

# หลักการเรียนรู้ของเครื่องและการประยุกต์

ผศ.ดร.สิริวัชริ์ ครีสุทธิยกุล

## Contents

<b>บทที่ 1 ความรู้พื้นฐานเกี่ยวกับการเรียนรู้ของเครื่อง</b>	<b>3</b>
1.1 What's algorithms? . . . . .	3
1.2 กิจกรรม Rule-based algorithm . . . . .	5
1.3 AI vs ML vs DL . . . . .	7
1.4 Data Analysis vs Data Analytics . . . . .	8
1.5 Types of Machine Learning? . . . . .	9
Supervised Learning . . . . .	10
Unsupervised Learning . . . . .	10
Reinforcement Learning . . . . .	11
1.6 ตัวอย่างการประยุกต์ใช้ ML ทางการศึกษา (การบ้าน) . . . . .	12
<b>บทที่ 2 ความรู้พื้นฐานในการพัฒนา Supervised Learning</b>	<b>13</b>
2.1 ติดตั้ง R และ RStudio . . . . .	13
2.2 ติดตั้ง Anaconda . . . . .	13
2.3 กิจกรรม My First Regression . . . . .	15
Fitting Linear Regression using lm() . . . . .	15
Evaluation Metrics . . . . .	16
2.4 กิจกรรม My First Classification . . . . .	18
Fitting Logistic Regression Model using glm() . . . . .	18
Evaluation Metrics . . . . .	20
คำถ้ามท้ายบท . . . . .	22
<b>บทที่ 3 กระบวนการพัฒนา Machine Learning Models</b>	<b>23</b>
3.1 Bias and Variance in ML models . . . . .	25
3.2 Underfitting, Overfitting และ Good fit models . . . . .	26
3.3 Training, validation, and Test Dataset . . . . .	26
3.4 Data Partitioning . . . . .	28

ๆดีข้อมูล mpg . . . . .	28
การแบ่งข้อมูลด้วยการสุ่มอย่างง่าย . . . . .	29
การแบ่งข้อมูลด้วยการสุ่มแบบชั้นภูมิ . . . . .	30
3.5 Tidymodels Framework . . . . .	32
Fitting Linear Regression using parsnip . . . . .	33
Evaluating models using yardstick . . . . .	37
3.6 กิจกรรมพัฒนา regression model ด้วย tidymodel framework . . . . .	39
สรุป . . . . .	40

# บทที่ 1 ความรู้พื้นฐานเกี่ยวกับการเรียนรู้ของเครื่อง

ปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทอย่างมากกับมนุษย์ โดยเข้ามาช่วยเหลือ สนับสนุน หรือแม้กระทั่งทำงานแทนมนุษย์ให้หลาย ๆ งาน ส่วนประกอบสำคัญส่วนหนึ่งของคอมพิวเตอร์ที่จะขาดไปไม่ได้ในการดำเนินงาน/กิจกรรมต่าง ๆ คือโปรแกรมคอมพิวเตอร์

ที่ผ่านมากโปรแกรมคอมพิวเตอร์ถูกพัฒนาขึ้นเป็นจำนวนมาก รวมถึงศาสตร์ในด้านการพัฒนาโปรแกรมคอมพิวเตอร์ก็ถูกพัฒนาไปอย่างมากเข่นกัน เราอาจจำแนกส่วนประกอบของโปรแกรมคอมพิวเตอร์ได้เป็น 2 ส่วนได้แก่ ส่วนต่อประสานกับผู้ใช้ (user interface: UI) และส่วนประมวลผลหรือความคุมการทำงานของโปรแกรมซึ่งในเชิงเทคนิคเรียกว่า อัลกอริทึม (algorithm) ของโปรแกรม ทั้งนี้ในโปรแกรมหนึ่ง ๆ อาจมีอัลกอริทึมมากกว่าหนึ่งตัวก็ได้

บทเรียนนี้จะกล่าวถึงภาพรวมของมโนทัศน์และคำศัพท์ที่เกี่ยวข้องกับศาสตร์การเรียนรู้ของเครื่องเพื่อเป็นพื้นฐานสำหรับการศึกษาในบทเรียนอื่น ๆ ต่อไป รายละเอียดมีดังนี้

## 1.1 What's algorithms?

อัลกอริทึมคือกระบวนการในการดำเนินงานที่มีขั้นตอนอย่างชัดเจน โดยมีวัตถุประสงค์เพื่อทำงาน/แก้ปัญหาที่กำหนดให้สำเร็จ การใช้อัลกอริทึมในการทำงานนั้นไม่ได้จำกัดเฉพาะงานทางด้านคอมพิวเตอร์ หรือสถิติและวิทยาการข้อมูลเท่านั้น แต่ในชีวิตประจำวันเรา ก็มีการใช้อัลกอริทึมเพื่อดำเนินงานต่าง ๆ อยู่เป็นประจำ เช่น

- การเดินทางจากบ้านไปยังร้านขายของสะดวกซื้อ งานดังกล่าวสามารถเรียนรู้และออกแบบเป็นขั้นตอนการเดินทางโดยอาจเริ่มต้นจากการออกประตูบ้าน เลี้ยวขวา เดินตรงไป เมื่อพบสามแยกให้เลี้ยวขวาอีกครั้งจะพบร้านสะดวกซื้อ
- การทดสอบเจียวที่อาจเริ่มจากการตั้งไฟ ใส่น้ำมัน ตอกไข่ ตีไข่ ใส่เครื่องปรุง ทอดไข่ และนำไปเจียวที่ได้เสริฟ

