ในแบบจำลองเพื่อวัดประสิทธิภาพ

```python
def embed_features(learner, xs, emb_szs):
    xs = xs.copy()
    for i, (col, (num_categories, emb_dim)) in enumerate(zip(learner.dls.cat_names, emb_szs)):
        # Get matrix containing each row's embedding vector
        emb = learner.model.embeds[i]
        emb_data = emb(tensor(xs[col], dtype=torch.int64))
        emb_names = [f'{col}_{j}' for j in range(emb_dim)]

        # Join the embedded category and drop the old feature column
        feat_df = pd.DataFrame(data=emb_data[:, :emb_dim], index=xs.index,
                               columns=emb_names)
        xs = xs.drop(col, axis=1)
        xs = xs.join(feat_df)
    return xs

# Assuming learner, xs, cont_names, cat_names, procs, dep_var, splits are defined
emb_szs = [('neighbourhood', 3)]
```

ภาพประกอบ 26 code แสดงการสร้าง embedded feature

การเข้ารหัสแบบ Entity Embedding จะใช้ไลบรารี่ Fastai โดยมีการกำหนดขนาด (dimension) ของคุณสมบัติที่ซ่อนอยู่ (embedded feature) ในตัวแปร neighbourhood เท่ากับ 3 ซึ่งจะกลายเป็นเวกเตอร์ (vector) ที่แทนคุณลักษณะของข้อมูลตัวแปรหมวดหมู่ในรูปแบบของตัวเลขจำนวนจริงที่แบบจำลองสามารถเรียนรู้และทำนายได้ง่ายขึ้น

neighbourhood_0	neighbourhood_1	neighbourhood_2
0.080254	-0.060312	0.061873
0.038078	-0.034595	0.059529
0.029849	0.013348	-0.084791
0.018630	-0.011961	0.011531
-0.007659	0.051591	-0.042207
...	...	...
-0.007659	0.051591	-0.042207
0.029849	0.013348	-0.084791
0.029849	0.013348	-0.084791
-0.007659	0.051591	-0.042207
0.022472	-0.019529	-0.024600

ภาพประกอบ 27 ตัวอย่าง embedded feature ที่ถูกแปลงไปเป็น vector ของข้อมูลในคอลัมน์ neighbourhood

สำหรับการเข้ารหัสแบบ One-hot Encoding จะมีการแทนค่าแต่ละค่าด้วยตัวแปรใหม่ที่เป็น binary มีค่าเป็น 0 หรือ 1 เท่านั้น โดยแต่ละ binary จะแทนค่าของ neighbourhood ที่เป็นไปได้แต่ละค่าเพื่อให้แบบจำลองสามารถเรียนรู้และทำนายได้ดีขึ้น

```
onehot_encoder = OneHotEncoder(sparse=False)
train_neighbourhood_onehot = onehot_encoder.fit_transform(train['neighbourhood_encoded'].values.reshape(-1, 1))
test_neighbourhood_onehot = onehot_encoder.transform(test['neighbourhood_encoded'].values.reshape(-1, 1))

train_neighbourhood_onehot_df = pd.DataFrame(train_neighbourhood_onehot,
                                             columns=[f'neighbourhood_{i}'
                                                      for i in range(train_neighbourhood_onehot.shape[1])])
test_neighbourhood_onehot_df = pd.DataFrame(test_neighbourhood_onehot,
                                            columns=[f'neighbourhood_{i}'
                                                     for i in range(test_neighbourhood_onehot.shape[1])])

train = pd.concat([train, train_neighbourhood_onehot_df], axis=1)
test = pd.concat([test, test_neighbourhood_onehot_df], axis=1)

train.drop(['neighbourhood', 'neighbourhood_encoded'], axis=1, inplace=True)
test.drop(['neighbourhood', 'neighbourhood_encoded'], axis=1, inplace=True)
```

ภาพประกอบ 28 code แสดงการเข้ารหัสแบบ One-hot Encoding

การเข้ารหัสด้วยวิธีการที่ต่างกันทำให้ฟีเจอร์ที่ได้มีความแตกต่างกัน โดยการเข้ารหัสแบบ Entity Embedding มีการเข้ารหัสในตัวแปร neighbourhood โดยกำหนดขนาด (dimension) เท่ากับ 3 ส่วนการเข้ารหัสแบบ One-hot Encoding จะมีการแทนค่าที่เป็นไปได้สำหรับตัวแปร โดยมีค่าเป็น 0 หรือ 1 จึงทำให้เกิดเป็นฟีเจอร์ใหม่สำหรับตัวแปร neighbourhood เท่ากับ 47 ฟีเจอร์ ดังนั้นจึงทำให้ในชุดข้อมูลที่ผ่านการเข้ารหัสแบบ Entity Embedding มีจำนวนฟีเจอร์ทั้งหมด 17 ฟีเจอร์ และการเข้ารหัสแบบ One-hot Encoding เกิดเป็นฟีเจอร์ใหม่สำหรับตัวแปร neighbourhood เท่ากับ 57 ฟีเจอร์

3.2.5 **การปรับปรุงข้อมูล** (Transform Data)

ในชุดข้อมูลมีการปรับปรุงข้อมูลให้มีการกระจายแบบปกติ สำหรับตัวแปรตาม (dependent variable) มีการปรับปรุงข้อมูลด้วยวิธี StandardScaler กับคอลัมน์ที่เป็นตัวแปรเชิงปริมาณทั้งหมด และสำหรับตัวแปรอิสระ (independent variable) มีการปรับการกระจายตัวของข้อมูล 2 วิธี คือ StandardScaler และ Logarithm ในคอลัมน์ price และเพื่อวัดประสิทธิภาพของแบบจำลองมีการทำการย้อนกลับของข้อมูล (inverse) ในตัวแปรตัวแปรอิสระเพื่อให้ข้อมูลกลับมาสู่รูปแบบเดิมก่อนที่จะนำไปใช้งานต่อ

3.2.6 การแบ่งชุดข้อมูลออกเป็นชุดข้อมูลสำหรับการฝึก (Training Data) และชุดข้อมูลสำหรับการทดสอบ (Test Data)

มีการแบ่งชุดข้อมูลออกเป็น 2 ส่วน ได้แก่ ชุดข้อมูลสำหรับการฝึก 80% และชุดข้อมูลสำหรับการทดสอบ 20% การแบ่งชุดข้อมูลออกเป็นชุดข้อมูลฝึกและชุดข้อมูลทดสอบมีการลบคอลัมน์ price ออกจากชุดข้อมูลฝึกเนื่องจากเป็นข้อมูลที่ผู้วิจัยต้องการทำนาย และเก็บคอลัมน์ที่เหลือลงในตัวแปร X ซึ่งจะเป็นชุดข้อมูลที่ไม่มีคอลัมน์ price อยู่ และมีการสร้างตัวแปร y โดยเลือกคอลัมน์ price จาก DataFrame และเก็บคอลัมน์นี้ลงในตัวแปร y ซึ่ง y จะเป็นชุดข้อมูลที่มีเฉพาะคอลัมน์ price เท่านั้น

3.3 การสร้างแบบจำลอง (Modeling)

ในงานวิจัยนี้เป็นการจัดการปัญหาเชิงถดถอย (regression) ซึ่งเป็นการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรอิสระ (independent variable) กับตัวแปรตาม (dependent variable) มีวัตถุประสงค์ในการทำนายค่าของตัวแปรตามจากค่าของตัวแปรอิสระ คือ price เพื่อทำนายความสัมพันธ์ระหว่างตัวแปรทั้งสองจึงมีการสร้างแบบจำลอง 4 ชนิด ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost เพื่อนำมาเปรียบเทียบประสิทธิการทำงานของแบบจำลอง ดังนี้

