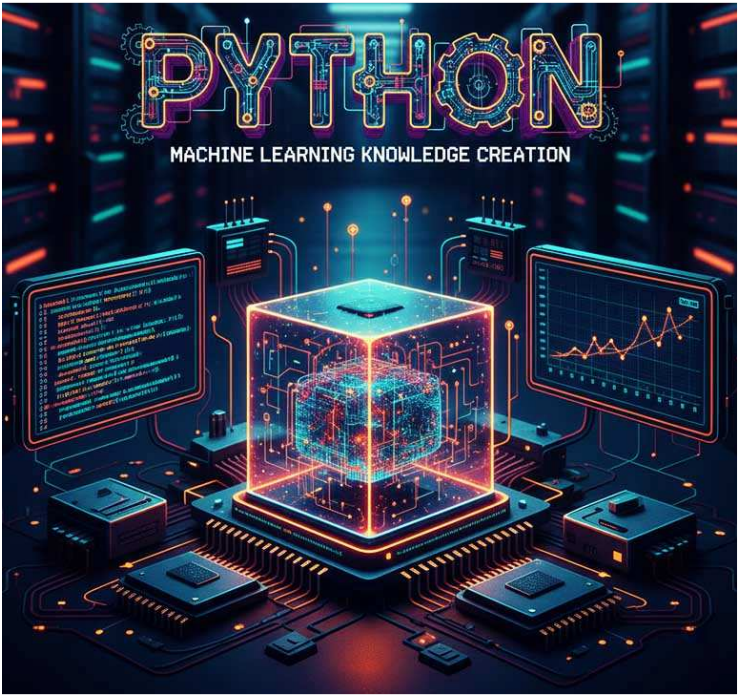
ความรู้เกี่ยวกับ การสร้างการเรียนรู้ให้เครื่องคอมพิวเตอร์ ด้วยภาษาไพธอน: แนวทางสำหรับการปฏิบัติ



: ปวีณ โชคนุกูล

ความรู้เกี่ยวกับ การสร้างการเรียนรู้ให้ เครื่องคอมพิวเตอร์ ด้วยภาษาไพธอน: แนวทางสำหรับการปฏิบัติ

ปวีณ โชคนุกูล

2569

ชื่อหนังสือ: ความรู้เกี่ยวกับ การสร้างการเรียนรู้ให้ เครื่องคอมพิวเตอร์
ด้วยภาษาไพธอน: แนวทางสำหรับการปฏิบัติ
Book title: Knowledge of Machine Learning with Python:
A Practical Guide
ชื่อผู้เขียน: ปวีณ โชคนุกูล (Paween Chokenukul)
พิมพ์ครั้งที่ 1 พฤษภาคม พุทธศักราช 2569
ลิขสิทธิ์ของ: ปวีณ โชคนุกูล
ราคา: 249 บาท

ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ
ปวีณ โชคนุกูล.
ความรู้เกี่ยวกับ การสร้างการเรียนรู้ให้ เครื่องคอมพิวเตอร์ ด้วยภาษาไพธอน: แนวทาง สำหรับการปฏิบัติ.-- สงขลา : [ม.ป.พ.], 2569.
618 หน้า.
1. ไพธอน (ภาษาคอมพิวเตอร์). 2. การเรียนรู้แบบถ่ายโอน (การเรียนรู้ของเครื่อง).
I. ปวีณ โชคนุกูล, ผู้วาดภาพประกอบ. II. ชื่อเรื่อง.
006.31
ISBN 978-616-631-865-4

จัดทำโดย: ปวีณ โชคนุกูล

คำนำ

หนังสือเล่มนี้จัดทำขึ้นเพื่อให้ความรู้ภาษาไทย เกี่ยวกับการเขียนคำสั่งภาษาไพธอน เพื่อสร้างการเรียนรู้ให้เครื่องคอมพิวเตอร์ โดยหนังสือได้แนะนำการเรียนรู้ของเครื่องคอมพิวเตอร์แบบต่าง ๆ และอัลกอริธึม หรือขั้นตอนของวิธีการแก้ปัญหาเรื่องใดเรื่องหนึ่งที่มีลักษณะเป็นชุดคำสั่งที่เรียงต่อกันรูปแบบต่าง ๆ ที่สำคัญ สำหรับสร้างการเรียนรู้ให้เครื่องคอมพิวเตอร์เพื่อการทำนายค่าผลลัพธ์ที่เป็นตัวเลข หรือเพื่อการทำนายเพื่อจำแนกประเภทข้อมูลได้แก่ ตัวเลข ตัวอักษร ไฟล์ภาพ ไฟล์เสียง และไฟล์วิดีโอ หรือเพื่อการจัดกลุ่มข้อมูล หรือเพื่อการลดมิติให้ได้จำนวนองค์ประกอบที่เหมาะสม หรือเพื่อการเรียนรู้แบบกึ่งมีผู้สอนสำหรับการจำแนกประเภท หรือการเรียนรู้แบบเสริมกำลัง ด้วยวิธีการวนซ้ำค่าฟังก์ชันมูลค่าสำหรับกระบวนการตัดสินใจแบบมาร์คอฟ หรือด้วยวิธีมอนติคาร์โล

ที่ นักเรียน นิสิต นักศึกษา และผู้สนใจโดยทั่วไป สามารถศึกษาอ่านทำความเข้าใจ และเรียนรู้ด้วยตนเอง เพื่อประโยชน์ในการใช้ประกอบเรียน และเพื่อใช้สำหรับทำงานที่เกี่ยวกับ การสร้างการเรียนรู้ให้เครื่องคอมพิวเตอร์ หรือ Machine Learning ได้

ปวีณ โชคนุกูล

สารบัญ

บทที่	เรื่อง	หน้า
	ปกใน	ii
	ชื่อหนังสือ ภาษาไทยและภาษาอังกฤษ ชื่อผู้เขียน พิมพ์ครั้งที่ ราคา ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ และจัดทำโดย	iii
	คำนำ	iv
1.	บทนำ	1
2.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Linear regression	8
3.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Ridge regression	35
4.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Logistic regression	44
5.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม K-nearest neighbors	48
6.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Naïve bays	111
7.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Support vector machines	116
8.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Linear discriminant analysis	189
9.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Decision tree	200
10.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Random forest	219
11.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Xgboost	233
12.	การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Neural network	279
13.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม K-means clustering	424
14.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม Dbscan clustering	438

สารบัญ (ต่อ 1)

บทที่	เรื่อง	หน้า
15.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม Hierarchy clustering	463
16.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม Gaussian mixture clustering	478
17.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม การลดมิติเพื่อหาจำนวนองค์ประกอบที่เหมาะสม แบบ PCA	488
18.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม การลดมิติแบบ TSNE	532
19.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม การลดมิติเพื่อหาจำนวนองค์ประกอบที่เหมาะสม แบบ UMAP	546
20.	การเรียนรู้แบบไม่มีผู้สอน ด้วยอัลกอริธึม การลดมิติเพื่อหาจำนวนองค์ประกอบที่เหมาะสม แบบ Fast ica	560
21.	การเรียนรู้แบบกึ่งมีผู้สอน เพื่อการจำแนก	573
22.	การเรียนรู้แบบเสริมกำลัง ด้วยอัลกอริธึม Markov decision process (mdp) value iteration	585
23.	การเรียนรู้แบบเสริมกำลัง ด้วยอัลกอริธึม Monte carlo methods	592
24.	สรุป	601
	เอกสารอ้างอิง	603
	ดัชนี	604
	ประวัติผู้เขียน	608

สารบัญภาพ

ภาพที่	ชื่อภาพ	หน้า
1.1	ลักษณะหน้าเว็บ google colab	5
5.1	ลักษณะภาพของ ไฟล์ sample8.jpg สำหรับตัวอย่าง 5.3	110
7.1	ลักษณะภาพของไฟล์ sample6.jpg สำหรับตัวอย่างที่ 7.2	188
12.1	ลักษณะภาพของไฟล์ sample.jpg สำหรับตัวอย่างที่ 12.7	389
12.2	ลักษณะภาพของไฟล์ sample_sound2.mp3 สำหรับตัวอย่าง 12.8	408
12.3	ลักษณะภาพของไฟล์ sample1.mp4 สำหรับตัวอย่าง 12.9	421
15.1	แผนภาพเดนโดรแกรม สำหรับการวิเคราะห์เพื่อจัดกลุ่มข้อมูลแบบลำดับชั้น	477
17.1	กราฟสครี แสดงความแปรปรวนที่อธิบายได้สะสม และจำนวนองค์ประกอบ	499
17.2	กราฟสครี แสดงการวิเคราะห์ความแปรปรวนสะสม และจำนวนองค์ประกอบ	517

บทที่ 1. บทนำ

ในยุคที่ข้อมูลมีปริมาณอย่างมหาศาลและมีการเพิ่มขึ้นอย่างรวดเร็ว เทคโนโลยีที่สามารถเรียนรู้จากข้อมูลและสามารถนำความรู้นั้นไปใช้ประโยชน์ได้ จึงกลายเป็นหัวใจสำคัญของนวัตกรรมสมัยใหม่ “การเรียนรู้ของเครื่อง” หรือ Machine Learning เป็นหนึ่งในศาสตร์ที่ได้รับความสนใจอย่างแพร่หลาย ทั้งใน แวดวงวิชาการ อุตสาหกรรม และการพัฒนาเทคโนโลยีในชีวิตประจำวัน ไม่ว่าจะเป็นระบบแนะนำสินค้า การรู้จำภาพและเสียงพูด การพยากรณ์ข้อมูล การทำนายผลลัพธ์ การจำแนกข้อมูล หรือแม้แต่การวิเคราะห์ข้อมูลเชิงลึกเพื่อการตัดสินใจ