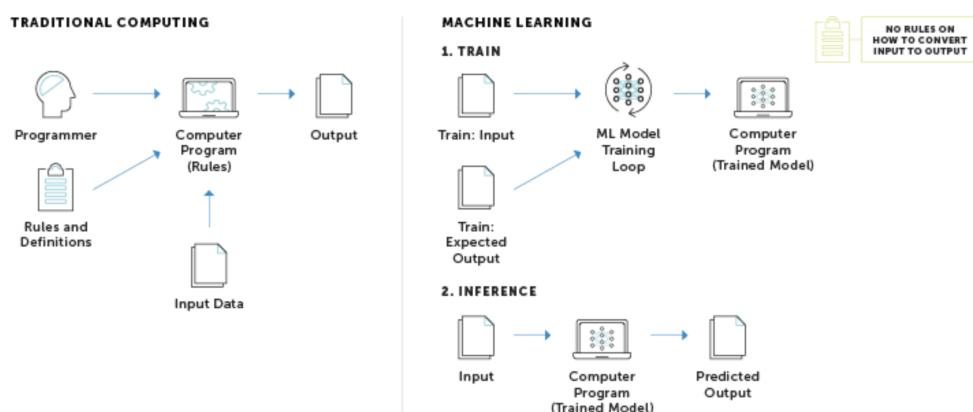


Figure 1: rule-based vs ML-based (<https://www.epam.com/insights/blogs/making-ai-more-human-black-box-models-lead-to-better-decision-making>)

ปัจจุบันโลกได้ก้าวเข้าสู่ยุคที่ให้เครื่องจักรทำงานบางอย่างแทนมนุษย์ได้ ซึ่งเป็นหลักการดำเนินการของเครื่องจักรต่าง ๆ จำเป็นต้องมีอัลกอริทึมที่ใช้สำหรับควบคุมการทำงาน การพัฒนาอัลกอริทึมดังกล่าวอาจทำได้สองวิธีการ วิธีการแรกเรียกว่า rule-based algorithm ที่ผู้พัฒนาเป็นผู้กำหนดขั้นตอนวิธีการทำงานและประมวลผลทั้งหมด เพื่อให้เป็นภาพของ rule-based algorithm ผู้อ่านลองพิจารณาด้วยอย่างในรูปด้านล่าง

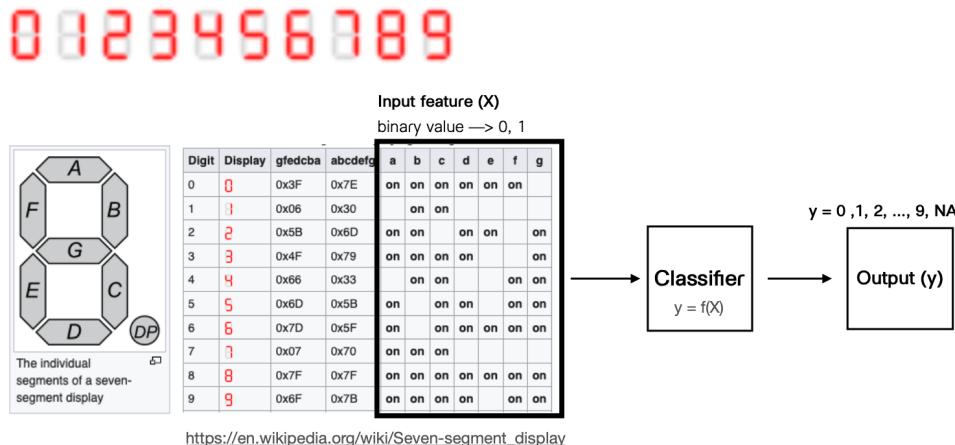


Figure 2: 7-segment display problem (ที่มา: wikipedia)

รูปข้างต้นเป็นปัญหาที่ว่า 7-segment display วัดถูประสงค์คือต้องการพัฒนาโปรแกรมที่จะใช้จำแนกตัวเลขบนป้ายไฟแบบ Digital โดยบนป้ายไฟ LED จำนวน 7 หลอด ได้แก่ A, B, C, D, E, F และ G การเปิด/ปิดไฟอย่างเหมาะสมตามตารางในรูป จะทำให้ได้ตัวเลข 0-9 บนป้ายไฟ

จากปัญหานี้ทางด้านการพัฒนาโปรแกรมสำหรับจำแนกตัวเลขบนป้าย (classifier) ด้วยวิธีการแบบ rule-based ผู้วิเคราะห์จะต้องมี sensor เพื่อตรวจสอบการเปิด/ปิดไฟแต่ละหน่วยบนป้าย จากนั้นจึงพัฒนาโปรแกรมเพื่ออ่านค่าของตัวเลข โดยอัลกอริทึมแบบ rule-based อาจมีลักษณะเป็นการเขียนคำสั่งด้วยการใช้ IF...ELSE ดังนี้

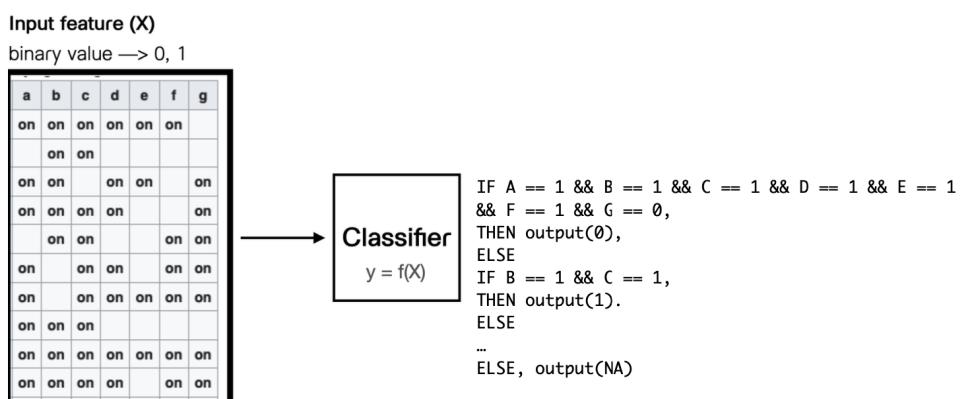


Figure 3: Rule-based Classifier

Table 1: ข้อมูลคะแนนสอบและพฤติกรรมการเรียนของนักเรียน

behav	score
0	1
1	2
2	2
3	3
4	4
5	4

การพัฒนา rule-based algorithms ไม่ได้จำเป็นต้องใช้แค่ if...else แต่เพียงอย่างเดียว แต่สามารถใช้กระบวนการทางคณิตศาสตร์เข้ามาช่วยในการสร้างได้ ลองพิจารณาดัวอย่างในกิจกรรมต่อไปนี้