3.3.1 Neural Network

ในแบบจำลอง Neural Network มีการกำหนดจำนวนชั้น (layer) ได้แก่ (8, 2), (16, 8), (32, 16), (64, 32) และ (128, 64) เพื่อดูประสิทธิภาพในการทำนายเมื่อมีการเพิ่มขึ้นของจำนวนชั้นในแบบจำลอง

3.3.2 Random Forest

ในแบบจำลอง Random Forest มีการกำหนดพารามิเตอร์ ได้แก่ max_depth = 30, max_features = (n), min_samples_split = 5, min_samples_leaf = 5 และ n_estimators = 100 ซึ่งได้มีการกำหนดการเพิ่มขึ้นของ max_features นั่นคือจำนวนคุณลักษณะ (features) ที่สุ่มเลือกเข้ามาใช้ในการตัดสินใจเมื่อสร้างโหนดในต้นไม้ โดยที่ n จะเป็นจำนวนทั้งหมดของคุณลักษณะที่สามารถใช้ได้ เพื่อดูว่าการเพิ่มขึ้นหรือลดลงของ max_features มีผลต่อประสิทธิภาพของแบบจำลองมากน้อยเพียงใด

3.3.3 K-Nearest Neighbors (KNN)

ในแบบจำลอง KNN มีการกำหนดค่า k เนื่องจากเป็นการกำหนดจำนวนของข้อมูลที่ใช้ในการทำนาย (prediction) โดยหลักการของ KNN คือการใช้ข้อมูลจำนวน k ตัวอย่างที่ใกล้ที่สุดกับข้อมูลใหม่เพื่อทำนายค่าของข้อมูลใหม่ ในแบบจำลองนี้จึงได้มีการกำหนดค่า k เพื่อดูประสิทธิภาพของแบบจำลอง

3.3.4 XGBoost

ในแบบจำลอง XGBoost มีการกำหนดพารามิเตอร์ ได้แก่ max_depth = 30, learning_rate = (n), reg_alpha = 0.1, reg_lambda = 0.1, subsample = 0.9, colsample_bytree = 0.9 และ n_estimators = 100 ซึ่งได้มีการกำหนดการเพิ่มขึ้นของ learning rate นั่นคืออัตราการเรียนรู้ของแบบจำลองที่เป็นพารามิเตอร์สำคัญที่มีผลต่อการฝึก ซึ่งการเพิ่มขึ้นหรือลดลงของ learning rate จะมีผลต่อการควบคุมการเรียนรู้ของแบบจำลอง ในกรณีที่ค่า learning rate มีค่าน้อยการทำงานของแบบจำลองก็จะเปลี่ยนไปน้อย แต่ในทางตรงกันข้ามถ้าค่า learning rate มีค่ามากการทำงานของแบบจำลองก็จะเปลี่ยนไปมาก

3.4 **การประเมินผลแบบจำลอง** (Model Evaluation)

ในงานวิจัยฉบับนี้เป็นการจัดการปัญหา Regression ซึ่งเป็นการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรอิสระ (independent variable) กับตัวแปรตาม (dependent variable) ซึ่งมีวัตถุประสงค์ในการทำนายค่าของตัวแปรตามจากค่าของตัวแปรอิสระหรือใช้เพื่อทำนายแนวโน้มหรือความสัมพันธ์ระหว่างตัวแปรทั้งสอง จึงใช้การวัดประสิทธิภาพการทำงานของแบบจำลองด้วยค่า RMSE, MAE, R-squared และ MAPE

3.4.1 Root Mean Squared Error (RMSE)

การวัดความแตกต่างระหว่างค่าจริงและค่าทำนาย หากค่าที่ได้มีค่าที่น้อยแสดงถึงค่าทำนายนั้นประมาณค่าได้ใกล้เคียงกับค่าจริง โดย RMSE จะนำผลลัพธ์ที่ได้จาก MSE ไปคำนวณเป็นรากที่สอง (Root) เพื่อให้ได้ค่าที่มีมิติเดียวกับตัวแปรต้นที่ถูกวัด หากค่า RMSE ที่น้อยกว่า หมายถึง การทำนายที่แม่นยำ ส่วนค่าที่มากขึ้นแสดงถึงความคลาดเคลื่อนที่มากขึ้นของการทำนาย ดังสมการ (1)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \widehat{Yi})^2} \quad (1)$$

โดยที่

- Yi คือ ค่าจริงของข้อมูลในตัวอย่างที่ i
- $\widehat{Yi}$ คือ ค่าที่ถูกทำนายหรือค่าทำนายของโมเดลสำหรับข้อมูลที่ i
- n คือ จำนวนข้อมูลในตัวอย่าง

3.4.2 Mean Absolute Error (MAE)

การหาค่าเฉลี่ยของความแตกต่างสมบูรณ์ระหว่างค่าพยากรณ์และค่าจริง หากค่า MAE ยิ่งน้อยแสดงว่าค่าทำนายมีค่าใกล้เคียงกับค่าจริง ซึ่งจะใช้การหาค่าเฉลี่ยของความต่างระหว่างค่าทำนายและค่าจริงโดยไม่ใช้การยกกำลัง ทำให้มีค่าที่มีมิติเดียวกับตัวแปรต้นที่ถูกวัด ค่า MAE ที่น้อยแสดงถึงการทำนายที่แม่นยำ ส่วนค่าที่มากขึ้นแสดงถึงความคลาดเคลื่อนที่มากขึ้นของการทำนายดังสมการ (2)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Yi - \widehat{Yi}\right| \quad (2)$$

โดยที่

- Yi คือ ค่าจริงของข้อมูลในตัวอย่างที่ i
- $\widehat{Yi}$ คือ ค่าที่ถูกทำนายหรือค่าทำนายของโมเดลสำหรับข้อมูลที่ i
- n คือ จำนวนข้อมูลในตัวอย่าง

3.4.3 R-Squared **หรือ** Coefficient of Determination

ค่าทางสถิติที่ใช้ในการอธิบายความเปลี่ยนแปลงของตัวแปรตามโดยเทียบกับความเปลี่ยนแปลงของตัวแปรอิสระ หากค่า R-Squared ยิ่งเข้าใกล้ 1 หมายถึง แบบจำลองสามารถอธิบายข้อมูลได้เพียงพอทุกประการดังสมการ (3)

$$R^2 = 1 - \frac{SSres}{SStot} \quad (3)$$

โดยที่

- $SSres$ หมายถึง ผลรวมของค่าความแตกต่างระหว่างค่าจริงและค่าทำนาย (Sum of Squares of Residuals)
- $SStot$ หมายถึง ผลรวมของความแตกต่างระหว่างค่าจริงและค่าเฉลี่ยของข้อมูล (Total Sum of Squares)

3.4.4 Mean absolute percentage error (MAPE)

ค่าทางสถิติที่ใช้ในการวัดความคลาดเคลื่อนของการทำนายจากค่าจริงโดยใช้เปอร์เซ็นต์เป็นหน่วยในการวัด โดยค่า MAPE จะคำนวณความคลาดเคลื่อนเฉลี่ยของการทำนายเป็นเปอร์เซ็นต์ของค่าจริง หากค่า MAPE ที่น้อยสุดแสดงถึงการทำนายที่แม่นยำ ส่วนค่าที่มากขึ้นแสดงถึงความคลาดเคลื่อนที่มากขึ้นดังสมการที่ (4)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Yi-\widehat{Yi}}{Yi}\right| \times 100\% \quad (4)$$

โดยที่

- Yi คือ ค่าจริงของข้อมูลในตัวอย่างที่ i
- $\widehat{Yi}$ คือ ค่าที่ถูกทำนายหรือค่าทำนายของโมเดลสำหรับข้อมูลที่ i
- n คือ จำนวนข้อมูลในตัวอย่าง