การเรียนรู้ของเครื่อง (Machine Learning) เป็นแขนงหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence: AI) ที่มุ่งเน้นการพัฒนา อัลกอริทึม หรือ ขั้นตอนของวิธีการแก้ปัญหาเรื่องใดเรื่องหนึ่งที่มีลักษณะเป็นชุดคำสั่งที่เรียงต่อกันรูปแบบต่าง ๆ ที่สามารถเรียนรู้รูปแบบ (Patterns) จากข้อมูลโดยไม่ต้องเขียนกฎเกณฑ์อย่างชัดเจนเหมือนการเขียนโปรแกรมแบบดั้งเดิม กล่าวอีกนัยหนึ่งคือ เครื่องคอมพิวเตอร์สามารถ “เรียนรู้จากประสบการณ์” และปรับปรุงประสิทธิภาพของตนเองได้เมื่อได้รับข้อมูลเพิ่มเติม ซึ่งแนวคิดนี้ได้เปลี่ยนวิธีการแก้ปัญหาในหลายสาขาอย่างสิ้นเชิง

ทั้งนี้ภาษาสำหรับสั่งงานคอมพิวเตอร์ได้แก่ ภาษาไพธอน (Python) เป็นเครื่องมือยอดนิยมสำหรับ การพัฒนาการเรียนรู้ของเครื่อง ด้วยเหตุผลหลายประการ ไม่ว่าจะเป็นความเรียบง่ายของไวยากรณ์ ความยืดหยุ่นในการใช้งาน และการมีไลบรารีที่ทรงพลังจำนวนมาก เช่น NumPy, Pandas, Scikit-learn, Keras, TensorFlow และ PyTorch ซึ่งช่วยให้นักพัฒนาและนักวิจัยสามารถสร้างแบบจำลอง (Models) ทดลองแนวคิด และนำไปใช้งานจริงได้อย่างรวดเร็ว

สำหรับรูปแบบของการเรียนรู้ของเครื่องจะแบ่งออกเป็น 4 ประเภทหลักได้แก่ (1) การเรียนรู้แบบมีผู้สอน (Supervised Learning), (2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning), (3) การเรียนรู้แบบกึ่งมีผู้สอน (Semi-supervised Learning) และ (4) การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) ซึ่งแต่ละแบบมีลักษณะวิธีการและการใช้งานแตกต่างกัน ดังนี้

1. Supervised Learning (การเรียนรู้แบบมีผู้สอน) คือการเรียนรู้ที่ใช้ “ข้อมูลที่มีคำตอบกำกับไว้แล้ว (Labeled Data)” กล่าวคือ ข้อมูลแต่ละตัวจะมีทั้ง “อินพุต (Input)” และ “เอาต์พุต (Output)” ที่ถูกต้องอยู่แล้ว มีหลักการทำงานคือ โมเดลจะเรียนรู้ความสัมพันธ์ระหว่างอินพุตและเอาต์พุต เช่น (1.1.1) อินพุตรูปภาพสัตว์ (1.1.2) เอาต์พุต “แมว” หรือ “สุนัข”โมเดลจะพยายามเรียนรู้ว่าอินพุตแบบไหนควรให้ผลลัพธ์อะไร ประเภทงานย่อยของการเรียนรู้แบบนี้ได้แก่ (1.2.1) Classification (การจำแนกประเภท) เช่น แยกอีเมลว่าเป็น Spam mail หรือไม่ (1.2.2) Regression (การพยากรณ์ค่า) เช่น ทำนายราคาบ้าน ตัวอย่างการใช้งานของการเรียนรู้ประเภทนี้ได้แก่ (1.3.1) การรู้จำใบหน้า (1.3.2) การพยากรณ์อากาศ (1.3.3) การทำนายผลลัพธ์ค่าตัวเลข (1.3.4) การทำนายเพื่อจำแนกประเภทข้อมูลได้แก่ ตัวเลข ตัวอักษร ไฟล์ภาพ ไฟล์เสียง และไฟล์วีโอ ข้อดีของการเรียนรู้ประเภทนี้ได้แก่ (1.4.1) มีความแม่นยำสูงเมื่อมีข้อมูลจำนวนมาก (1.4.2) เข้าใจและควบคุมได้ง่าย ส่วนข้อเสียของการเรียนรู้ประเภทนี้คือ ต้องใช้ข้อมูลที่มี label ซึ่งมักหายากและมีต้นทุนสูง

2. Unsupervised Learning (การเรียนรู้แบบไม่มีผู้สอน) คือการเรียนรู้จาก “ข้อมูลที่ไม่มีคำตอบกำกับ (Unlabeled Data)” โมเดลต้องค้นหารูปแบบหรือโครงสร้างในข้อมูลด้วยตัวเอง มีหลักการทำงานคือ โมเดลจะพยายาม “จัดกลุ่ม” หรือ “ค้นหาความสัมพันธ์” ในข้อมูล เช่น จัดกลุ่มลูกค้าตามพฤติกรรมการซื้อ เทคนิคหลักของการเรียนรู้ประเภทนี้ได้แก่ (2.1.1) Clustering (การจัดกลุ่ม)

เช่น แบ่งลูกค้าเป็นกลุ่ม และ (2.1.2) Dimensionality Reduction (ลดมิติข้อมูล) เช่น ลดจำนวนตัวแปรเพื่อให้วิเคราะห์ง่ายขึ้น ตัวอย่างการใช้งาน (2.2.1) การแบ่งกลุ่มลูกค้า (Customer Segmentation) (2.2.2) ระบบแนะนำสินค้า (2.2.3) การตรวจจับความผิดปกติ (Anomaly Detection) ข้อดีของการเรียนรู้ประเภทนี้ ได้แก่ (2.3.1) ไม่ต้องใช้ Label (2.3.2) เหมาะกับข้อมูลขนาดใหญ่ที่ยังไม่ได้จัดหมวดหมู่ ส่วนข้อเสียของการเรียนรู้ประเภทนี้ได้แก่ (2.4.1) ผลลัพธ์อาจตีความยาก (2.4.2) ความแม่นยำไม่แน่นอน

3. Semi-supervised Learning (การเรียนรู้แบบกึ่งมีผู้สอน) คือ การเรียนรู้ที่เป็นการผสมระหว่าง การเรียนรู้แบบมีผู้สอน (Supervised) และ การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) โดยใช้ข้อมูลทั้งแบบมี Label และไม่มี Label มีหลักการทำงานคือ (3.1.1) ใช้ข้อมูลที่มี Label จำนวนเล็กน้อย (3.1.2) ใช้ข้อมูลที่ไม่มี Label จำนวนมากช่วยเสริมการเรียนรู้ ตัวอย่างเช่น มีภาพทั้งหมด 1,000 รูป โดยรูป 100 แรกเป็นรูปมี Label และอีก 900 รูปเป็นรูปที่ไม่มี Label ซึ่งใช้รูปทั้งหมดนี้เพื่อฝึกโมเดล ตัวอย่าง การใช้งานของการเรียนรู้ประเภทนี้ ได้แก่ (3.2.1) การทำนายเพื่อจำแนกประเภทของข้อมูล (3.2.2) การรู้จำภาพ (Image Recognition) (3.2.3) การประมวลผลภาษา (NLP) และ (3.2.4) งานที่ Label มีราคาแพง ข้อดีของการเรียนรู้ประเภทนี้ (3.3.1) ลดต้นทุนการทำ Label (3.3.2) ให้ประสิทธิภาพดีกว่าใช้ข้อมูล Labeled อย่างเดียว (ในบางกรณี) ส่วนข้อเสียของการเรียนรู้ประเภทนี้ (3.4.1) มีความซับซ้อนในการออกแบบ (3.4.2) ถ้าข้อมูลไม่มี Label มีคุณภาพไม่ดี อาจทำให้โมเดลเรียนรู้ผิด

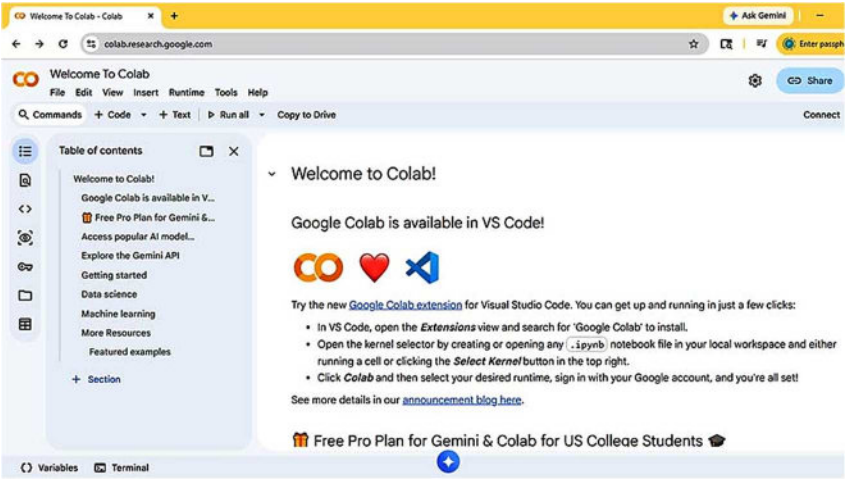
4. Reinforcement Learning (การเรียนรู้แบบเสริมกำลัง) คือ การเรียนรู้ผ่าน “การลองผิดลองถูก (Trial and Error)” โดยมีตัวแทน (Agent) ที่เรียนรู้จากการโต้ตอบกับสภาพแวดล้อม มีหลักการทำงานคือ (4.1.1) ตัวแทน (Agent) ทำการกระทำ (Action) (4.1.2) ได้รับผลตอบแทน (Reward) หรือบทลงโทษ (Penalty) และ (4.1.3) ปรับปรุงกลยุทธ์เพื่อให้ได้รางวัลมากที่สุด หรือทำการส่งอินพุตออกไปแล้วตรวจสอบเอาต์พุตที่ได้รับว่า เป็นบวกหรือเป็นลบ