## 1.2 กิจกรรม Rule-based algorithm

ข้อมูลในตาราง 1 ประกอบด้วยคะแนนสอบ (score) กับคะแนนพฤติกรรมการเรียนของนักเรียน (behav) ลองดำเนินการดังนี้

1. ลองวิเคราะห์ความสัมพันธ์เบื้องต้นระหว่าง score กับ behav ความสัมพันธ์ระหว่างตัวแปรทั้งสองมีลักษณะเป็นอย่างไร
2. หากให้วิธีการทางคณิตศาสตร์เพื่อสร้างสมการที่จะใช้เป็นตัวแทนความสัมพันธ์ที่พบรอบในข้อ 1. จะได้สมการเป็นอย่างไร
3. สมการที่สร้างได้จากข้อ 2. มีประสิทธิภาพในการทำนาย score ดีหรือไม่ อย่างไร

ตัวอย่างในรูปข้างต้นแสดงให้เห็นว่า ในกรณีที่ข้อมูลนำเข้าไม่มีความคลาดเคลื่อนหรือมีความคลาดเคลื่อนอยู่ในระดับต่ำมาก ๆ การพัฒนาโปรแกรมด้วยวิธีการแบบ rule-based ก็สามารถทำได้อย่างมีประสิทธิภาพ อย่างไรก็ตามในปัจจุบันทั่วไปในโลกจริง เป็นการยากที่จะมีข้อมูลนำเข้าที่มีคุณสมบัติดังกล่าว ผู้อ่านลงพิจารณาปัจจุบัน Handwriting Digit Recognition (Geron, 2019) ซึ่งเป็นปัจจุบันการจำแนกตัวเลข เช่นเดียวกับปัจจุบัน 7-segment display ข้างต้น แต่มีความแตกต่างกันคือ input feature สำหรับปัจจุบันนี้จะเป็นลายมือของคนจริง ๆ ไม่ใช่ข้อมูลแบบ binary ของไฟ LED



Figure 4: Handwritting Digit Recognition

จากรูปข้างต้นจะเห็นว่าการเรียนกฎเกณฑ์แบบ rule-based เพื่อจำแนกตัวเลขในรูปข้างต้นทำได้ยากมากและเป็นไปแทนไม่ได้เลยที่จะพัฒนาอัลกอริทึมแบบ rule-based สำหรับจำแนกตัวเลขจากลายมือดังกล่าวได้อย่างมีประสิทธิภาพ ดังนั้นการที่จะพัฒนาโปรแกรมให้ทำงานแทนมนุษย์ได้จริง ๆ นั้น วิธีการแบบ rule-based เป็นวิธีการที่ยังมีข้อจำกัดอยู่มาก ทั้งนี้ เพราะเป็นวิธีการขาดความยืดหยุ่นและไม่สามารถใช้ได้หากข้อมูลนำเข้ามีความแตกต่างไปจากกฎเกณฑ์ที่วางเอาไว้มีจุดเด่นอย่างเด่นอย่าง จึงมีแนวโน้มที่จะเกิดความคลาดเคลื่อนสูงมากเมื่อนำไปrogramแบบ rule-based ไปใช้ในสถานการณ์ทั่วไป

วิธีการที่สองเรียกว่า machine learning-based ที่เป็นวิธีการสมัยใหม่และปัจจุบันถูกใช้เป็นวิธีการหลักวิธีหนึ่งในการพัฒนาโปรแกรม การเรียนรู้ของเครื่อง เป็นศาสตร์ย่อยแขนงหนึ่งภายในศาสตร์ทางด้านสถิติและวิทยาการข้อมูล ซึ่งเกี่ยวข้องกับการพัฒนาและใช้อัลกอริทึม ที่มีความสามารถในการเรียนรู้สารสนเทศจากข้อมูลได้ แล้วนำสารสนเทศดังกล่าวมาใช้งาน อัลกอริทึมในกลุ่มนี้ถูกใช้มากสำหรับการสร้างโมเดลเพื่อทำนาย จำแนก หรือตัดสินใจ โดยโมเดลที่สร้างขึ้นจะมีความแตกต่างไปจากโมเดลที่ได้จากการอัลกอริทึมแบบ rule-based ตรงที่มีการใช้วิธีการทางสถิติและความน่าจะเป็นเข้ามาช่วยจัดการกับความคลาดเคลื่อนที่อาจเกิดขึ้นในข้อมูลนำเข้า โมเดลที่พัฒนาขึ้นจึงมีความยืดหยุ่นและมีประสิทธิภาพมากกว่าโมเดลที่พัฒนาด้วยอัลกอริทึมแบบ rule-based ประสิทธิภาพดังกล่าวไม่ได้หมายถึงประสิทธิภาพในการทำงานแต่เพียงอย่างเดียว แต่ยังหมายถึงประสิทธิภาพในการพัฒนาโมเดลด้วย

ตัวอย่างต่อไปนี้แสดงการใช้ ML-based เพื่อจำแนกตัวเลขจากลายมือในปัจจุบัน Handwriting Digit Recognition ข้างต้น

- <https://www.kaggle.com/code/pranjalrathore/digit-recognizer-minst>
- <https://www.kaggle.com/code/alphahostusmc/mnist-cnnv2>

- <https://www.kaggle.com/code/kobakhit/digital-recognizer-in-r>
- <https://www.kaggle.com/code/ivoruaro/mnist-xgboost-r>

### 1.3 AI vs ML vs DL

ปัจจุบันมีการใช้คำว่า AI, ML และ DL แทนกันไปมาจนบางครั้งเหมือนว่าจะเป็นคำเดียวกัน ในความเป็นจริงทั้งสามคำดังกล่าวมิได้เป็นสิ่งเดียวกันเลยที่เดียว แต่มีทั้งส่วนที่เหมือนและแตกต่างกัน รายละเอียดมีดังนี้