บทที่ 4
ผลการดำเนินการวิจัย

ในการวิจัยการทำนายราคาที่พักบน Airbnb โดยการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณ โดยใช้ชุดข้อมูลจากเว็บไซต์ http://insideairbnb.com/get-the-data/ ที่เก็บรวบรวมข้อมูลต่าง ๆ เกี่ยวกับที่พักบน Airbnb โดยเลือกใช้ชุดข้อมูลของกรุงเทพมหานคร โดยในชุดข้อมูลมีการปรับปรุงข้อมูลให้มีการกระจายแบบเป็นปกติในตัวแปรตาม (dependent variable) มีการปรับปรุงข้อมูลด้วยวิธี StandardScaler กับคอลัมน์ที่เป็นตัวแปรเชิงปริมาณทั้งหมด และในตัวแปรอิสระ (independent variable) มีการปรับการกระจายตัวของข้อมูล 2 วิธี คือ StandardScaler และ Logarithm ในคอลัมน์ price และหลังจากนั้นเพื่อวัดประสิทธิภาพของแบบจำลองมีการทำการย้อนกลับของข้อมูล (inverse) ในตัวแปรตัวแปรอิสระเพื่อให้ข้อมูลกลับมาสู่รูปแบบเดิมก่อนที่จะนำไปใช้งานต่อ

ในงานวิจัยนี้เป็นการจัดการปัญหาเชิงถดถอย (regression) ซึ่งเป็นการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรอิสระ (independent variable) กับตัวแปรตาม (dependent variable) มีวัตถุประสงค์ในการทำนายค่าของตัวแปรตามจากค่าของตัวแปรอิสระ คือ price เพื่อทำนายความสัมพันธ์ระหว่างตัวแปรทั้งสองจึงมีการสร้างแบบจำลอง 4 ชนิด ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost เพื่อนำมาเปรียบเทียบประสิทธิการทำงานของแบบจำลอง ดังนี้.

4.1 **ประสิทธิภาพของแบบจำลอง** Neural Network

ในการทดลองนี้เป็นการปรับจำนวนโหนดในชั้นต่าง ๆ ในแบบจำลอง Neural Network ในชุดข้อมูลฝึก (training data) ดังแสดงในตาราง 7

ตาราง 7 เปรียบเทียบประสิทธิภาพของแบบจำลอง Neural Network ในชุดข้อมูลฝึกเมื่อมีการปรับจำนวนโหนดในชั้นต่าง ๆ

Transform Data	Layer	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	(8, 2)	882.94	657.61	0.15	1,775.51	1,496.77	-2.46
	(16, 8)	876.33	651.39	0.16	859.23	627.09	0.19
	(32, 16)	872.26	642.95	0.17	860.99	615.09	0.19
	(64, 32)	869.62	648.69	0.17	843.77	620.38	0.22
	(128, 64)	**865.86**	**645.01**	**0.18**	**838.96**	**630.13**	**0.23**
Logarithm	(8, 2)	879.38	653.76	0.15	869.20	648.03	0.17
	(16, 8)	873.68	645.48	0.16	860.30	645.68	0.19
	(32, 16)	870.92	646.88	0.17	859.18	614.28	0.19
	(64, 32)	866.58	639.04	0.17	848.47	625.83	0.21
	(128, 64)	**862.30**	**637.86**	**0.18**	**841.97**	**602.51**	**0.22**

โดยมีการเปรียบเทียบกับชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลอง ดังแสดงในตาราง 8

ตาราง 8 เปรียบเทียบประสิทธิภาพของแบบจำลอง Neural Network ในชุดข้อมูลทดสอบเมื่อมีการปรับจำนวนโหนดในชั้นต่าง ๆ

Transform Data	Layer	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	(8, 2)	898.02	658.13	0.12	1,770.70	1,488.87	-2.41
	(16, 8)	892.77	655.54	0.13	879.16	638.34	0.16
	(32, 16)	888.53	644.74	0.14	880.58	626.38	0.16
	(64, 32)	886.81	651.80	0.14	872.16	637.09	0.17
	(128, 64)	**884.15**	**647.37**	**0.15**	**871.82**	**652.93**	**0.17**
Logarithm	(8, 2)	899.33	661.29	0.12	886.97	654.63	0.14
	(16, 8)	893.34	652.46	0.13	880.16	654.92	0.16
	(32, 16)	891.86	655.81	0.13	879.45	626.03	0.16
	(64, 32)	889.41	649.26	0.14	872.42	641.28	0.17
	(128, 64)	**885.81**	**648.58**	**0.15**	**869.72**	**618.37**	**0.18**

จากตาราง 7 และ 8 เมื่อมีการการเปรียบเทียบข้อมูลระหว่างชุดข้อมูลฝึก (training data) และชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลอง ในมิติของการเข้ารหัส (encoding method) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ One-hot Encoding ในมิติของการปรับปรุงข้อมูล (transform data) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การปรับปรุงข้อมูลด้วยวิธี Logarithm โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติ มีค่า RMSE เท่ากับ 869.72 MAE เท่ากับ 618.37 และ R-squared เท่ากับ 0.18 ซึ่งมีจำนวนชั้นเท่ากับ (128, 64) จะเห็นว่าเมื่อมีจำนวนชั้นที่เพิ่มขึ้นแบบจำลองสามารถเรียนรู้ได้ดีมากขึ้น เนื่องจากมีการเรียนรู้คุณลักษณะที่ซับซ้อนมากขึ้นและมีความสัมพันธ์ระหว่างตัวแปรที่มีความซับซ้อนมากขึ้นในแบบจำลองซึ่งสามารถแสดงให้เห็นถึงลักษณะของข้อมูลได้ดีขึ้น

4.2 **ประสิทธิภาพของแบบจำลอง** Random Forest

ในการทดลองนี้เป็นการปรับจำนวนโหนดในชั้นต่าง ๆ ในแบบจำลอง Random Forest ในชุดข้อมูลฝึก (training data) ดังแสดงในตาราง 9

ตาราง 9 เปรียบเทียบประสิทธิภาพของแบบจำลอง Random Forest ในชุดข้อมูลฝึกเมื่อมีการเพิ่ม max_feature

Transform Data	max_ feature	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	0.1	731.97	531.47	0.42	773.70	567.53	0.34
	0.2	715.46	515.60	0.44	725.78	522.05	0.42
	0.3	698.04	499.77	0.47	703.77	501.87	0.46
	0.4	681.34	485.47	0.49	686.40	486.85	0.48
	0.5	657.06	463.14	0.53	673.53	474.06	0.50
	0.6	648.48	456.49	0.54	664.37	466.22	0.52
	0.7	**642.81**	**451.13**	**0.55**	**657.04**	**459.67**	**0.53**
Logarithm	0.1	729.53	528.73	0.41	774.86	568.56	0.34
	0.2	710.30	510.23	0.45	726.73	523.45	0.42
	0.3	690.73	492.29	0.48	704.17	503.23	0.46
	0.4	675.42	479.15	0.50	686.92	486.57	0.48
	0.5	651.40	457.73	0.53	673.73	474.51	0.50
	0.6	643.97	451.95	0.54	664.97	466.66	0.52
	0.7	**639.42**	**447.52**	**0.55**	**656.66**	**459.60**	**0.53**

โดยมีการเปรียบเทียบกับชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลอง ดังแสดงในตาราง 10

ตาราง 10 เปรียบเทียบประสิทธิภาพของแบบจำลอง Random Forest ในชุดข้อมูลทดสอบเมื่อมีการเพิ่ม max_feature