จากนั้นตัดสินใจให้สอดคล้องกับเอาต์พุตดังกล่าว หรือเอาต์พุตขณะใดขึ้นกับอินพุตขณะนั้น ตัวอย่างเช่น การฝึกสุนัข (4.2.1) กรณีทำถูก ทำการให้รางวัล (4.2.2) กรณีทำผิด ทำการไม่ให้รางวัล ตัวอย่างการใช้งานของการเรียนรู้แบบนี้ได้แก่ (4.3.1) การเรียนรู้ด้วยวิธีการวนซ้ำค่าฟังก์ชันมูลค่าสำหรับกระบวนการตัดสินใจแบบมาร์คอฟ (4.3.2) การเรียนรู้ด้วยวิธีมอนติคาร์โล (4.3.3) เกม เช่น AI เล่นหมากรุก (4.3.4) รถยนต์ไร้คนขับ (4.3.5) หุ่นยนต์ควบคุมการเคลื่อนไหว สำหรับแนวคิดสำคัญของการเรียนรู้แบบนี้ได้แก่ (4.4.1) Policy (นโยบายการตัดสินใจ) (4.4.2) Reward (รางวัล) (4.4.3) Environment (สภาพแวดล้อม) ข้อดีของการเรียนรู้แบบนี้ได้ (4.5.1) เรียนรู้พฤติกรรมที่ซับซ้อนได้ดี (4.5.2) เหมาะกับปัญหาที่ต้องตัดสินใจต่อเนื่อง ส่วนข้อเสียของการเรียนรู้แบบนี้ได้แก่ (4.6.1) ใช้เวลาฝึกนาน (4.6.2) ต้องออกแบบ Reward ให้ดี

สรุปทั้ง 4 วิธีเป็นแนวทางสำคัญในการเรียนรู้ของเครื่อง ซึ่งจะถูกเลือกใช้แตกต่างกันตามลักษณะของข้อมูลและปัญหา โดยที่ (1) ถ้ามีข้อมูลพร้อมคำตอบจะใช้การเรียนรู้แบบ Supervised Learning (การเรียนรู้แบบมีผู้สอน) (2) ถ้าไม่มีคำตอบ จะใช้การเรียนรู้แบบ Unsupervised Learning (การเรียนรู้แบบไม่มีผู้สอน) (3) ถ้ามี label น้อย จะใช้การเรียนรู้แบบ Semi-supervised Learning (การเรียนรู้แบบกึ่งมีผู้สอน) และ (4) ถ้าต้องเรียนรู้จากการโต้ตอบ จะใช้การเรียนรู้แบบ Reinforcement Learning (การเรียนรู้แบบเสริมกำลัง)

ประเภทการเรียนรู้	การใช้ Label	ลักษณะเด่น
Supervised	ใช้	ทำนายผลจากตัวอย่างที่รู้คำตอบ
Unsupervised	ไม่ใช้	ค้นหารูปแบบในข้อมูล
Semi-supervised	ใช้ + ไม่ใช้	ผสมข้อดีของสองแบบ
Reinforcement	ไม่ใช้ label โดยตรง	เรียนรู้จากรางวัล/บทลงโทษ

ทั้งนี้สำหรับ เนื้อหาในหนังสือเล่มนี้ จะใช้การเขียนคำสั่งภาษาไพธอน แบบออนไลน์ และดูผลลัพธ์การทำคำสั่ง บนเว็บไซต์ google colab https://colab.research.google.com/



ภาพที่ 1.1 ลักษณะหน้าเว็บ google colab https://colab.research.google.com/

สภาวะ Overfitting และ Underfitting

1. ภาวะการเรียนรู้มากเกินไปของแบบจำลอง (Overfitting) คือ สถานการณ์ที่โมเดลหรือแบบจำลองเรียนรู้ข้อมูลสำหรับฝึกสอน (Training data) มากเกินไปจนจำรายละเอียดเล็ก ๆ หรือ สิ่งรบกวน (Noise) ในข้อมูลนั้นแทนที่จะเรียนรู้รูปแบบทั่วไป (General pattern) ทำให้ทำนายข้อมูลใหม่พลาด หรือภาวะที่แบบจำลองจดจำข้อมูลเก่ามากเกินไป จนทำให้ทำนายข้อมูลใหม่ได้ไม่ดี หรือ เรียนรู้มากเกินไปโดยจำได้แต่สิ่งรบกวน หรือแม่นมากกับข้อมูลที่เคยเห็นแต่เมื่อเจอข้อมูลใหม่จะทำนายผลลัพธ์พลาด

เกิดขึ้นได้อย่างไรสาเหตุหลัก ๆ ได้แก่ (1) โมเดลหรือแบบจำลองมีความซับซ้อนเกินไป เช่นต้นไม้ใน Decision Tree ลึกเกินไปหรือมี Parameter เป็นจำนวนมาก (2) ข้อมูลมี Noise สูง โดยโมเดลไปจำ “ความบังเอิญ” ในข้อมูล (3) ข้อมูลน้อยเกินไป คือ ไม่มีข้อมูลพอให้เรียนรู้รูปแบบข้อมูล (Pattern) จริง

ลักษณะของ Overfitting จะได้แก่ มีค่า Accuracy บน Training data สูงมาก และมีค่า Accuracy บน Test data ต่ำ ซึ่งหมายถึงโมเดลหรือแบบจำลองไม่สามารถ Generalize ได้ หรือไม่สามารถทำนายข้อมูลใหม่ได้

โดยวิธีแก้ Overfitting ได้แก่ (1) ลดความซับซ้อนของโมเดล ด้วยการตัดกิ่ง (Pruning) ใน Decision Tree หรือ ลดจำนวน Layer/Parameter (2) ใช้ข้อมูลเพิ่ม ด้วยการเพิ่มชุดข้อมูล (Dataset) หรือทำ Data augmentation

ส่วน **2. ภาวะการเรียนรู้น้อยเกินไปของแบบจำลอง (Underfitting)** คือ ภาวะที่มีขนาดการเรียนรู้ที่น้อยเกินไปทำให้จับรูปแบบที่แท้จริงของข้อมูลไม่ได้ ทำให้ทำนายได้ไม่ดีหรือทำนายพลาดทั้งกับข้อมูลที่เคยเห็น (Training) และกับข้อมูลใหม่ (Test) หรือเป็นภาวะที่โมเดลหรือแบบจำลอง

ยังไม่เข้าใจข้อมูล จนทำให้ผลการทำนายเป็นไปในทางที่ไม่ดีทั้งกับ ข้อมูลเก่าและข้อมูลใหม่

ลักษณะของ Underfitting ได้แก่ (1) Accuracy บน training data มีค่าต่ำ และ (2) Accuracy บน Test data มีค่าต่ำ ซึ่งหมายถึงโมเดลยังไม่สามารถเรียนรู้ความสัมพันธ์พื้นฐานของข้อมูลได้

สาเหตุที่ทำให้เกิด ภาวะ Underfitting ได้แก่ (1) โมเดลง่ายเกินไป เช่น ใช้เส้นตรงไป Fit ข้อมูลที่เป็นโค้ง หรือ Decision Tree ตื้นเกินไป (2) จำนวนตัวแปรอิสระหรือ Feature ไม่เพียงพอ หรือมีข้อมูลที่ใช้ไม่ครอบคลุมหรือไม่เกี่ยวข้อง (3) ทำการฝึกสอนไม่เพียงพอหรือเทรน (Train) ไม่พอ เช่น จำนวน Epoch น้อยเกินไป หรือโมเดลยังเรียนรู้ไม่จบ

วิธีแก้ ภาวะ Underfitting ได้แก่ (1) ทำการเพิ่มความซับซ้อนของโมเดลหรือแบบจำลอง โดยใช้โมเดลที่ยืดหยุ่นมากขึ้นเช่น เพิ่มความลึก (Depth) ของ Decision Tree หรือ เพิ่มจำนวน Layer ใน Neural network (2) ทำการเพิ่มจำนวนตัวแปรอิสระหรือ Feature โดยการสร้างตัวแปรอิสระหรือ Feature ใหม่ (Feature engineering) (3) ทำการฝึกสอนให้ยาวนานขึ้นหรือ เทรน (Train) ให้ยาวนานขึ้น เช่น เพิ่มจำนวนของ epoch

ความสามารถในการสรุปทั่วไป (Generalize) คือ ความสามารถของโมเดลที่สามารถทำการทำนายข้อมูลตัวใหม่ได้ดี

นอยส์ (Noise) คือ ค่ารบกวน หรือจุดรบกวน หรือสิ่งรบกวน

ค่าผิดปกติ (Outlier) คือ จุดข้อมูลที่มีค่าห่างจากข้อมูลส่วนใหญ่ของชุดข้อมูลอย่างชัดเจน หรือจุดข้อมูลที่แตกต่างจากกลุ่มหลักอย่างชัดเจน หรือเป็นข้อมูลที่มีค่าไม่เหมือนข้อมูลอื่น โดยมีความแตกต่างจากข้อมูลปกติอย่างชัดเจน

บทที่ 2. การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม

Linear regression

การวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย (Simple linear regression) คือ เทคนิคทางสถิติที่ใช้ทำนายผลลัพธ์สำหรับ กรณีที่ข้อมูลตัวอย่าง มีตัวแปรอิสระ (x : Independence variable) หรือ ฟีเจอร์ (Feature) เพียง 1 ตัว มีค่าเป็นตัวเลข และ มีตัวแปรตาม (y : Dependence variable) หรือ เป้าหมาย หรือผลลัพธ์ (Target) เพียง 1 ตัว มีค่าเป็นตัวเลข โดยทั้งตัวแปรอิสระและตัวแปรตาม มีความสัมพันธ์กันแบบ เชิงบวก หรือ เชิงลบ อย่างใดอย่างหนึ่ง