- **AI ย่อมาจาก Artificial Intelligent** เป็นเทคนิคหรือวิธีการที่นักวิทยาการข้อมูลใช้เพื่อพัฒนาโปรแกรมคอมพิวเตอร์ รวมถึงหุ่นยนต์หรือจักรกลที่สามารถเลียนแบบการทำงานต่าง ๆ ของมนุษย์ได้ AI จะมีความสามารถในการทำงานใกล้เคียงหรือดีกว่ามนุษย์ ทั้งความสามารถในการจำจำแนก และตัดสินใจดำเนินงานเองโดยอาศัยข้อมูลที่เป็นไปได้ทั้งข้อมูลตัวเลข ข้อความ รูปภาพ และเสียง ตัวอย่างของ AI เช่น รถยนต์หรือยานพาหนะไร้คนขับ, AlphaGo - DeepMind, Chatgpt เป็นต้น
- **Machine Learning (ML)** เป็นกลุ่มของเทคนิคหรือศาสตร์ที่อยู่ในรากของ AI ที่เกี่ยวข้องกับการใช้ประยุกต์ใช้ทฤษฎีทางสถิติและคณิตศาสตร์เพื่อเรียนรู้หรือลักษณะของสารสนเทศจากข้อมูล สารสนเทศดังกล่าวสามารถนำมาใช้ได้หลายลักษณะ ทั้งการบรรยาย อธิบาย ทำนาย และตัดสินใจ ML ถือเป็นส่วนประกอบที่สำคัญที่สันสนับสนุนการทำงานของ AI
- **Deep Learning (DL)** เป็นแขนงย่อย (subdivision) ของ ML ที่เกี่ยวข้องกับการใช้เทคนิคที่เรียกว่าเครือข่ายประสาทเทียม (artificial neural network: ANN) ที่มีความลึกของเครือข่ายหลายชั้นเพื่อเรียนรู้หรือลักษณะของสารสนเทศจากข้อมูลและใช้ในวัตถุประสงค์หลักคือเพื่อทำนาย/จำแนกค่าสังเกตของตัวแปรตาม นอกจากนี้ลักษณะเฉพาะตัวที่โดดเด่นของ DL คือเครือข่ายประสาทเทียมที่ใช้ในการเรียนรู้นั้นถูกพัฒนาขึ้นเลียนแบบการทำงานของเซลล์เครือข่ายสมองของมนุษย์ การเรียนรู้ของเครื่องที่ใช้ DL จึงสามารถเรียนรู้ข้อมูลที่มีความซับซ้อน เช่น ข้อความ ภาพ และเสียงได้มีประสิทธิภาพมากกว่าการใช้เทคนิค ML แบบปกติ

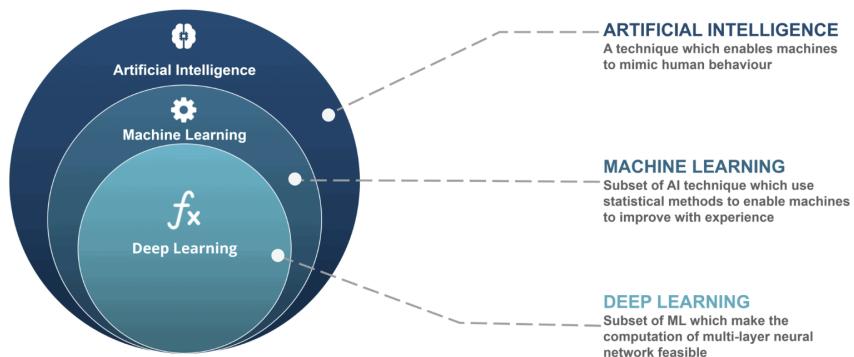


Figure 5: AI, ML และ DL (<https://k21academy.com/datascience/deep-learning/dl-vs-ml/>)

จากความหมายในข้างต้นจะเห็นว่า DL ถือเป็น machine learning ตัวหนึ่งที่ใช้ในวัตถุประสงค์เพื่อทำนายหรือจำแนกค่าสังเกตของตัวแปรตาม เมื่อเปรียบเทียบความแตกต่างระหว่าง machine learning algorithm ในกลุ่มที่ใช้สำหรับทำนาย กับ

DL มีความแตกต่างหนึ่งที่เห็นได้อย่างชัดเจนคือในส่วนของการเรียนรู้ของโมเดล ตั้งรูปด้านล่าง

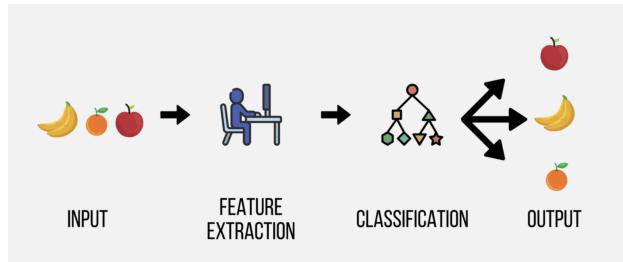


Figure 6: ML (<https://www.advancinganalytics.co.uk/blog/2021/12/15/understanding-the-difference-between-ai-ml-and-dl-using-an-incredibly-simple-example>)

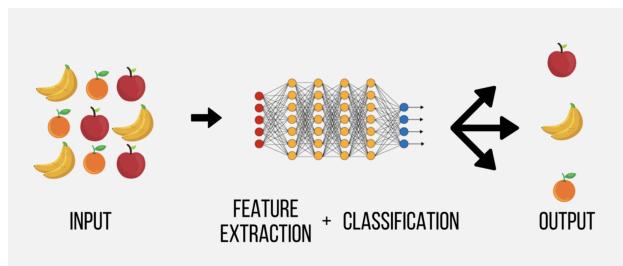


Figure 7: DL (<https://www.advancinganalytics.co.uk/blog/2021/12/15/understanding-the-difference-between-ai-ml-and-dl-using-an-incredibly-simple-example>)