Transform Data	max_ feature	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	0.1	857.32	609.42	0.20	841.93	610.68	0.23
	0.2	851.77	603.14	0.21	830.39	594.14	0.25
	0.3	848.18	598.26	0.22	825.55	586.86	0.26
	0.4	843.76	595.50	0.23	820.31	582.60	0.27
	0.5	842.90	592.79	0.22	818.94	578.47	0.27
	0.6	841.36	591.01	0.23	817.09	575.92	0.27
	0.7	**840.33**	**590.56**	**0.24**	**816.18**	**573.80**	**0.28**
Logarithm	0.1	850.00	608.65	0.21	842.67	612.85	0.23
	0.2	844.82	603.17	0.22	829.63	594.65	0.25
	0.3	840.12	596.90	0.23	825.73	589.23	0.25
	0.4	835.99	592.90	0.24	821.28	583.21	0.26
	0.5	833.43	590.19	0.24	820.21	580.32	0.27
	0.6	833.49	589.30	0.24	817.46	577.58	0.28
	0.7	**832.56**	**587.63**	**0.25**	**817.23**	**575.17**	**0.29**

ในแบบจำลอง Random Forest มีการกำหนดพารามิเตอร์ ได้แก่ max_depth = 30, max_features = (n), min_samples_split = 5, min_samples_leaf = 5, n_estimators = 100

จากตาราง 9 และ 10 เมื่อมีการการเปรียบเทียบข้อมูลระหว่างชุดข้อมูลฝึก (training data) และชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลอง ในมิติของการเข้ารหัส (encoding method) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ One-hot Encoding ในมิติของการปรับปรุงข้อมูล (transform data) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การปรับปรุงข้อมูลด้วยวิธี StandardScaler โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติมีค่า RMSE เท่ากับ 816.18 MAE เท่ากับ 573.80 และ R-squared เท่ากับ 0.28 ซึ่งมีจำนวน max_features เท่ากับ 0.7 จะเห็นว่าเมื่อมีการเพิ่มขึ้นของ max_feature แบบจำลองจะมีประสิทธิภาพที่ดีขึ้น นั่นคือจำนวน

คุณลักษณะ (features) ที่สุ่มเลือกเข้ามาใช้ในการตัดสินใจเมื่อสร้างโหนดในต้นไม้มีเพิ่มขึ้นซึ่งสามารถช่วยลดความเชื่อมโยงระหว่างต้นไม้ ทำให้แบบจำลองมีประสิทธิภาพในการทำนายกับข้อมูลที่ไม่เคยเห็นมาก่อนได้ดีขึ้นเนื่องจากแบบจำลองไม่มีความเชื่อมโยงมากนักกับข้อมูลเดิมและมีความยืดหยุ่นมากขึ้นในการตัดสินใจ

4.3 **ประสิทธิภาพของแบบจำลอง** K-Nearest Neighbors (KNN)

ในการทดลองนี้เป็นการปรับจำนวนโหนดในชั้นต่าง ๆ ในแบบจำลอง K-Nearest Neighbors (KNN) ในชุดข้อมูลฝึก (training data) ดังแสดงในตาราง 11

ตาราง 11 เปรียบเทียบประสิทธิภาพของแบบจำลอง KNN ในชุดข้อมูลฝึกเมื่อมีการเพิ่มค่า K

Transform Data	K	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
StandardScaler	3	**675.62**	**471.56**	**0.50**	**693.26**	**487.09**	**0.45**
	4	717.43	509.54	0.44	739.84	529.27	0.40
	5	744.87	534.02	0.39	766.45	554.17	0.36
	6	764.85	551.96	0.36	786.39	572.48	0.32
	7	781.16	565.64	0.33	801.00	586.20	0.30
	8	792.17	576.19	0.32	811.51	595.33	0.28
	9	802.35	584.76	0.30	821.17	604.02	0.26
Logarithm	3	**669.30**	**466.17**	**0.51**	**694.26**	**489.09**	**0.41**
	4	710.80	502.34	0.44	739.79	529.28	0.40
	5	737.01	524.64	0.40	766.81	554.41	0.36
	6	759.37	544.24	0.37	786.63	572.58	0.32
	7	773.15	557.20	0.34	801.17	586.22	0.30
	8	783.02	565.69	0.33	811.42	595.16	0.28
	9	793.84	574.33	0.31	821.31	604.15	0.26

โดยมีการเปรียบเทียบกับชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลอง ดังแสดงในตาราง 12

ตาราง 12 เปรียบเทียบประสิทธิภาพของแบบจำลอง KNN ในชุดข้อมูลทดสอบเมื่อมีการเพิ่มค่า K

Transform Data	K	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
StandardScaler	3	945.41	666.83	0.03	989.75	697.99	-0.07
	4	926.10	659.08	0.07	962.96	683.87	-0.01
	5	913.80	651.98	0.09	952.70	680.05	0.01
	6	904.98	647.93	0.11	941.35	674.27	0.04
	7	899.13	644.22	0.12	936.29	675.30	0.05
	8	897.24	643.56	0.12	935.03	674.52	0.05
	9	**894.94**	**643.47**	**0.13**	**930.76**	**674.70**	**0.06**
Logarithm	3	928.83	655.42	0.06	990.38	698.49	-0.07
	4	911.53	651.75	0.10	963.01	683.81	-0.01
	5	905.49	649.76	0.11	952.14	679.92	0.01
	6	900.69	643.04	0.12	940.69	673.68	0.04
	7	893.50	640.13	0.13	936.57	675.72	0.05
	8	889.13	637.79	0.14	935.29	674.83	0.05
	9	**885.38**	**636.56**	**0.15**	**931.12**	**675.24**	**0.06**

จากตาราง 11 และ 12 เมื่อมีการการเปรียบเทียบข้อมูลระหว่างชุดข้อมูลฝึก (training data) และชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลองจะเห็นว่าในข้อมูลฝึกเมื่อมีการเพิ่มขึ้นของค่า K ประสิทธิภาพของแบบจำลองจะลดลง โดยผลลัพธ์ที่ดีที่สุดในมิติของการเข้ารหัส (encoding method) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ Entity Embedding ในมิติของการปรับปรุงข้อมูล (transform data) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การปรับปรุงข้อมูลด้วยวิธี Logarithm โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติมีค่า RMSE เท่ากับ 669.30 MAE เท่ากับ 466.17 และ R-squared เท่ากับ 0.51 ซึ่งมีค่า K เท่ากับ 3 แต่ในข้อมูลทดสอบเมื่อมีการเพิ่มขึ้นของค่า K ประสิทธิภาพของแบบจำลองจะเพิ่มขึ้น โดยผลลัพธ์ที่ดีที่สุดในมิติของการเข้ารหัส (encoding method) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ Entity Embedding ในมิติของการปรับปรุงข้อมูล (transform data) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การ

ปรับปรุงข้อมูลด้วยวิธี Logarithm โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติมีค่า RMSE เท่ากับ 885.38, MAE เท่ากับ 636.56 และ R-squared เท่ากับ 0.15 ซึ่งมีค่า K เท่ากับ 9

จากผลการทดลองจะเห็นว่าเมื่อมีการเพิ่มขึ้นของค่า K ในข้อมูลฝึกประสิทธิภาพของแบบจำลองจะลดลง แต่ในข้อมูลทดสอบประสิทธิภาพของแบบจำลองจะเพิ่มขึ้น อาจเกิดจากเมื่อมีการเพิ่มค่า K ในข้อมูลฝึกแบบจำลองมีแนวโน้มที่จะกลายเป็นเส้นตรงไปที่ข้อมูลฝึก ซึ่งอาจทำให้มีความผิดพลาดในการทำนายบนข้อมูลทดสอบ แต่ในกรณีตรงกันข้ามเมื่อเพิ่มค่า K ในข้อมูลทดสอบจะมีการใช้ข้อมูลจำนวนมากเพื่อตัดสินใจ และเพิ่มความสามารถในการจัดกลุ่มข้อมูลที่ถูกต้องมากขึ้นเนื่องจากมีการพิจารณาข้อมูลจำนวนมากในการตัดสินใจ จึงทำให้ประสิทธิภาพของแบบจำลองดีขึ้น