สมการการถดถอยเชิงเส้นอย่างง่ายจะมีลักษณะได้แก่

$$y = \beta_0 + \beta_1 x$$

เมื่อ

β_0 คือ จุดตัดแกน y หรือ y-intercept

β_1 คือ ค่าสัมประสิทธิ์ หรือ Coefficient ของตัวแปรอิสระ

x คือ ตัวแปรอิสระ หรือ ฟีเจอร์

การวิเคราะห์การถดถอยเชิงเส้นพหุคูณ (Multiple linear regression: MLR) คือ เทคนิคทางสถิติที่ใช้ทำนายผลลัพธ์สำหรับ กรณีที่ข้อมูลตัวอย่าง มีตัวแปรอิสระ (x : Independence variable) หรือ ฟีเจอร์ (Feature) มากกว่า 1 ตัว เช่น x_1, x_2, x_3, x_4, x_5 ,..., x_n และ มีตัวแปรตาม (y : Dependence variable) หรือ เป้าหมาย หรือผลลัพธ์ (Target) เพียง 1 ตัว มีค่าเป็นตัวเลข โดยทั้งตัวแปรอิสระและตัวแปรตาม โดยทั้งตัวแปรอิสระและตัวแปรตาม มีความสัมพันธ์กันแบบ เชิงบวก หรือ เชิงลบ อย่างใดอย่างหนึ่ง

สมการการถดถอยเชิงเส้นพหุคูณจะมีลักษณะได้แก่

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n$$

เมื่อ

β_0 คือ จุดตัดแกน y หรือ y-intercept

$\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ คือ ค่าสัมประสิทธิ์ หรือ Coefficient ของตัวแปรอิสระ

$x_1, x_2, x_3, \ldots, x_n$ คือ ตัวแปรอิสระ หรือ ฟีเจอร์

โดย R^2 หรือ R-Squared คือ ค่าที่แสดงระดับความใกล้เคียงของผลการทำนายกับข้อมูลจริง หรือเป็นค่าที่บอกระดับความถูกต้องสำหรับการทำนายผลลัพธ์ ค่านี้เป็นตัวบ่งชี้ว่าแบบจำลองนั้น ๆ มีความน่าเชื่อถือเพียงใด

ตัวอย่างที่ 2.1 คำสั่งการวิเคราะห์การถดถอยเชิงเส้นอย่างง่าย

```
from sklearn.linear_model import LinearRegression
import numpy as np

x = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
x = np.array(x).reshape(-1,1)
y = [21, 23, 27, 29, 31, 35, 37, 39, 41, 43]

model = LinearRegression()
model.fit(x, y)
score = model.score(x, y)

y_10 = model.predict([[10]])
print('x=10, y =','{0:.4f}'.format(y_10[0]))

y_14 = model.predict([[14]])
print('x=14, y =', '{0:.4f}'.format(y_14[0]))

print('Intercept =', '{0:.4f}'.format(model.intercept_))
print('Coefficient =', '{0:.4f}'.format(model.coef_[0]))
interc = '{0:.4f}'.format(model.intercept_)
coef = '{0:.4f}'.format(model.coef_[0])
print(f'Predict equation is: y = {interc} + ({coef})x')
print('R-Squared: ', '{0:.6f}'.format(score))
accur1 = score * 100
accur2 = '{0:.4f}'.format(accur1)
print(f'Accuracy level is {accur2} %')
```

ลักษณะของข้อมูลที่ใช้ในตัวอย่างนี้

x : 1, 3, 5, 7, 9, 11, 13, 15, 17, 19

y : 21, 23, 27, 29, 31, 35, 37, 39, 41, 43

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 2.1

x=10, y = 32.6000

x=14, y = 37.5939

Intercept = 20.1152

Coefficient = 1.2485

Predict equation is: y = 20.1152 + (1.2485) x

R-Squared: 0.992237

Accuracy level is 99.2237 %

ตัวอย่างที่ 2.2 คำสั่งการวิเคราะห์การถดถอยเชิงเส้นพหุคูณ

```
import numpy as np
from sklearn.linear_model import LinearRegression

d = [ [50, 33, 30, 205], [55, 35, 45, 220],
      [64, 40, 45, 230], [64, 45, 50, 235],
      [66, 46, 55, 240], [67, 47, 55, 245],
      [68, 48, 61, 250], [67, 48, 60, 255]]
d  = np.array(d)
x = d[:, 0:3]
y = d[:, 3]

model = LinearRegression()
model.fit(x, y)
r_square = model.score(x, y)

print(   )
ic =   '{:.4f}'.format(model.intercept_)
print('Intercept is: ',ic)
coef = model.coef_
coef0 = '{:.4f}'.format(model.coef_[0])
coef1 = '{:.4f}'.format(model.coef_[1])
coef2 = '{:.4f}'.format(model.coef_[2])
print('Coefficient is: ',coef)
print('Enter your values are want to predict !')
x1 = float(input('Enter value x1: '))
x2 = float(input('Enter value x2: '))
x3 = float(input('Enter value x3: '))
x_predict = [[x1, x2, x3]]
y_predict = model.predict(x_predict)
result = "{:.4f}".format(y_predict[0])
print(f'Predict equation is: y = {ic} + ({coef0})x1 + ({coef1})x2 + ({coef2})x3')
print('R-square is: ', '{:.4f}'.format( r_square))
print('Predict value is: ',result)
```

ลักษณะของข้อมูลที่ใช้ในตัวอย่างนี้

x1	x2	x3	y
50	33	30	205
55	35	45	220
64	40	45	230
64	45	50	235
66	46	55	240
67	47	55	245
68	48	61	250
67	48	60	255

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 2.2

Intercept is: 130.8020

Coefficient is: [0.36426077 0.84444382 0.90346615]

Enter your values are want to predict !

Enter value x1: 1

Enter value x2: 1

Enter value x3: 0

Predict equation is: y = 130.8020 + (0.3643) x1 + (0.8444) x2 + (0.9035) x3

R-square is: 0.9747

Predict value is: 132.0107

การวิเคราะห์การถดถอยพหุแปร หรือการวิเคราะห์การถดถอยพหุคูณหลายตัวแปรตาม (Multivariate regression) คือเทคนิคทางสถิติที่ใช้วิเคราะห์ความสัมพันธ์ระหว่างตัวแปรอิสระหลายตัว (Independent variables) กับตัวแปรตามมากกว่าหนึ่งตัว (Multiple dependent variables) หรืออาจกล่าวได้ง่าย ๆ ว่า เป็นการทำนายผลลัพธ์ หลายตัวพร้อมกัน หรือ Simple regression ใช้กับ ตัวแปรอิสระ 1 ตัว และตัวแปรตาม 1 ตัว ส่วน Multiple regression ใช้กับ ตัวแปรอิสระหลายตัว และ ตัวแปรตาม 1 ตัว ขณะที่ Multivariate regression ใช้กับ ตัวแปรอิสระหลายตัว และ ตัวแปรตาม “หลายตัว” โดยใช้เมื่อต้องการ วิเคราะห์หลายผลลัพธ์พร้อมกัน

รูปแบบทางคณิตศาสตร์การวิเคราะห์การถดถอยพหุคูณหลายตัวแปรตามได้แก่

$Y = XB + E$

เมื่อ

Y คือ เมทริกซ์ของตัวแปรตาม

X คือ เมทริกซ์ของตัวแปรอิสระ

B คือ เมทริกซ์ของค่าสัมประสิทธิ์

E คือ ค่าความคลาดเคลื่อน (Error)

ตัวอย่างที่ 2.3 คำสั่งการวิเคราะห์การถดถอยพหุคูณหลายตัวแปรตาม (1)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import os

# 1. กำหนดชื่อไฟล์
file_name = '/content/regression_4input_2output.csv'

# 2. โหลดข้อมูลจากไฟล์ CSV
print("กำลังอ่านข้อมูลและเตรียมการฝึกสอนโมเดล...")
df = pd.read_csv(file_name)

# 3. แยกคอลัมน์ Input (X) และ Output (y)
X = df[['v1', 'v2', 'v3', 'v4']]
y  = df[['output1', 'output2']]

# 4. แบ่งข้อมูลออกเป็นชุดฝึกสอน (Train 80%) และชุดทดสอบเพื่อหาความแม่นยำ (Test 20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

ตัวอย่างที่ 2.3 คำสั่งการวิเคราะห์การถดถอยพหุคูณหลายตัวแปรตาม (1) ต่อ 1

```
# 5. สร้างและฝึกสอนโมเดล Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)

# 6. ประเมินความแม่นยำของโมเดล (ใช้ข้อมูล Test Set)
# โดยจะใช้ค่า R-squared (คะแนนสูงสุดคือ 1.0 หรือ 100%)
y_pred_test = model.predict(X_test)
r2_out1 = r2_score(y_test['output1'], y_pred_test[:, 0])
r2_out2 = r2_score(y_test['output2'], y_pred_test[:, 1])

print("\n--- ผลการประเมินความแม่นยำของโมเดล (R-squared Score) ---")
print(f"ความแม่นยำสำหรับ Output 1: {r2_out1:.4f} (คิดเป็น {(r2_out1*100):.2f}%)")
print(f"ความแม่นยำสำหรับ Output 2: {r2_out2:.4f} (คิดเป็น {(r2_out2*100):.2f}%)")

# 7. กำหนดข้อมูลใหม่ จำนวน 1 ชุดเพื่อทำนายผล (s1, s2, s3, s4)
# **คุณสามารถเปลี่ยนค่าตัวเลขตรงนี้เป็นค่าข้อมูลจริงของคุณได้เลย**
print()
print('Pleased enter integer number from 0 to 15, Not exceed 15 ')
var1 = float(input('Enter value var1: '))
var2 = float(input('Enter value var2: '))
var3 = float(input('Enter value var3: '))
var4 = float(input('Enter value var4: '))
s1 = var1
s2 = var2
s3 = var3
s4 = var4

# สร้าง DataFrame สำหรับข้อมูลที่ต้องการทำนาย (ต้องให้ชื่อคอลัมน์ตรงกับตอนฝึกโมเดล)
new_data = pd.DataFrame({
    'v1': [s1],
    'v2': [s2],
    'v3': [s3],
    'v4': [s4]
})
```