## 1.4 Data Analysis vs Data Analytics

คำศัพท์ที่ควรทราบเมื่อศึกษาหัวข้อเกี่ยวกับวิทยาการข้อมูลหรือการเรียนรู้ของเครื่องคือคำว่า data analysis และ data analytics เราอาจเห็นหลายที่ใช้คำพทที่สองคิดแทนกันไปมา เช่นเดียวกับ AI, ML หรือ DL ข้างต้น แต่ในความเป็นจริงทั้งสองคำนี้มีความแตกต่างกัน แม้มีความสัมพันธ์ซึ่งกันและกัน

**Data Analysis** เป็นกระบวนการทางสถิติและวิทยาการข้อมูลที่เกี่ยวข้องกับการเก็บรวบรวมข้อมูล การจัดระเบียบและจัดกรรชทำข้อมูล และการวิเคราะห์ข้อมูล โดยการวิเคราะห์ข้อมูลนี้มีวัตถุประสงค์หลักเพื่อข้อสรุปสำหรับการบรรยายหรืออธิบาย ตัวแปรหรือความสัมพันธ์ระหว่างตัวแปรภายในประชากรที่ศึกษา มักใช้ตอบคำตามในลักษณะ

- ที่ผ่านมาเกิดอะไรขึ้น?
- สิ่งที่เกิดขึ้นนั้นเป็นอย่างไร?
- เพราะอะไรถึงเป็นแบบนั้น?

จะเห็นว่าลักษณะของ data analysis เน้นการอธิบายปรากฏการณ์หรือความเป็นไปในอดีตเป็นหลัก ตัวอย่างการใช้ Data Analysis เช่นการวิเคราะห์สถิติบรรยาย การเปรียบเทียบค่าเฉลี่ย การวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรด้วย correlation หรือ regression หรือการวิเคราะห์ความสัมพันธ์เชิงสาเหตุด้วย structural equation model เป็นต้น

**Data Analytics** เป็นกระบวนการที่ผู้เคราะห์นำข้อมูลที่มีมาให้เคราะห์ประมวลผลหรือการดำเนินงานที่เกี่ยวข้อง เพื่อสร้างเป็นแบบจำลองหรือโมเดล แล้วนำโมเดลที่สร้างขึ้นไปใช้งานในเชิงการทำงาน การตัดสินใจ หรือให้ข้อเสนอแนะจากข้อมูลสำหรับเหตุการณ์ที่จะเกิดขึ้นในอนาคต กระบวนการทำงานของ data analytic ส่วนใหญ่คล้ายกับ data analysis กล่าวคือมีส่วนของการบันทึกที่จะต้องจัดการเก็บรวบรวมข้อมูล และจัดเตรียมข้อมูลเมื่ອันกัน แต่ความแตกต่างคือส่วนของการวิเคราะห์ที่ไม่ได้เน้นการอธิบายความเป็นไปในอดีต แต่เน้นการนำโมเดลที่สร้างขึ้นไปทำนาย ช่วยคิดหรือตัดสินใจแทนมนุษย์ การวิเคราะห์ข้อมูลทางด้าน data analytics ส่วนใหญ่จะมีการใช้กลอกริทึมการเรียนรู้ของเครื่องเรียนมาใช้

## 1.5 Types of Machine Learning?

การเรียนรู้ของเครื่อง (ML) เป็นศาสตร์ย่อยแขนงหนึ่งภายใต้ศาสตร์ทางด้านสถิติและวิทยาการข้อมูล ซึ่งเกี่ยวข้องกับการใช้อัลกอริทึม (algorithms) ในการเรียนรู้/ค้นหาความรู้จากข้อมูล แล้วนำความรู้ที่ได้มาใช้งานตั้งแต่การบรรยายสภาพของข้อมูล (descriptive) การวินิจฉัย (diagnostic) เพื่อหาสาเหตุหรือปัจจัยที่ก่อให้เกิดผลลัพธ์ที่สนใจ การทำนาย (predictive) เพื่อสร้างโมเดลที่เรียนรู้ความสัมพันธ์ในข้อมูลเพื่อทำนายผลลัพธ์ของตัวแปรที่สนใจ ผลลัพธ์ที่ได้จากการทำนายนี้สามารถนำมาโมเดลเพื่อช่วยวางแผน/ตัดสินใจ (prescriptive) ดำเนินการเพื่อนำไปสู่ผลลัพธ์ที่คาดหวัง เทคนิคการเรียนรู้ของเครื่องอาจจำแนกได้เป็น 3 ประเภท ตามวัตถุประสงค์หรือความสามารถของอัลกอริทึมการเรียนรู้ ได้แก่

- การเรียนรู้ที่มีการชี้นำ (supervised learning)
- การเรียนรู้แบบไม่มีการชี้นำ (unsupervised learning)
- การเรียนรู้แบบที่มีการเสริมแรง (reinforcement learning)