4.4 ประสิทธิภาพของแบบจำลอง XGBoost

ในการทดลองนี้เป็นการปรับจำนวนโหนดในชั้นต่าง ๆ ในแบบจำลอง XGBoost ในชุดข้อมูลฝึก (training data) ดังแสดงในตาราง 13

ตาราง 13 แสดงการเปรียบเทียบประสิทธิภาพของแบบจำลอง XGBoost ในชุดข้อมูลฝึกเมื่อมีการเพิ่ม learning_rate

Transform Data	learning_ rate	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	0.01	706.87	517.48	0.45	721.73	527.41	0.43
	0.02	580.59	407.89	0.63	606.69	426.46	0.60
	0.03	502.58	339.63	0.72	535.86	363.19	0.69
	0.04	447.96	291.26	0.78	489.48	322.44	0.74
	0.05	406.98	255.51	0.82	454.79	291.90	0.77
	0.06	376.90	228.68	0.84	428.18	268.81	0.80
	0.07	**355.02**	**207.65**	**0.86**	**407.28**	**250.26**	**0.82**

ตาราง 13 (ต่อ)

Transform Data	learning_ rate	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Logarithm	0.01	466.23	335.30	0.76	720.11	526.14	0.44
	0.02	298.53	176.56	0.90	605.11	426.02	0.59
	0.03	253.39	110.31	0.92	532.62	362.48	0.71
	0.04	243.04	82.07	0.93	485.59	320.17	0.76
	0.05	240.70	69.84	0.93	453.69	290.29	0.77
	0.06	240.16	64.48	0.94	423.08	267.87	0.81
	0.07	**240.04**	**62.14**	**0.95**	**405.11**	**249.76**	**0.82**

โดยมีการเปรียบเทียบกับชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลอง ดังแสดงในตาราง 14

ตาราง 14 แสดงการเปรียบเทียบประสิทธิภาพของแบบจำลอง XGBoost ในชุดข้อมูลทดสอบเมื่อมีการเพิ่ม learning_rate

Transform Data	learning_ rate	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	0.01	882.93	620.84	0.15	839.59	607.56	0.23
	0.02	869.71	597.32	0.16	805.98	571.32	0.29
	0.03	868.70	593.12	0.16	796.85	557.62	0.31
	0.04	866.48	587.88	0.17	796.07	554.50	0.31
	0.05	865.94	583.00	0.18	795.67	553.14	0.31
	0.06	864.88	582.96	0.19	794.52	552.23	0.32
	0.07	**862.71**	**580.73**	**0.21**	**791.22**	**550.42**	**0.33**

ตาราง 14 (ต่อ)

Transform Data	learning_ rate	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Logarithm	0.01	893.12	625.55	0.13	838.60	606.50	0.24
	0.02	891.03	623.82	0.14	804.88	570.22	0.29
	0.03	890.08	622.19	0.14	795.55	556.60	0.32
	0.04	889.49	621.54	0.15	793.89	552.78	0.33
	0.05	887.48	617.23	0.16	792.00	550.11	0.35
	0.06	883.28	616.17	0.17	791.52	549.23	0.37
	0.07	**881.21**	**611.53**	**0.20**	**787.22**	**544.42**	**0.37**

ในแบบจำลอง XGBoost มีการกำหนดพารามิเตอร์ ได้แก่ max_depth = 30, learning_rate = (n), reg_alpha = 0.1, reg_lambda = 0.1, subsample = 0.9, colsample_bytree = 0.9, n_estimators = 100

จากตาราง 13 และ 14 เมื่อมีการการเปรียบเทียบข้อมูลระหว่างชุดข้อมูลฝึก (training data) และชุดข้อมูลทดสอบ (test data) เพื่อประเมินประสิทธิภาพของแบบจำลองในมิติของการเข้ารหัส (encoding method) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ One-hot Encoding ในมิติของการปรับปรุงข้อมูล (transform data) วิธีที่ให้ผลลัพธ์ดีที่สุด คือ การปรับปรุงข้อมูลด้วยวิธี StandardScaler โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติมีค่า RMSE เท่ากับ 787.22 , MAE เท่ากับ 544.42 และ R-squared เท่ากับ 0.37 ซึ่งมีค่า learning_rate เท่ากับ 0.7 เห็นว่าเมื่อมีการเพิ่มขึ้นของ learning_rate จะช่วยให้แบบจำลองมีประสิทธิภาพมากขึ้น เนื่องจากค่า learning rate มีบทบาทสำคัญในการกำหนดการเคลื่อนไหวของแบบจำลองในการปรับค่าพารามิเตอร์ เมื่อเพิ่ม learning rate อาจช่วยให้แบบจำลองมีการปรับค่าได้มากขึ้นในแต่ละรอบของการฝึก ทำให้มีโอกาสในการพบค่าที่ดีกว่า

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

ในการวิจัยการทำนายราคาที่พักบน Airbnb โดยการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณ โดยใช้ชุดข้อมูลจากเว็บไซต์ http://insideairbnb.com/get-the-data/ ที่เก็บรวบรวมข้อมูลต่าง ๆ เกี่ยวกับที่พักบน Airbnb โดยเลือกใช้ชุดข้อมูลของกรุงเทพมหานคร ผู้วิจัยได้วัดประสิทธิภาพของแบบจำลองเพื่อนำมาเปรียบเทียบและสรุปผล โดยแบ่งหัวข้อในการสรุปผลได้ดังต่อไปนี้

5.1 สรุปผลการวิจัย

จากผลการทดสอบประสิทธิภาพของแบบจำลองในชุดข้อมูลฝึก (training data) ผ่านวิธีการเข้ารหัสแบบ Entity Embedding และ One-hot Encoding ผู้วิจัยได้เลือกผลลัพธ์ที่ดีที่สุดทั้ง 3 มิติ ได้แก่ มิติของการเข้ารหัส (encoding method) มิติของการปรับปรุงข้อมูล (transform data) และมิติของการทดสอบประสิทธิภาพของแบบจำลอง (model evaluation) ในมิติของการเข้ารหัสวิธีที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ Entity Embedding ในมิติของการปรับปรุงข้อมูลวิธีที่ให้ผลลัพธ์ดีที่สุด คือ การปรับปรุงข้อมูลด้วยวิธี Logarithm และในมิติของการทดสอบประสิทธิภาพของแบบจำลองแบบจำลองที่ให้ผลลัพธ์ดีที่สุด คือ XGBoost โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติ มีค่า RMSE เท่ากับ 240.04, MAE เท่ากับ 62.14 และ R-squared เท่ากับ 0.95 ซึ่งเมื่อพิจารณาในมิติเดียวกันแต่ด้วยการเข้ารหัสแบบ One-hot Encoding วิธีการปรับปรุงข้อมูลและการทดสอบประสิทธิภาพของแบบจำลองยังคงเป็นการปรับปรุงข้อมูลด้วยวิธี Logarithm ในแบบจำลอง XGBoost เช่นเดียวกับการเข้ารหัสแบบ Entity Embedding โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติ มีค่า RMSE เท่ากับ 405.11 MAE เท่ากับ 249.76 และ R-squared เท่ากับ 0.82 ดังแสดงในตาราง 15