ตัวอย่างที่ 2.3 คำสั่งการวิเคราะห์การถดถอยพหุคูณหลายตัวแปรตาม (1) ต่อ 2

```
# 8. ทำนายผลและแสดงค่า
prediction = model.predict(new_data)
predicted_output1 = prediction[0][0]
predicted_output2 = prediction[0][1]

print("\n--- ผลการทำนายข้อมูลใหม่ ---")
print(f"ข้อมูลที่ป้อนเข้า (Input) -> v1={s1}, v2={s2}, v3={s3}, v4={s4}")
print(f"ผลการทำนาย Output 1   -> {predicted_output1:.4f}")
print(f"ผลการทำนาย Output 2   -> {predicted_output2:.4f}")
```

ลักษณะของข้อมูลที่ใช้กับตัวอย่าง 2.4 ไฟล์ regression_4input_2output.csv

v1,v2,v3,v4,output1,output2

0,0,0,0,0,0

1,1,1,1,6,8

1,1,1,3,7,9

1,1,1,5,10,12

1,1,1,7,12,14

1,1,1,9,14,16

1,1,1,11,16,18

1,1,1,13,18,20

1,1,1,15,20,22

1,1,3,3,10,12

1,1,5,5,14,16

1,1,7,7,18,20

1,1,9,9,22,24

1,1,11,11,26,28

1,1,13,13,30,32

1,1,15,15,34,36

1,3,3,3,12,14

1,5,5,5,18,20

1,7,7,7,24,26

1,9,9,9,30,32

1,11,11,11,36,38

1,13,13,13,42,44

1,15,15,15,48,50

3,3,3,3,14,16

3,3,3,1,12,14

3,3,3,5,16,18

3,3,3,7,18,20

3,3,3,9,20,22

3,3,3,11,22,24

3,3,3,13,24,26

3,3,3,15,26,28

3,3,1,1,10,12

3,3,5,5,18,20

3,3,7,7,22,24

3,3,11,11,30,32

3,3,13,13,34,36

3,3,15,15,38,40

3,1,1,1,8,10

3,5,5,5,20,22

3,7,7,7,26,28

3,9,9,9,32,34

3,11,11,11,38,40

3,13,13,13,44,46

3,15,15,15,50,52

5,5,5,5,22,24

5,5,5,1,18,20

5,5,5,3,20,22

5,5,5,7,24,26

5,5,5,9,26,28

5,5,5,11,28,30

5,5,5,13,30,32

5,5,5,15,32,34

5,5,1,1,14,16

5,5,3,3,18,20

5,5,7,7,26,28

5,5,9,9,30,32

5,5,11,11,34,36

5,5,13,13,38,40

5,5,15,15,42,44

5,1,1,1,10,12

5,3,3,3,16,18

5,7,7,7,28,30

5,9,9,9,34,36

5,11,11,11,40,42

5,13,13,13,46,48

5,15,15,15,52,54

7,7,7,7,30,32

7,7,7,1,24,26

7,7,7,3,26,28

7,7,7,5,28,30

7,7,7,9,32,34

7,7,7,11,34,36

7,7,7,13,36,38

7,7,7,15,38,40

7,7,1,1,18,20

7,7,3,3,22,24

7,7,5,5,26,28

7,7,9,9,34,36

7,7,11,11,38,40

7,7,13,13,42,44

7,7,15,15,46,48

7,1,1,1,12,14

7,3,3,3,18,20

7,5,5,5,24,26

7,9,9,9,36,38

7,11,11,11,42,44

7,13,13,13,48,50

7,15,15,15,54,56

9,9,9,9,38,40

9,9,9,1,30,32

9,9,9,3,32,34

9,9,9,5,34,36

9,9,9,7,36,38

9,9,9,11,40,42

9,9,9,13,42,44

9,9,9,15,44,46

9,9,1,1,22,24

9,9,3,3,26,28

9,9,5,5,30,32

9,9,7,7,34,36

9,9,11,11,42,44

9,9,13,13,46,48

9,9,15,15,50,52

9,1,1,1,14,16

9,3,3,3,20,22

9,5,5,5,26,28

9,7,7,7,32,34

9,11,11,11,44,46

9,13,13,13,50,52

9,15,15,15,56,58

11,11,11,11,46,48

11,11,11,1,36,38

11,11,11,3,38,40

11,11,11,5,40,42

11,11,11,7,42,44

11,11,11,9,44,46

11,11,11,13,48,50

11,11,11,15,50,52

11,11,1,1,26,28

11,11,3,3,30,32

11,11,5,5,34,36

11,11,7,7,38,40

11,11,9,9,42,44

11,11,13,13,50,52

11,11,15,15,54,56

13,13,13,13,54,56

13,13,13,1,42,44

13,13,13,3,44,46

13,13,13,5,46,48

13,13,13,7,48,50

13,13,13,9,50,52

13,13,13,11,52,54

13,13,13,15,56,58

13,13,1,1,30,32

13,13,3,3,34,36

13,13,5,5,38,40

13,13,7,7,42,44

13,13,9,9,46,48

13,13,11,11,50,52

13,13,15,15,58,60

13,1,1,1,18,20

13,3,3,3,24,26

13,5,5,5,30,32

13,7,7,7,36,38

13,9,9,9,42,44

13,11,11,11,48,50

13,15,15,15,60,62

15,15,15,15,62,64

15,15,15,1,48,50

15,15,15,3,50,52

15,15,15,5,52,54

15,15,15,7,54,56

15,15,15,9,56,58

15,15,15,11,58,60

15,15,15,13,60,62

15,15,1,1,34,36

15,15,3,3,38,40

15,15,5,5,42,44

15,15,7,7,46,48

15,15,9,9,50,52

15,15,11,11,54,56

15,15,13,13,58,60

15,1,1,1,20,22

15,3,3,3,26,28

15,7,7,7,38,40

15,9,9,9,44,46

15,11,11,11,50,52

15,13,13,13,56,58

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 2.3

กำลังอ่านข้อมูลและเตรียมการฝึกสอนโมเดล...

--- ผลการประเมินความแม่นยำของโมเดล (R-squared Score) ---

ความแม่นยำสำหรับ Output 1: 1.0000 (คิดเป็น 100.00%)

ความแม่นยำสำหรับ Output 2: 0.9999 (คิดเป็น 99.99%)

Pleased enter integer number from 0 to 15, Not exceed 15

Enter value var1: 5

Enter value var2: 5

Enter value var3: 5

Enter value var4: 5

--- ผลการทำนายข้อมูลใหม่ ---

ข้อมูลที่ป้อนเข้า (Input) -> v1=5.0, v2=5.0, v3=5.0, v4=5.0

ผลการทำนาย Output 1 -> 21.9249

ผลการทำนาย Output 2 -> 23.8715

ตัวอย่างที่ 2.4 คำสั่งการวิเคราะห์การถดถอยพหุคูณหลายตัวแปรตาม (2)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import os

def run_regression_analysis(csv_file, new_sample):
    # 1. โหลดข้อมูล
    df = pd.read_csv(csv_file)

    # 2. แยก Features (X) และ Targets (y)
    X = df[['v1', 'v2', 'v3', 'v4']]
     y = df[['output1', 'output2', 'output3']]

    # 3. แบ่งข้อมูลสำหรับ Training และ Testing
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)

    # 4. สร้างและฝึกโมเดล Multivariate Linear Regression
    model = LinearRegression()
    model.fit(X_train, y_train)

    # 5. ทำนายค่าจาก Test set เพื่อหาความแม่นยำ
    y_pred = model.predict(X_test)
```

ตัวอย่างที่ 2.4 คำสั่งการวิเคราะห์การถดถอยพหุคูณหลายตัวแปรตาม (2) ต่อ 1

```
    # 6. รายงานค่าความแม่นยำ (Metrics)
    print("=== รายงานค่าความแม่นยำ (Accuracy Metrics) ===")
    outputs = ['output1', 'output2', 'output3']
    for i, col_name in enumerate(outputs):
        r2 = r2_score(y_test.iloc[:, i], y_pred[:, i])
        mse = mean_squared_error(y_test.iloc[:, i], y_pred[:, i])
        print(f"[{col_name}]")
        print(f"  - R-squared (ความแม่นยำ): {r2:.4f}")
        print(f"  - Mean Squared Error: {mse:.4f}")
     print("-" * 45)

  # 7. ทำนายค่าสำหรับข้อมูลชุดใหม่ (s1, s2, s3, s4)
  sample_df = pd.DataFrame([new_sample], columns=['v1', 'v2', 'v3', 'v4'])
  prediction = model.predict(sample_df)

  print("\n=== ผลการทำนายสำหรับข้อมูลตัวอย่างใหม่ ===")
  print(f"Input: v1={new_sample[0]}, v2={new_sample[1]},
        v3={new_sample[2]}, v4={new_sample[3]}")
  print(f"ทำนาย Output 1: {prediction[0][0]:.4f}")
  print(f"ทำนาย Output 2: {prediction[0][1]:.4f}")
  print(f"ทำนาย Output 3: {prediction[0][2]:.4f}")

if __name__ == "__main__":
    filename = '/content/regression_4input_3output.csv'