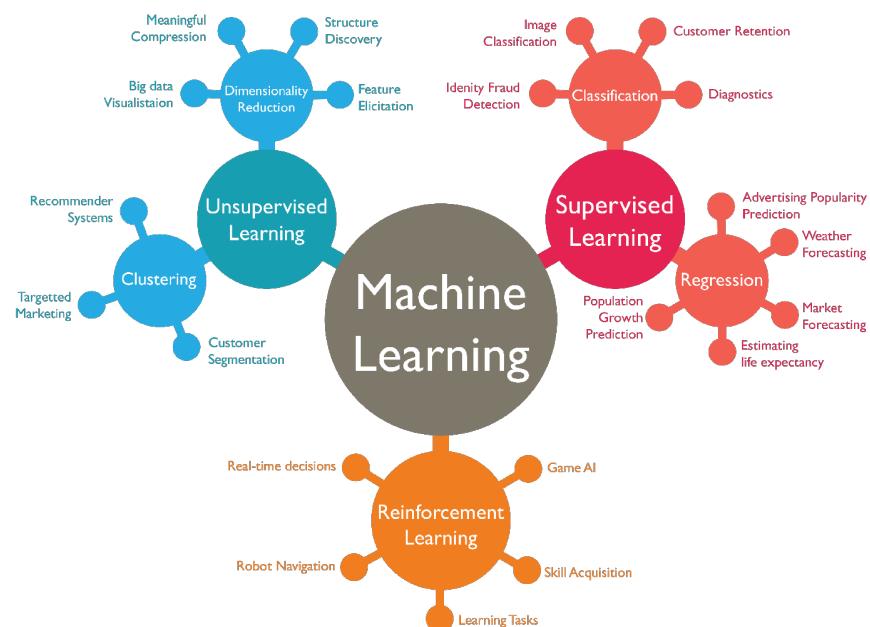


Figure 8: ประเภทของ ML

## Supervised Learning

ผู้เคราะห์จะใช้ supervised learning เมื่อมีวัตถุประสงค์ที่ต้องการสร้างโมเดลทำงาน/จำแนกค่าสังเกตของตัวแปรตามด้วยข้อมูลค่าสังเกตของตัวแปรอิสระ โดย supervised learning เป็นกลุ่มของอัลกอริทึมที่จะเรียนรู้รูปแบบความสัมพันธ์ระหว่างตัวแปรอิสระกับตัวแปรตาม และใช้รูปแบบความสัมพันธ์ที่เรียนรู้จากข้อมูลในอดีตนี้ในการทำนายข้อมูลที่ไม่ทราบค่าที่จะเกิดขึ้นในอนาคต เช่น

- การทำนายสถานะการเป็นหนี้ของลูกค้า (ลูกหนี้ขันดี ลูกหนี้ปกติ ลูกหนี้เสีย) โดยอิงกับข้อมูลล่วงตัว ข้อมูลที่เกี่ยวข้องกับเครดิตทางการเงิน และข้อมูลพฤติกรรมการดำเนินชีวิต
- ผู้พัฒนาคอสเรียนออนไลน์ใช้ supervised learning เพื่อทำนายผลการเรียนของนักเรียน หรือแนวโน้มการ drop out ของนักเรียนในคอร์สเรียน โดยอิงจากพฤติกรรมการเรียนที่แสดงในระบบการเรียนรู้ออนไลน์
- การพัฒนาระบบวินิจฉัยความยืดมั่นผูกพันของนักเรียนด้วยการรู้จำใบหน้าโดยใช้การเรียนรู้เชิงลึก



Figure 9: ลักษณะของ ML ประเภท supervised learning ([https://3.bp.blogspot.com/-occLtedKtRw/W8RVv5QyIII/AAAAAAAEBg/fdvwBPGxdfQ1izWa\\_l95-SW4kgYSMgAsgCLcBGAs/s1600](https://3.bp.blogspot.com/-occLtedKtRw/W8RVv5QyIII/AAAAAAAEBg/fdvwBPGxdfQ1izWa_l95-SW4kgYSMgAsgCLcBGAs/s1600))

การที่จะใช้ supervised learning ได้นั้นผู้เคราะห์ยังจำเป็นต้องมีชุดข้อมูลต้นแบบที่ภายในชุดข้อมูลประกอบด้วยข้อมูลของตัวแปรตามหรือผลลัพธ์ที่ต้องการทำนาย และตัวแปรอิสระหรือข้อมูลที่จะใช้เป็นตัวทำนายผลลัพธ์ที่ต้องการตั้งกล่าว ในเชิงเทคนิคจะเรียกชุดข้อมูลต้นแบบดังกล่าวว่า ชุดข้อมูลฝึกหัด (training dataset) นอกจากนี้ supervised learning ยังจำแนกเป็นประเภทย่อยได้อีก 2 ประเภทตามลักษณะของตัวแปรตาม ได้แก่ regression และ classification

- **Regression** เป็นโมเดลสำหรับทำนายตัวแปรตามเชิงปริมาณ
- **Classification** เป็นโมเดลสำหรับทำนายตัวแปรตามแบบจัดประเภท

## Unsupervised Learning

ภาษาไทยอาจใช้คำว่าการเรียนรู้แบบไม่มีการชี้นำ การเรียนรู้ประเภทนี้มีความแตกต่างจาก supervised learning กล่าวคือ ชุดข้อมูลฝึกหัดไม่จำเป็นต้องมีค่าสังเกตของตัวแปรตาม และวัตถุประสงค์ของการใช้ unsupervised learning คือการสร้างหรือสกัดสารสนเทศออกจากข้อมูล ซึ่งอาจจำแนกได้เป็น การจัดกลุ่ม (clustering) และการหาความสัมพันธ์ (association)