ตาราง 15 แสดงผลการทดสอบประสิทธิภาพของแบบจำลองในชุดข้อมูลฝึก

Transform Data	Model	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	NN	865.86	645.01	0.18	838.96	630.13	0.23
	RF	642.81	451.13	0.55	657.04	459.67	0.53
	KNN	675.62	471.56	0.50	693.26	487.09	0.45
	XGB	355.02	207.65	0.86	407.28	250.26	0.82
Logarithm	NN	862.30	637.86	0.18	841.97	602.51	0.22
	RF	639.42	447.52	0.55	656.66	459.60	0.53
	KNN	669.30	466.17	0.51	694.26	489.09	0.41
	XGB	**240.04**	**62.14**	**0.95**	**405.11**	**249.76**	**0.82**

ผลการทดสอบประสิทธิภาพของแบบจำลองในชุดข้อมูลทดสอบ (test data) ในมิติของการเข้ารหัสวิธีที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ One-hot Encoding ในมิติของการปรับปรุงข้อมูลวิธีที่ให้ผลลัพธ์ดีที่สุด คือ การปรับปรุงข้อมูลด้วยวิธี Logarithm และในมิติของการทดสอบประสิทธิภาพของแบบจำลองแบบจำลองที่ให้ผลลัพธ์ดีที่สุด คือ XGBoost โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติ มีค่า RMSE เท่ากับ 787.22, MAE เท่ากับ 544.42 และ R-squared เท่ากับ 0.37 ซึ่งเมื่อพิจารณาในมิติเดียวกันแต่ด้วยการเข้ารหัสแบบ Entity Embedding วิธีการปรับปรุงข้อมูลที่ให้ผลลัพธ์ดีที่สุด คือ การปรับปรุงข้อมูลด้วยวิธี Logarithm และในมิติของการทดสอบประสิทธิภาพของแบบจำลอง แบบจำลองที่ให้ผลลัพธ์ดีที่สุด คือ Random Forest โดยเมื่อพิจารณาค่าความคลาดเคลื่อนในทุก ๆ มิติ มีค่า RMSE เท่ากับ 832.56 MAE เท่ากับ 587.63 และ R-squared เท่ากับ 0.25 ดังแสดงในตาราง 16

ตาราง 16 แสดงผลการทดสอบประสิทธิภาพของแบบจำลอง

Transform Data	Model	Entity Embedding			One-hot Encoding		
		RMSE	MAE	R^2	RMSE	MAE	R^2
Standard-Scaler	NN	884.15	647.37	0.15	871.82	652.93	0.17
	RF	840.33	590.56	0.24	816.18	573.80	0.28
	KNN	894.94	643.47	0.13	930.76	674.70	0.06
	XGB	862.71	580.73	0.21	791.22	550.42	0.33
Logarithm	NN	885.81	648.58	0.15	869.72	618.37	0.18
	RF	**832.56**	**587.63**	**0.25**	817.23	575.17	0.27
	KNN	885.38	636.56	0.15	931.12	675.24	0.06
	XGB	881.21	611.53	0.20	**787.22**	**544.42**	**0.37**

งานวิจัยนี้เป็นการศึกษาการทำนายราคาที่พักบน Airbnb ในพื้นที่กรุงเทพมหานคร โดยใช้วิธีการเข้ารหัสแบบ Entity Embedding และ One-hot Encoding ในชุดข้อมูลมีการปรับปรุงข้อมูลให้มีการกระจายแบบเป็นปกติ สำหรับตัวแปรตาม (dependent variable) มีการปรับปรุงข้อมูลด้วยวิธี StandardScaler กับคอลัมน์ที่เป็นตัวแปรเชิงปริมาณทั้งหมด และตัวแปรอิสระ (independent variable) มีการปรับการกระจายตัวของข้อมูล 2 วิธี คือ StandardScaler และ Logarithm ในคอลัมน์ price และเพื่อวัดประสิทธิภาพของแบบจำลองมีการทำการย้อนกลับของข้อมูล (inverse) ในตัวแปรตัวแปรอิสระเพื่อให้ข้อมูลกลับมาสู่รูปแบบเดิมก่อนที่จะนำไปใช้งานต่อในการวัดประสิทธิภาพของแบบจำลอง เพื่อทำนายความสัมพันธ์ระหว่างตัวแปรทั้งสองจึงมีการสร้างแบบจำลอง 4 ชนิด ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost เพื่อนำมาเปรียบเทียบประสิทธิการทำงานของแบบจำลอง โดยพิจารณาค่าความคลาดเคลื่อนด้วยค่า RMSE, MAE และ R-squared

ผลการวิจัยแสดงให้เห็นว่าผลการทดสอบประสิทธิภาพของแบบจำลองในชุดข้อมูลฝึกเกิดการเรียนรู้ที่มากเกินไป (overfit) เมื่อเปรียบเทียบกับชุดข้อมูลทดสอบ อาจเป็นผลมาจากการเข้ารหัสแบบ Entity Embedding ซึ่งทำให้เกิดการเรียนรู้มากเกินไปในชุดข้อมูลฝึกเมื่อเปรียบเทียบกับชุดข้อมูลทดสอบ โดยเฉพาะถ้ามีจำนวนข้อมูลฝึกน้อยเกินไปหรือมีความซับซ้อนในข้อมูลหมวดหมู่ (categorical variable) ซึ่งส่งผลให้แบบจำลองมีความสามารถในการเรียนรู้ลักษณะ

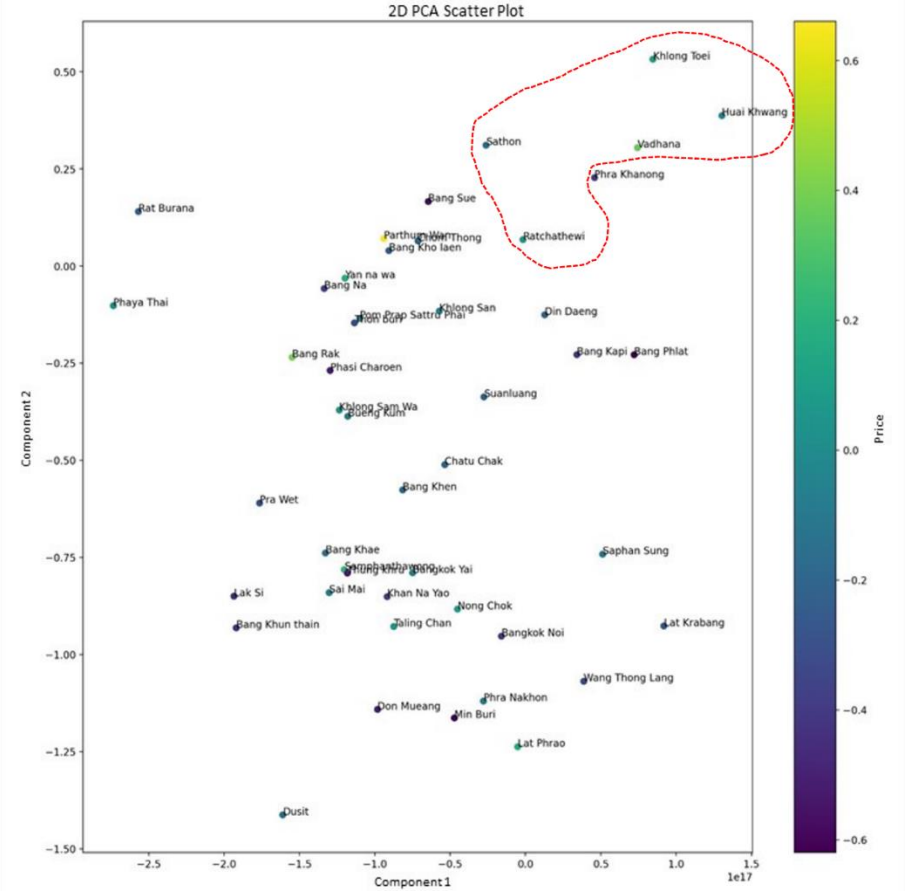
ของข้อมูลฝึกได้ดีมากเกินไปเมื่อเปรียบเทียบกับข้อมูลทดสอบที่มีลักษณะที่แตกต่าง อาจทำให้ประสิทธิภาพของแบบจำลองในข้อมูลทดสอบมีประสิทธิภาพแย่ลง ซึ่งเมื่อพิจารณาในชุดข้อมูลทดสอบการเข้ารหัสแบบ One-hot Encoding ร่วมกับการทำ Logarithm ในแบบจำลอง XGBoost ให้ผลลัพธ์ที่ดีที่สุด แต่เมื่อเปรียบเทียบกับการเข้ารหัสแบบ Entity Embedding ผลการทดสอบประสิทธิภาพของการเข้ารหัสทั้ง 2 แบบไม่ได้แตกต่างกันมากนักและไม่ได้มีประสิทธิภาพในการทำงานที่สูง นั่นเป็นเพราะข้อมูลไม่ได้มีจำนวนรายการที่เยอะและฟีเจอร์ที่มีในชุดข้อมูลไม่เพียงพอต่อการทำนายราคา จึงมีความเป็นไปได้ที่จะมีปัจจัยอื่นที่มีผล เช่น ตำแหน่งที่ตั้ง สิ่งอำนวยความสะดวกของที่พัก การตกแต่งภายใน จึงมีความเป็นไปได้ที่ปัจจัยต่าง ๆ เหล่านี้จะมีผลต่อการทำนายราคา แต่การเข้ารหัสแบบ Entity Embedding สามารถใช้เทคนิคการลดมิติของข้อมูล (PCA : Principal Component Analysis) เพื่อลดขนาดของ Embedded Feature ลงมาให้สามารถแสดงผลในรูปแบบการแสดงภาพ (visualization) ได้โดยที่ยังสามารถรักษาความสัมพันธ์ของข้อมูลในการแสดงผลได้ดี