    # กำหนดค่าตัวอย่าง s1, s2, s3, s4 ที่ต้องการทำนาย
    # ตัวอย่าง: s1=5, s2=5, s3=5, s4=5
    s_values = [5, 5, 5, 5]

    run_regression_analysis(filename, s_values)
```

ลักษณะของข้อมูลที่ใช้กับตัวอย่าง 2.5 ไฟล์ regression_4input_3output.csv

v1,v2,v3,v4,output1,output2,output3

0,0,0,0,0,0,0

1,1,1,1,6,8,10

1,1,1,3,7,9,11

1,1,1,5,10,12,14

1,1,1,7,12,14,16

1,1,1,9,14,16,18

1,1,1,11,16,18,20

1,1,1,13,18,20,22

1,1,1,15,20,22,24

1,1,3,3,10,12,14

1,1,5,5,14,16,18

1,1,7,7,18,20,22

1,1,9,9,22,24,26

1,1,11,11,26,28,30

1,1,13,13,30,32,34

1,1,15,15,34,36,38

1,3,3,3,12,14,16

1,5,5,5,18,20,22

1,7,7,7,24,26,28

1,9,9,9,30,32,34

1,11,11,11,36,38,40

1,13,13,13,42,44,46

1,15,15,15,48,50,52

3,3,3,3,14,16,18

3,3,3,1,12,14,16

3,3,3,5,16,18,20

3,3,3,7,18,20,22

3,3,3,9,20,22,24

3,3,3,11,22,24,26

3,3,3,13,24,26,28

3,3,3,15,26,28,30

3,3,1,1,10,12,14

3,3,5,5,18,20,22

3,3,7,7,22,24,26

3,3,11,11,30,32,34

3,3,13,13,34,36,38

3,3,15,15,38,40,42

3,1,1,1,8,10,12

3,5,5,5,20,22,24

3,7,7,7,26,28,30

3,9,9,9,32,34,36

3,11,11,11,38,40,42

3,13,13,13,44,46,48

3,15,15,15,50,52,54

5,5,5,5,22,24,26

5,5,5,1,18,20,22

5,5,5,3,20,22,24

5,5,5,7,24,26,28

5,5,5,9,26,28,30

5,5,5,11,28,30,32

5,5,5,13,30,32,34

5,5,5,15,32,34,36

5,5,1,1,14,16,18

5,5,3,3,18,20,22

5,5,7,7,26,28,30

5,5,9,9,30,32,34

5,5,11,11,34,36,38

5,5,13,13,38,40,42

5,5,15,15,42,44,46

5,1,1,1,10,12,14

5,3,3,3,16,18,20

5,7,7,7,28,30,32

5,9,9,9,34,36,38

5,11,11,11,40,42,44

5,13,13,13,46,48,50

5,15,15,15,52,54,56

7,7,7,7,30,32,34

7,7,7,1,24,26,28

7,7,7,3,26,28,30

7,7,7,5,28,30,32

7,7,7,9,32,34,36

7,7,7,11,34,36,38

7,7,7,13,36,38,40

7,7,7,15,38,40,42

7,7,1,1,18,20,22

7,7,3,3,22,24,26

7,7,5,5,26,28,30

7,7,9,9,34,36,38

7,7,11,11,38,40,42

7,7,13,13,42,44,46

7,7,15,15,46,48,50

7,1,1,1,12,14,16

7,3,3,3,18,20,22

7,5,5,5,24,26,28

7,9,9,9,36,38,40

7,11,11,11,42,44,46

7,13,13,13,48,50,52

7,15,15,15,54,56,58

9,9,9,9,38,40,42

9,9,9,1,30,32,34

9,9,9,3,32,34,36

9,9,9,5,34,36,38

9,9,9,7,36,38,40

9,9,9,11,40,42,44

9,9,9,13,42,44,46

9,9,9,15,44,46,48

9,9,1,1,22,24,26

9,9,3,3,26,28,30

9,9,5,5,30,32,34

9,9,7,7,34,36,38

9,9,11,11,42,44,46

9,9,13,13,46,48,50

9,9,15,15,50,52,54

9,1,1,1,14,16,18

9,3,3,3,20,22,24

9,5,5,5,26,28,30

9,7,7,7,32,34,36

9,11,11,11,44,46,48

9,13,13,13,50,52,54

9,15,15,15,56,58,60

11,11,11,11,46,48,50

11,11,11,1,36,38,40

11,11,11,3,38,40,42

11,11,11,5,40,42,44

11,11,11,7,42,44,46

11,11,11,9,44,46,48

11,11,11,13,48,50,52

11,11,11,15,50,52,54

11,11,1,1,26,28,30

11,11,3,3,30,32,34

11,11,5,5,34,36,38

11,11,7,7,38,40,42

11,11,9,9,42,44,46

11,11,13,13,50,52,54

11,11,15,15,54,56,58

13,13,13,13,54,56,58

13,13,13,1,42,44,46

13,13,13,3,44,46,48

13,13,13,5,46,48,50

13,13,13,7,48,50,52

13,13,13,9,50,52,54

13,13,13,11,52,54,56

13,13,13,15,56,58,60

13,13,1,1,30,32,34

13,13,3,3,34,36,38

13,13,5,5,38,40,42

13,13,7,7,42,44,46

13,13,9,9,46,48,50

13,13,11,11,50,52,54

13,13,15,15,58,60,62

13,1,1,1,18,20,22

13,3,3,3,24,26,28

13,5,5,5,30,32,34

13,7,7,7,36,38,40

13,9,9,9,42,44,46

13,11,11,11,48,50,52

13,15,15,15,60,62,64

15,15,15,15,62,64,66

15,15,15,1,48,50,52

15,15,15,3,50,52,54

15,15,15,5,52,54,56

15,15,15,7,54,56,58

15,15,15,9,56,58,60

15,15,15,11,58,60,62

15,15,15,13,60,62,64

15,15,1,1,34,36,38

15,15,3,3,38,40,42

15,15,5,5,42,44,46

15,15,7,7,46,48,50

15,15,9,9,50,52,54

15,15,11,11,54,56,58

15,15,13,13,58,60,62

15,1,1,1,20,22,24

15,3,3,3,26,28,30

15,7,7,7,38,40,42

15,9,9,9,44,46,48

15,11,11,11,50,52,54

15,13,13,13,56,58,60

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 2.4

=== รายงานค่าความแม่นยำ (Accuracy Metrics) ===

[output1]

- R-squared (ความแม่นยำ): 1.0000
- Mean Squared Error: 0.0043

[output2]

- R-squared (ความแม่นยำ): 0.9999
- Mean Squared Error: 0.0129

[output3]

- R-squared (ความแม่นยำ): 0.9998
- Mean Squared Error: 0.0260

=== ผลการทำนายสำหรับข้อมูลตัวอย่างใหม่ ===

Input: v1=5, v2=5, v3=5, v4=5

ทำนาย Output 1: 21.9249

ทำนาย Output 2: 23.8715

ทำนาย Output 3: 25.8181

บทที่ 3. การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Ridge regression

อัลกอริธึมการวิเคราะห์การถดถอยแบบริจ (Ridge Regression) คือ อัลกอริธึมที่เป็นการขยายจากการวิเคราะห์การถดถอยเชิงเส้นแบบปกติ โดยทำการเพิ่ม บทลงโทษ (Penalty) ให้กับขนาดของค่าสัมประสิทธิ์ (Coefficients) โดยใช้แนวคิด Regularization เพื่อควบคุมความซับซ้อนของโมเดล โดยการเพิ่ม Penalty เข้าไปในสมการ ทำให้ผลลัพธ์ที่ได้เป็น (1) โมเดลมีความเสถียร (2) ลดการเกิด Overfitting และ (3) สามารถทำนายข้อมูลใหม่ได้ดีขึ้น

สำหรับหลักการสำคัญได้แก่ (1) การแก้ปัญหา Overfitting สำหรับกรณีที่ถ้าโมเดลมีการเรียนรู้ข้อมูล หรือฝึกสอนมากเกินไป จะทำให้โมเดลทำนายข้อมูลใหม่ได้ไม่ดี การวิเคราะห์การถดถอยแบบริจ จะทำให้ค่าสัมประสิทธิ์มีค่าไม่ใหญ่จนเกินไป ทำให้โมเดล Generalize ได้ดีขึ้น หรือทำการอธิบายกรณีทั่วไปได้ดีขึ้น หมายถึงความสามารถของโมเดลในการ นำสิ่งที่เรียนรู้จากข้อมูลฝึก (Training data) ไปใช้ทำนายข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนได้อย่างแม่นยำ หรือ Generalize เป็นความสามารถในการทำนายข้อมูลใหม่ได้ดี (2) การจัดการ Multicollinearity คือเมื่อ ตัวแปรอิสระมีความสัมพันธ์กันสูง การวิเคราะห์การถดถอยแบบริจ จะทำให้ ทำให้เมทริกซ์สามารถกลับมา Invert ได้ และลด Variance ของโมเดล (3) Bias-Variance Tradeoff คือ การที่ทำการเพิ่ม Bias เพียงเล็กน้อย แต่จะทำให้ลดค่า Variance ลงอย่างมาก ทำให้ผลรวมค่าความผิดพลาดมีค่าดีขึ้น

ขั้นตอนการทำงาน ได้แก่ (1) การเตรียมข้อมูล (2) ทำ Feature scaling (3) การเลือกค่า Lambda โดยถ้า Lambda มีค่าเป็น 0 จะ

กลายเป็น การวิเคราะห์การถดถอยเชิงเส้น ส่วนถ้า Lambda มีค่ามาก ๆ ค่าสัมประสิทธิ์จะมีขนาดเล็กลง โดยวิธีเลือกได้แก่ (3.1) Cross-validation และ(3.2) Grid search (4) ทำการคำนวณค่าสัมประสิทธิ์ (5) ทำการสร้างโมเดล

สำหรับการทำนายค่าผลลัพธ์ จะทำเมื่อได้โมเดลแล้ว โดยจะใส่ข้อมูลใหม่ หรือ x_{new} และคูณข้อมูลกับค่าน้ำหนัก หรือ Weight: w โดย $y_{new} = x_{new} \cdot w$ ตัวอย่างเช่น โมเดล $y = 3 + 0.6x_1 + 1.4x_2$ โดยถ้าข้อมูลใหม่ มีค่าเป็น $x_1 = 6$ และ $x_2 = 5$ จะได้ว่า $y = 3 + 0.6(6) + 1.4(5) = 13.6$

สำหรับจุดเด่นของ การวิเคราะห์การถดถอยแบบริจ ได้แก่ (1) ช่วยลด Overfitting (2) ทำงานดีเมื่อ ตัวแปรอิสระมีจำนวนมาก (3) ช่วยแก้ปัญหา Multicollinearity (5) ทำให้โมเดลเสถียรขึ้น

ตัวอย่างที่ 3.1 คำสั่งการวิเคราะห์การถดถอยแบบริจ (Ridge Regression)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error, r2_score

def main():
    # 1. โหลดข้อมูลจากไฟล์ CSV
    try:
        # อ่านไฟล์ re_3inp_data.csv (ต้องอยู่ในโฟลเดอร์เดียวกันกับสคริปต์)
        df = pd.read_csv('/content/re_3inp_data.csv')
    except FileNotFoundError:
        print("ข้อผิดพลาด: ไม่พบไฟล์ 're_3inp_data.csv' โปรดตรวจสอบให้แน่ใจว่าไฟล์มีอยู่จริง")
        return

    # ตรวจสอบว่าคอลัมน์ครบถ้วนหรือไม่
    required_columns = ['v1', 'v2', 'v3', 'output']
    if not all(col in df.columns for col in required_columns):
        print(f"ข้อผิดพลาด: ไฟล์ CSV ต้องมีคอลัมน์ {required_columns}")
        return

    # 2. แยกข้อมูล Input (X) และ Output (y)
    X = df.iloc[:, 0:3]
    y  = df.iloc[:, 3]

    # 3. แบ่งข้อมูลเป็น Train Set (80%) และ Test Set (20%) สำหรับวัดความแม่นยำ
    X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2,
    random_state=42)
```