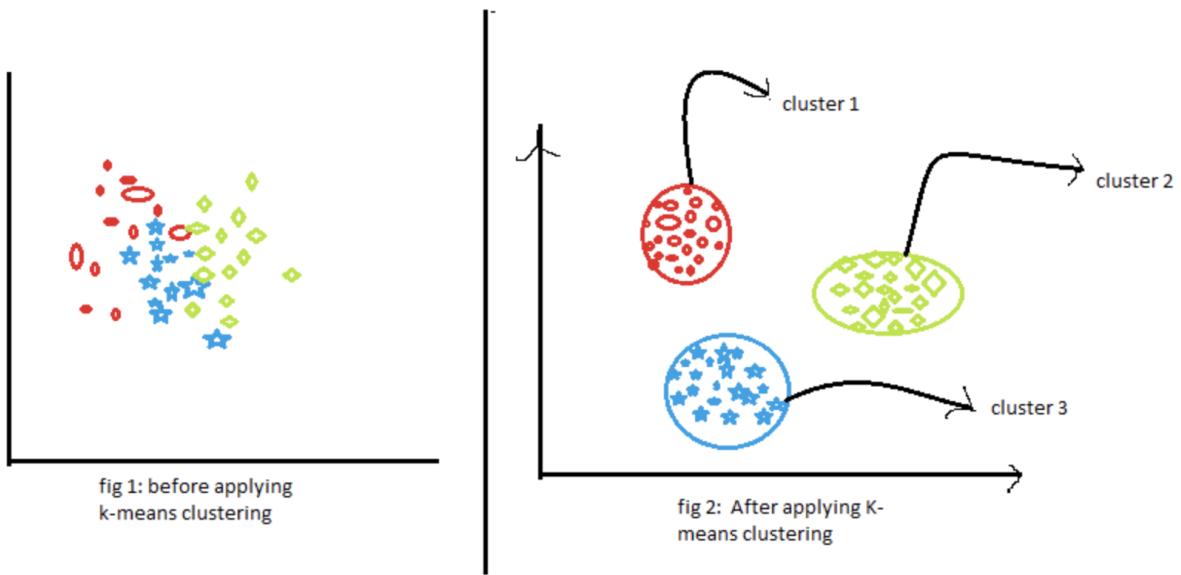


Figure 10: ลักษณะของการ clustering

### Reinforcement Learning

เป็นอัลกอริทึมการเรียนรู้ (เรียกว่า agent) ที่เรียนรู้ด้วยการใช้ feedback ที่มีการให้รางวัล (reward) เมื่ออัลกอริทึมสามารถทำงานได้สำเร็จ และมีการทำโทษ (punishments) เมื่อล้มเหลว ผู้พัฒนาอัลกอริทึมประเภทนี้จะให้ agent ทำการเรียนรู้งานที่การให้ feedback ดังกล่าวแบบทวนซ้ำจนกระทั่งอัลกอริทึมสามารถทำงานที่กำหนดได้อย่างมีประสิทธิภาพตามที่ต้องการ

- <https://www.youtube.com/watch?v=ldXxDNjS5jw>
- <https://www.youtube.com/watch?v=n2gE7n11h1Y>
- <https://www.youtube.com/watch?v=2tamH76Tjvw>

## 1.6 ตัวอย่างการประยุกต์ใช้ ML ทางการศึกษา (การบ้าน)

ขอให้นิสิตสีบคันงานวิจัยทางการศึกษาที่มีการใช้ machine learning จากนั้นสรุปสาระสำคัญจากงานวิจัยดังกล่าวส่งเป็นการบ้านชิ้นที่ 1 กำหนดส่ง 25 มกราคม 2566 โดยรายงานสรุปที่จะส่งขอให้มีความยาวไม่เกิน 2 หน้า A4 โดยมีรายละเอียดครอบคลุมหัวข้อดังนี้

- ชื่องานวิจัย
- ความเป็นมา หรือ motivation ของงานวิจัย
- วัตถุประสงค์ของการวิจัย
- กลุ่มเป้าหมาย
- ตัวแปรและข้อมูลที่ใช้ในการวิจัย
- อัลгорิทึมการเรียนรู้ของเครื่องที่ใช้ในการวิจัย
- ผลการวิจัยที่สำคัญ
- จุดเด่นและข้อสังเกตของการวิจัย

## บทที่ 2 ความรู้พื้นฐานในการพัฒนา Supervised Learning

Supervised learning เป็นการเรียนรู้ของเครื่องที่มีการนำมาประยุกต์ใช้งานอย่างแพร่หลาย บทเรียนนี้จะกล่าวถึงโมเดลที่สำคัญของการบวนการพัฒนา supervised learning model ทั้งในกลุ่มของ regression และ classification models

### 2.1 ติดตั้ง R และ RStudio



เครื่องมือที่สามารถใช้จัดการข้อมูลและพัฒนาโมเดลการเรียนรู้ของเครื่องในปัจจุบันมีหลายตัว R เป็นโปรแกรมภาษาหนึ่งที่มีความสามารถสูงในการพัฒนาโมเดลการเรียนรู้ของเครื่องดังกล่าว ก่อนจะไปสู่บทเรียนในหัวข้อด้านไป ขอให้ผู้อ่านดาวน์โหลด และติดตั้ง R และ RStudio IDE ก่อน โดยดาวน์โหลด R ที่นี่ [และดาวน์โหลด Rstudio ที่นี่](#)

ปัจจุบัน RStudio เป็น IDE ที่ค่อนข้างครบเครื่องสำหรับผู้ใช้ R และสามารถใช้เขียนภาษาอื่น ๆ ได้หลายตัวทั้ง Python, HTML, Markdown, Stan เป็นต้น ดังนั้นในกรณีที่ต้องการใช้ Python อาจใช้บน RStudio เลยก็ได้ อย่างไรก็ตามการใช้งาน Python บน RStudio เป็นการใช้งานผ่านตัวแปลงคือ package reticulate จึงอาจทำให้การประมวลผลซักก่าวการทำงานบน Editor ของ Python โดยเฉพาะ

### 2.2 ติดตั้ง Anaconda



Anaconda เป็นชุดโปรแกรมที่รวมความ package การทำงานสำหรับนักสถิติและวิทยาการข้อมูล โดยมีทั้งเครื่องมือสำหรับภาษา R และ Python โดย Anaconda จะรวม package ที่เกี่ยวข้องสำหรับการทำงานทางด้านวิทยาการข้อมูลเอาไว้มากกว่า 1,000 ตัว และสามารถติดตั้งได้ทั้งบน Windows, Linux และ Mac OS ซึ่งช่วยอำนวยความสะดวกให้กับผู้ใช้โดยไม่ต้องดาวน์โหลดและติดตั้ง package พื้นฐานที่ละเอียดอ่อน เช่น pandas, numpy, matplotlib และ seaborn ข้อควรระวังคือ Anaconda นั้นรวม package เอาไว้เป็นจำนวนมากมากการติดตั้งจึงค่อนข้างเปลืองพื้นที่ HDD ของเครื่องคอมพิวเตอร์สำหรับผู้ที่มีพื้นที่จำกัดอาจเลี่ยงไปติดตั้ง Miniconda หรือใช้ Python บน Google Colab ขั้นตอนการติดตั้ง Anaconda ให้ดำเนินการดังนี้