5.2 อภิปรายผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาการแปลงข้อมูลเชิงกลุ่มให้เป็นข้อมูลเชิงปริมาณโดยใช้เทคนิคการเข้ารหัสแบบ Entity Embedding และ One-hot Encoding เพื่อทำนายราคาโดยใช้ชุดข้อมูลที่พักบน Airbnb ในชุดข้อมูลของกรุงเทพมหานคร ผู้วิจัยได้เปรียบเทียบผลของการเข้ารหัสตัวแปรเชิงกลุ่มโดยใช้เทคนิคการเข้ารหัส 2 ประเภท ได้แก่ Entity Embedding และ One-hot Encoding โดยการเข้ารหัสที่ให้ผลลัพธ์ดีที่สุด คือ การเข้ารหัสแบบ One-hot Encoding ซึ่งเมื่อเปรียบเทียบกับการเข้ารหัสแบบ Entity Embedding แล้วมีผลลัพธ์ที่ไม่ได้แตกต่างกันมากนัก ในขั้นตอนการปรับปรุงข้อมูล (transform data) มีการปรับปรุงข้อมูลให้มีการกระจายแบบปกติสำหรับตัวแปรตาม (dependent variable) มีการปรับปรุงข้อมูลด้วยวิธี StandardScaler กับคอลัมน์ที่เป็นตัวแปรเชิงปริมาณทั้งหมด และสำหรับตัวแปรอิสระ (independent variable) มีการปรับการกระจายตัวของข้อมูล 2 วิธี คือ StandardScaler และ Logarithm ในคอลัมน์ price ซึ่งผลลัพธ์การปรับปรุงข้อมูลด้วยวิธี Logarithm ให้ผลลัพธ์ที่ดีที่สุดในมิติของการปรับปรุงข้อมูล ซึ่งก็ให้ให้ผลลัพธ์ที่ไม่แตกต่างกันมากนักเมื่อเปรียบเทียบกับการปรับปรุงข้อมูลด้วยวิธี StandardScaler และเมื่อพิจารณาผลลัพธ์ของการทดสอบประสิทธิภาพในแบบจำลองทั้ง 4 ประเภท ได้แก่ Neural Network, Random Forest, K-Nearest Neighbors (KNN) และ XGBoost พบว่าแบบจำลองที่ให้ผลลัพธ์ดีที่สุด คือ XGBoost แต่ก็ไม่ได้มีประสิทธิภาพสูงสุดที่โดด

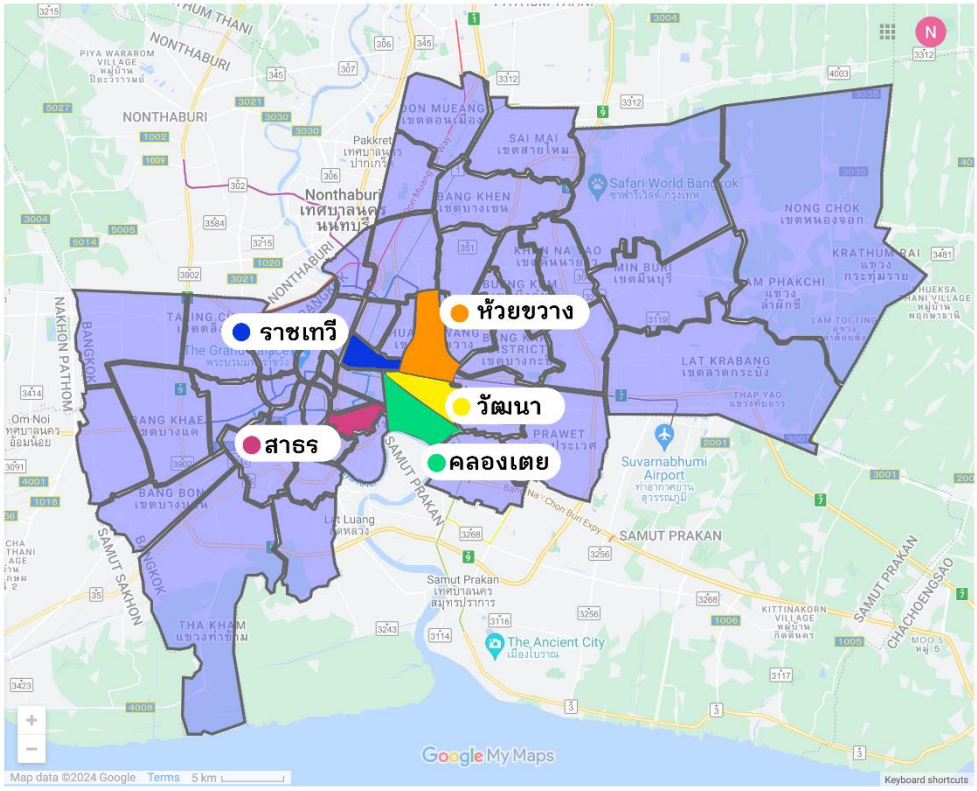
เด่นจากแบบจำลองทั้ง 4 ประเภท ซึ่งเมื่อพิจารณาในทุก ๆ มิติร่วมกันอาจมีผลจากชุดข้อมูลที่ไม่ได้ครอบคลุมไปถึงปัจจัยอื่น เช่น ตำแหน่งที่ตั้ง สิ่งอำนวยความสะดวกของที่พัก การตกแต่งภายใน จึงมีความเป็นไปได้ที่ปัจจัยต่าง ๆ เหล่านี้จะมีผลต่อการทำนายราคา ดังนั้นการพิจารณาปัจจัยและตัวแปรอื่น ๆ ร่วมด้วยอาจจะมีผลต่อการกำหนดราคาของที่พักบน Airbnb ซึ่งอาจช่วยให้แบบจำลองการทำนายราคามีความแม่นยำและถูกต้องมากยิ่งขึ้น