ตัวอย่างที่ 3.1 คำสั่งการวิเคราะห์การถดถอยแบบริจ (Ridge Regression) ต่อ 1

```
# 4. สร้างและฝึกสอน (Train) โมเดล Ridge Regression
# กำหนด alpha = 1.0 เป็นค่าเริ่มต้น (สามารถปรับแต่งเพื่อลด Overfitting ได้)
model = Ridge(alpha=4)
model.fit(X_train, y_train)

# 5. ประเมินความแม่นยำของโมเดล (Evaluation)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("="*40)
print(" รายงานความแม่นยำของโมเดล (Model Evaluation)")
print("="*40)
print(f"Mean Squared Error (MSE): {mse:.4f} (ยิ่งน้อยยิ่งดี)")
print(f"R-squared (R2 Score):     {r2:.4f} (อธิบายความแปรปรวนได้ {r2*100:.2f}%,
 ยิ่งใกล้ 1 ยิ่งแม่นยำ)")
print("="*40)
print()

# 6. ทำนายข้อมูลตัวอย่าง 1 ชุด ได้แก่ s1, s2, s3
# กำหนดค่าข้อมูลตัวอย่าง (คุณสามารถแก้ไขค่า s1, s2, s3 ตรงนี้ได้)
print('Pleased enter integer number from 0 to 15, Not exceed 15 ')
var1 = float(input('Enter value var1: '))
var2 = float(input('Enter value var2: '))
var3 = float(input('Enter value var3: '))
s1 = var1
s2 = var2
s3 = var3

# สร้างเป็น DataFrame เพื่อใส่เข้าไปในโมเดล (ป้องกันคำเตือนจาก sklearn)
sample_data = pd.DataFrame([[s1, s2, s3]], columns=['v1', 'v2', 'v3'])
```

ตัวอย่างที่ 3.1 คำสั่งการวิเคราะห์การถดถอยแบบริจ (Ridge Regression) ต่อ 2

```
  # ทำการทำนายผลลัพธ์
    predicted_output = model.predict(sample_data)[0]

    print("="*40)
    print(" ผลการทำนายข้อมูลตัวอย่าง (Prediction)")
    print("="*40)
    print(f"ข้อมูลนำเข้า (Inputs):")
    print(f"  v1 (s1) = {s1}")
    print(f"  v2 (s2) = {s2}")
    print(f"  v3 (s3) = {s3}")
    print("-" * 40)
    print(f"ผลลัพธ์ที่ทำนายได้ (Predicted Output): {predicted_output:.4f}")
    print("="*40)

if __name__ == "__main__":
    main()
```

ลักษณะของข้อมูลที่ใช้กับตัวอย่าง 3.1 ไฟล์ re_3inp_data.csv

v1,v2,v3,output

0,0,0,0

1,1,1,4

1,2,3,7

1,3,4,9

1,5,6,13

2,1,1,5

2,4,6,13

2,1,3,7

3,4,3,11

3,4,5,13

3,1,3,8

4,1,1,7

4,2,4,11

4,2,6,13

5,1,2,9

5,3,3,12

5,2,4,12

5,5,4,13

6,4,1,12

6,4,3,14

6,3,5,15

6,4,6,17

6,5,7,19

6,7,8,22

7,1,1,9

7,3,4,15

7,5,5,18

7,6,7,21

8,5,1,15

8,7,3,19

8,4,3,16

8,8,3,20

8,3,6,18

9,1,2,13

9,3,4,17

9,5,5,20

9,6,3,19

9,7,5,22

9,8,3,21

9,10,2,22

10,1,3,15

10,3,5,19

10,5,7,23

10,6,9,26

10,9,10,30

11,1,3,16

11,2,4,18

11,4,6,22

11,7,8,27

11,9,5,26

11,10,6,28

11,10,9,31

11,12,13,37

12,14,3,30

12,5,3,21

12,8,6,27

12,19,7,39

12,4,10,27

12,10,5,28

12,11,7,31

13,1,4,19

13,3,6,23

13,5,7,26

13,7,9,30

13,9,11,34

13,11,13,38

14,1,2,18

14,3,4,22

14,5,6,26

14,7,8,30

14,9,10,34

14,12,13,40

15,1,2,19

15,3,4,23

15,5,6,27

15,7,8,31

15,9,9,34

15,11,12,39

15,13,14,42

15,15,15,46

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 3.1

```
========================================
 รายงานความแม่นยำของโมเดล (Model Evaluation)
========================================
Mean Squared Error (MSE): 0.0586 (ยิ่งน้อยยิ่งดี)
R-squared (R2 Score): 0.9990 (อธิบายความแปรปรวนได้ 99.90%, ยิ่งใกล้ 1 ยิ่งแม่นยำ)
========================================
Pleased enter integer number from 0 to 15, Not exceed 15
Enter value var1: 2
Enter value var2: 4
Enter value var3: 6
========================================
 ผลการทำนายข้อมูลตัวอย่าง (Prediction)
========================================
ข้อมูลนำเข้า (Inputs):
  v1 (s1) = 2.0
  v2 (s2) = 4.0
  v3 (s3) = 6.0
----------------------------------------------------------------------------
ผลลัพธ์ที่ทำนายได้ (Predicted Output): 12.9059
========================================
```

บทที่ 4. การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม Logistic regression

อัลกอริธึมการวิเคราะห์การถดถอยแบบลอจีสติก (Logistic regression) คือ อัลกอริธึมที่ใช้สำหรับการจำแนกประเภท หรือ Classification กับปัญหาที่มีผลลัพธ์เป็นแบบไม่ต่อเนื่องเช่น ใช่หรือไม่ใช่ ได้หรือไม่ได้ 0 หรือ1 เป็นการทำนายความน่าจะเป็นของประเภทหรือคลาส หรือเป็นโมเดลที่ใช้ เส้นตรง ร่วมกับ ฟังก์ชัน Sigmoid เพื่อทำนายความน่าจะเป็น โดยจะแปลงผลลัพธ์ให้มีค่าอยู่ระหว่าง 0 กับ 1 แล้วใช้ค่า ขีดเริ่มเปลี่ยนหรือ Threshold ในการตัดสินประเภทหรือคลาส ที่เรียนรู้เพื่อลดความผิดพลาดด้วย Gradient Descent หลักการของการวิเคราะห์การถดถอยแบบลอจีสติก เป็นการนำแนวคิดของ การวิเคราะห์การถดถอยเชิงเส้นมาปรับให้เหมาะกับการจำแนกประเภทโดยการใช้ฟังก์ชันพิเศษที่เรียกว่า ฟังก์ชัน Sigmoid หรือ ฟังก์ชันลอจีสติก

การทำงานของการวิเคราะห์การถดถอยแบบลอจีสติก ได้แก่ (1) รับข้อมูลเข้า ได้แก่ ข้อมูลตัวแปรอิสระ (2) คำนวณค่าผลรวมเชิงเส้น โดยใช้ค่าน้ำหนัก (Weights) คูณกับ ตัวแปรอิสระแต่ละตัว แล้วนำมารวมกัน (3) ทำการแปลงค่าเป็นค่าความน่าจะเป็น โดยใช้ ฟังก์ชัน Sigmoid เพื่อปรับค่าให้อยู่ในช่วง 0 ถึง 1 (4) ทำการตัดสินใจผลลัพธ์ โดยตั้งค่า ขีดเริ่มเปลี่ยนเช่น 0.4 โดย ถ้ามีค่ามากกว่า 0.4 ให้ทำนายว่าเป็นประเภท 1 ส่วนถ้ามีค่าน้อยกว่า 0.4 ให้ทำนายว่าเป็นประเภท 0