แม้ว่าการเข้ารหัสแบบ Entity Embedding จะไม่ได้ให้ผลลัพธ์ที่ดีที่สุด แต่เป็นวิธีที่น่าสนใจที่สามารถนำไปประยุกต์ใช้กับข้อมูลที่มีค่าความหลากหลายสูงได้ เพราะสามารถใช้เทคนิคการลดมิติของข้อมูล (PCA : Principal Component Analysis) เพื่อลดขนาดของ Embedded Feature ลงมาให้สามารถแสดงผลในรูปแบบการแสดงภาพ (visualization) เพื่อให้เห็นความสัมพันธ์ของข้อมูลได้ง่ายขึ้น โดยที่ยังสามารถรักษาความสัมพันธ์ของข้อมูลได้ดี



ภาพประกอบ 29 การแสดงภาพของตัวแปร neighbourhood ที่ใช้เทคนิคการลดมิติของข้อมูล (PCA)

ในงานวิจัยฉบับนี้จากภาพประกอบ 29 พบว่าการลดมิติของข้อมูลด้วยวิธี PCA (Principal Component Analysis) จุดบนภาพไม่ได้มาจากองค์ประกอบ (component) ที่เกี่ยวข้องโดยตรงกับที่ตั้งทางภูมิศาสตร์แต่กลับสอดคล้องกับที่ตั้งทางภูมิศาสตร์ จึงเป็นสิ่งที่น่าสนใจและสะท้อนถึงตัวแปรอื่น ๆ ในชุดข้อมูลที่นำมาใช้ในงานวิจัย เช่น ประเภทของห้องพัก ราคารายวันสำหรับการเข้าพัก จำนวนรีวิว จากการแสดงภาพจะพบว่าเขตคลองเตย เขตสาธร เขตห้วยขวาง เขตวัฒนา และเขตราชเทวี เป็นกลุ่มข้อมูลที่มีความถี่สูงและมีจำนวนรายการเพียงพอที่สามารถอยู่ในกลุ่มที่สอดคล้องกับที่ตั้งทางภูมิศาสตร์ ซึ่งทั้ง 5 เขตเป็นเขตที่มีความสำคัญทางธุรกิจ อาจเป็นเพราะเขตเหล่านี้มีพื้นที่ตั้งอยู่ใจกลางกรุงเทพมหานครซึ่งมีระบบโครงสร้างพื้นฐานที่เอื้ออำนวยต่อการทำธุรกิจ และเป็นที่ตั้งของศูนย์การค้า บริษัท และองค์กรชั้นนำ เมื่อนำมาเทียบกับแผนที่ของกรุงเทพมหานครจะพบว่าทั้ง 5 เขต เป็นเขตพื้นที่ที่ตั้งอยู่ในใกล้กัน ดังภาพประกอบ 30



ภาพประกอบ 30 แผนที่กรุงเทพมหานคร

ที่มา : (Maps, 2023)

5.3 ข้อเสนอแนะ

1. ในการวิจัยนี้ในขั้นตอนการเตรียมข้อมูลมีการลบข้อมูลที่เป็นข้อมูลที่หายไป (missing data) และข้อมูลส่วนเกิน (outlier) ออกไปอาจทำให้ข้อมูลที่สำคัญบางส่วนถูกลบทิ้งไป ซึ่งอาจมีผลต่อการทำนายของแบบจำลอง หากมีวิธีที่ไม่จำเป็นจะต้องลบข้อมูลดังกล่าว อาจช่วยเพิ่มประสิทธิภาพในการทำนายของแบบจำลองให้มีประสิทธิภาพมากขึ้น

2. การเข้ารหัสด้วยแบบ One-hot Encoding จะเป็นการเพิ่มขนาดของข้อมูลซึ่งมีความต้องการพื้นที่ในการเก็บข้อมูลมากกว่าการเข้ารหัสแบบ Entity Embedding เพราะเมื่อมีจำนวนหมวดหมู่มากหรือข้อมูลมีการเข้ารหัสแบบ binary ก็จะไม่สามารถจับต้องความสัมพันธ์แบบต่อเนื่องได้ซึ่งอาจทำให้แบบจำลองไม่สามารถเรียนรู้ความสัมพันธ์ระหว่างประเภทในคุณลักษณะเดียวกันได้

3. เนื่องจากในการวิจัยนี้ได้ใช้การเข้ารหัสเพียง 2 ประเภท ได้แก่ Entity Embedding และ One-hot Encoding เท่านั้น ดังนั้นอาจจะมีการเข้ารหัสชนิดอื่นอื่นที่สามารถเรียนรู้ได้ดีกับจำนวนข้อมูลที่มีและทำนายค่าออกมากได้มีความแม่นยำและมีประสิทธิภาพที่ดีกว่า

4. งานวิจัยนี้หากมีปริมาณข้อมูลเพียงพอ เช่น ตำแหน่งที่ตั้ง สิ่งอำนวยความสะดวกของที่พัก การตกแต่งภายใน ที่สามารถที่จะใช้สร้างแบบจำลองที่มีประสิทธิภาพได้มากขึ้นก็จะเห็นความสัมพันธ์ระหว่างแต่ละหมวดหมู่ได้ดีขึ้นจากการทำ Entity Embedding

บรรณานุกรม

Amzhao. (2023). *Quantum Neural Networks*. Medium. https://medium.com/mit-6-s089-intro-to-quantum-computing/quantum-neural-networks-7b5bc469d984

Baruah, I. D. (2023). *All you need to know about encoding techniques*. Medium. https://medium.com/anolytics/all-you-need-to-know-about-encoding-techniques-b3a0af68338b

Gnat, S. (2021). Impact of Categorical Variables Encoding on Property Mass Valuation. *Procedia Computer Science*, *192*, 3542-3550. https://doi.org/https://doi.org/10.1016/j.procs.2021.09.127

Guo, C., & Berkhahn, F. (2016). Entity Embeddings of Categorical Variables. *arXiv*. https://arxiv.org/pdf/1604.06737

Hooi, E. K. J., Zainal, A., Kassim, M. N., & Ayub, Z. (2022, 6-7 Oct. 2022). Feature Encoding For High Cardinality Categorical Variables Using Entity Embeddings: A Case Study in Customs Fraud Detection. 2022 International Conference on Cyber Resilience (ICCR),

Lee, C. (2023). How can we use neural network with entity embedding for product valuations? A case study for the car industry. *International Journal of Information Management Data Insights*, *3*(2), 100187. https://doi.org/https://doi.org/10.1016/j.jjimei.2023.100187

Ma, Y., & Zhang, Z.-j. (2020). Travel Mode Choice Prediction Using Deep Neural Networks With Entity Embeddings. *IEEE Access*, *8*, 64959-64970.

Maps, G. M. (2023). แผนที่กรุงเทพ *50* เขต. https://www.google.com/maps/d/viewer?mid=19vgGq-gj8wIK47tMoXBIT7Gff4U&hl=en_US&ll=13.725127956901979%2C100.63339249999997&z=10

R, S. E. (2024). *Understand Random Forest Algorithms With Examples*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Saxena, S. (2023). *What are Categorical Data Encoding Methods | Binary Encoding*.

Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/

Srivastava, T. (2024). *A Complete Guide to K-Nearest Neighbors*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Verma, N. (2022). *XGBoost Algorithm Explained in Less Than 5 Minutes*. Medium. https://medium.com/@techynilesh/xgboost-algorithm-explained-in-less-than-5-minutes-b561dcc1ccee

Wang, B., Shaaban, K., & Kim, I. (2019). Reveal the hidden layer via entity embedding in traffic prediction. *Procedia Computer Science*, *151*, 163-170. https://doi.org/https://doi.org/10.1016/j.procs.2019.04.025

ประวัติผู้เขียน