การเรียนรู้หรือการฝึกสอนของโมเดล จะใช้การเรียนรู้ด้วยการปรับค่าน้ำหนักให้เหมาะสมที่สุด ด้วยกระบวนการได้แก่ (1) ฟังก์ชันค่าเสียหาย (Loss function) โดยใช้ Cross entropy ซึ่งวัดว่าโมเดลทำนายผิดพลาด

มากแค่ไหนจากค่าจริง โดยถ้าค่ายิ่งน้อยแสดงถึงโมเดลยิ่งดี ถ้าค่ายิ่งมากแสดงถึงโมเดลยิ่งแย่ (2) ปรับน้ำหนัก ด้วยการใช้วิธี Gradient Descent โดยค่อย ๆ ปรับน้ำหนักให้ Loss ลดลง ด้วยการทำซ้ำหลาย ๆ รอบ

การทำนายผลลัพธ์ เมื่อโมเดลถูกฝึกสอนแล้ว จะทำขั้นตอนได้แก่ (1) ทำการใส่ข้อมูลเข้าไปใหม่ (2) คำนวณค่า z (3) ใช้ฟังก์ชัน Sigmoid หาค่าความน่าจะเป็น และ (4) ทำการเปรียบเทียบกับ ค่าขีดเริ่มเปลี่ยน หรือ Threshold เพื่อให้ได้คำตอบสุดท้าย

สำหรับประเภทของการวิเคราะห์การถดถอยแบบลอจีสติกได้แก่ (1) การวิเคราะห์การถดถอยแบบลอจีสติก แบบไบนารี่ (Binary Logistic Regression) ที่ทำนายได้สองประเภทได้แก่ 1หรือ 0 (2) การวิเคราะห์การถดถอยแบบลอจีสติก แบบมัลติโนเมียล (Multinomial Logistic Regression) ที่ทำนายได้มากกว่าสองประเภท และ (3) การวิเคราะห์การถดถอยแบบลอจีสติก แบบออดินอล (Ordinal Logistic Regression) ที่ทำนายประเภทแบบมีลำดับ เช่น น้อย หรือปานกลาง หรือมาก

ตัวอย่างที่ 4.1 คำสั่งการวิเคราะห์การถดถอยแบบลอจิสติกเพื่อการจำแนก

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
x = [[2, 3, 2], [3, 3, 1], [1, 2, 2], [3, 4, 3], [5, 4, 4], [8, 8, 8], [8, 9, 9], [7, 7, 8],
      [6, 7, 7], [10, 9, 8]]
y = ['Honda', 'Honda', 'Honda', 'Honda', 'Honda', 'Toyota', 'Toyota', 'Toyota',
      'Toyota', 'Toyota']

model.fit(x, y)

print('Enter your values are want to predict !')
print('Values are 1, 10, or integer between 1 and 10')
x1 = float(input('Enter value x1: '))
x2 = float(input('Enter value x2: '))
x3 = float(input('Enter value x3: '))
x_predict = [[x1, x2, x3]]
y_predict = model.predict(x_predict)

print('Predict group is:',y_predict[0])
print()
print()
```

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 4.1

Enter your values are want to predict !

Values are 1, 10, or integer between 1 and 10

Enter value x1: 1

Enter value x2: 2

Enter value x3: 3

Predict group is: Honda

บทที่ 5. การเรียนรู้แบบมีผู้สอน ด้วยอัลกอริธึม K-nearest neighbors

สำหรับอัลกอริธึม เพื่อนบ้านใกล้เคียงที่สุด K ตัว (K-Nearest Neighbors) ทำงานโดยอาศัยแนวคิดที่ว่า ข้อมูลที่อยู่ใกล้กัน มักจะมีลักษณะที่คล้ายคลึงกัน โดยเมื่อมีข้อมูลเข้ามาใหม่ อัลกอริธึม จะทำการได้แก่ (1) วัดระยะห่างระหว่างข้อมูลใหม่นี้กับข้อมูลทั้งหมดในชุดฝึก หรือ Training data จากนั้นจะเลือก เพื่อนบ้านที่ใกล้ที่สุด จำนวน K จุด แล้วใช้ข้อมูลของเพื่อนบ้านเหล่านั้น สำหรับการทำนายค่าผลลัพธ์ โดยค่า K หรือ จำนวนเพื่อนบ้านที่เราจะพิจารณา มีตัวอย่างเช่น K = 3 จะใช้จุดที่ใกล้ที่สุด จำนวน 3 จุด ส่วน ถ้า K = 5 จะใช้จุดที่ใกล้ที่สุด จำนวน 5 จุด หรือ ถ้า K = 7 จะใช้จุดที่ใกล้ที่สุด จำนวน 7 จุด ทั้งนี้สำหรับการวัดระยะห่าง (Distance Metrics) จะใช้วิธีได้แก่ (1) Euclidean distance (ระยะห่างเส้นตรง) ที่หาได้จาก $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ (2) Manhattan distance และ (3) Minkowski distance ทั้งนี้ ถ้าระยะห่างมากจะมีความเหมือนน้อย และถ้าระยะห่างน้อยจะมีความเหมือนมาก หรือระดับความเหมือนจะเปลี่ยนแปลงตามระยะห่าง ซึ่งในการตัดสินใจค่าผลลัพธ์แบ่งเป็นสองกรณีได้แก่ (1) กรณีการจำแนกประเภท จะใช้วิธี โหวตเสียงข้างมาก เช่น กรณีที่มีเพื่อนบ้านมี 5 จุด โดย 3 จุดเป็นการเดิน และ 2 จุดเป็นการวิ่ง จะทำนายว่าเป็น การเดิน หรือจะเปรียบเทียบความเหมือนของตำแหน่งข้อมูลที่จะทำนายกับตำแหน่งข้อมูลที่อยู่รอบ ๆ ว่าอยู่ใกล้กับข้อมูลใดมากที่สุด จะถือว่าตำแหน่งที่ทำนายผลเป็นประเภทเดียวกับตำแหน่งที่อยู่ใกล้ที่สุดนั้น (2) กรณีการทำนายค่าตัวเลขผลลัพธ์ด้วยวิธีการวิเคราะห์การถดถอย จะใช้การหาค่าเฉลี่ยของ K จุด เช่น ข้อมูลตำแหน่งข้างเคียงที่อยู่ใกล้ที่สุด 3 จุด ได้แก่ 3, 6, 9 จะได้ค่า

ตัวเลขผลลัพธ์เท่ากับ 6 ทั้งนี้ต้องเลือกค่า K ให้เหมาะสมเนื่องจากส่งผลต่อผลลัพธ์การทำนาย

ตัวอย่างที่ 5.1 คำสั่งการวิเคราะห์เพื่อจำแนกประเภทด้วยวิธีเพื่อนบ้านใกล้เคียงที่สุด k ตัว (K-Nearest Neighbors)

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('/content/object_size2.csv')
x = df.iloc[: ,[0, 1]]
y = df[['object_size']]

k = 5
model = KNeighborsClassifier(n_neighbors=k)
model.fit(x,y)

x_pred = [[10, 9]]
y_pred = model.predict(x_pred)

h = x_pred[0][0]
w = x_pred[0][1]
p = y_pred[0]

print('K =', k )
print(f'Height: {h}, Width: {w}')
print('Predict Object size is: ',p)
print('Accuracy: ', '{:.4f}'.format(model.score(x,y)))
print()
```

ลักษณะของข้อมูลที่ใช้กับตัวอย่าง 5.1 ไฟล์ object_size2.csv

x_para,y_para ,object_size

5,5,S

30,20,L

30,30,L

7,8,S

8,9,S

12,13,M

14,15,M

22,24,L

10,10,S

20,20,M

18,17,M

6,6,S

9,9,S

1,1,S

2,2,S

15,15,M

19,18,M

16,16,M

11,11,M

11,13,M

3,3,S

7,7,S

25,25,L

27,27,L

29,29,L

21,21,L

23,23,M

13,13,M

19,19,M

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 5.1

K = 5

Height: 10, Width: 9

Predict Object size is: S

Accuracy: 0.9310

ตัวอย่างที่ 5.2 คำสั่งการวิเคราะห์เพื่อจำแนกประเภทด้วยวิธี
เพื่อนบ้านใกล้เคียงที่สุด k ตัว (K-Nearest Neighbors)

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('/content/object_size1.csv')

x = df.iloc[: ,[0, 1, 2]]
y = df[['object_size']]

k = 5
model = KNeighborsClassifier(n_neighbors=k)
model.fit(x,y)

x_predict = [[12, 13, 11]]
y_predict = model.predict(x_predict)

h  = x_predict[0][0]
w = x_predict[0][1]
d  = x_predict[0][2]
p  = y_predict[0]

print('K =', k )
print(f'Height: {h}, Width: {w}, Depth: {d}')
print('Predict Object Size is:', p)
print('Accuracy: ', '{:.4f}'.format(model.score(x,y)))
print()
print()
```

ลักษณะของข้อมูลที่ใช้กับตัวอย่าง 5.2 ไฟล์ object_size1.csv

height_ob,width_ob ,depth_ob,object_size

5,5,5,Small

30,20,25,Large

30,30,30,Large

7,8,9,Small

8,9,10,Small

12,13,12,Medium

14,15,16,Medium

22,24,21,Large

10,10,10,Small

20,20,20,Medium

18,17,17,Medium

6,6,6,Small

9,9,9,Small

1,1,1,Small

2,2,1,Small

15,15,14,Medium

19,18,18,Medium

16,16,15,Medium

11,11,11,Medium

11,13,12,Medium

3,3,3,Small

7,7,7,Small

25,25,25,Large

27,27,25,Large

29,29,28,Large

21,21,21,Large

23,23,23,Medium

13,13,13,Medium

19,19,19,Medium

ผลลัพธ์การทำคำสั่งตามตัวอย่างที่ 5.2

K = 5

Height: 12, Width: 13, Depth: 11

Predict Object Size is: Medium

Accuracy: 0.9310