1. ดาวน์โหลดและติดตั้ง Anaconda ได้ที่ <https://www.anaconda.com/>
2. ดำเนินการติดตั้งตามด้วยขั้นตอนการติดตั้ง
3. ถ้ามีการถามว่าต้องการติดตั้ง VS Code หรือไม่ สามารถเลือกติดตั้งหรือไม่ก็ได้ แต่แนะนำว่าติดตั้งไว้ดีกว่า เพราะมีประโยชน์ในการเขียน Python
4. เมื่อติดตั้งเสร็จให้กดเปิด Anaconda Navigator ขึ้นมา หน้าต่างนี้จะเหมือนเป็น Hub ที่รวมโปรแกรมต่าง ๆ สำหรับการทำงาน เช่น Jupyter Notebook, JupyterLab ซึ่งเป็น gen ใหม่ของ Jupyter Notebook, VS Code ซึ่งเราสามารถใช้ Editor พากน์ในการเขียน Python ได้

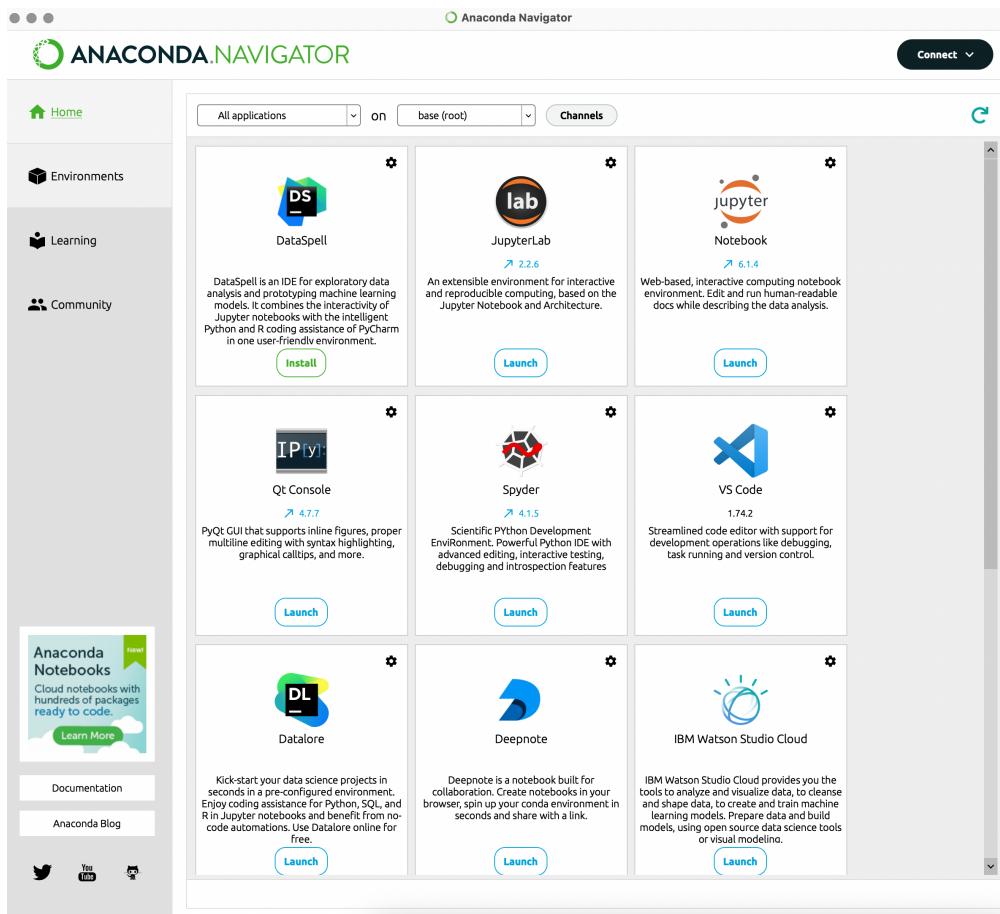


Figure 11: Anaconda Navigator

## 2.3 กิจกรรม My First Regression

จากตัวอย่างข้อมูลของกิจกรรมในหัวข้อ 1.2 ([1.2 กิจกรรม Rule-based algorithm](#)) ในหัวข้อนี้เราจะสร้างโมเดลทำนายตัวใหม่โดยจะใช้อัลกอริทึมการเรียนรู้ของเครื่องเป็นเครื่องมือในการพัฒนา จากกิจกรรม 1.2 เราทราบแล้วว่าความสัมพันธ์ระหว่างตัวแปรในชุดข้อมูลเป็นความสัมพันธ์เชิงเส้นตรง ดังนั้นกิจกรรมนี้เราจะใช้อัลกอริทึม linear regression ในการพัฒนาโมเดลทำนาย

อัลกอริทึม linear regression เป็นอัลกอริทึมทางสถิติที่ใช้สำหรับทำนายแนวโน้มค่าสังเกตของตัวแปรตามที่ไม่ทราบค่าโดยอิงกับค่าสังเกตของตัวแปรอิสระที่ทราบค่า การเรียนรู้ของ linear regression จะพยายามสร้างสมการเส้นตรงที่ดีที่สุด (best linear equation) ที่สามารถใช้เป็นตัวแทนความสัมพันธ์ระหว่างตัวแปรตามกับตัวแปรอิสระที่พบรูปในชุดข้อมูล ในเชิงเทคนิคการทำสมการเส้นตรงด้วยอัลกอริทึมนี้เป็นการเพื่อนค่าหาของพารามิเตอร์ภายในสมการเส้นตรง ได้แก่ พารามิเตอร์จุดตัดแกน y และพารามิเตอร์ความชัน ที่ทำให้สมการเส้นตรงมีความคลาดเคลื่อนในการทำนายต่ำที่สุด โดยที่ความคลาดเคลื่อนในการทำนายดังกล่าวคำนวณจากค่าผลรวมของความคลาดเคลื่อนกำลังสอง (sum squared error: SSE) สามารถคำนวณได้ดังนี้

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

เมื่อ  $\hat{y}_i$  เป็นผลรวมเชิงเส้น (linear combination) ของตัวแปรอิสระที่ใช้ทำนายตัวแปรตาม

### Fitting Linear Regression using lm()

จากข้อมูลในกิจกรรม 1 สามารถใช้อัลกอริทึม linear regression เพื่อหาโมเดลทำนายที่เหมาะสมด้วยโปรแกรม R ได้ดังนี้

```
# import data
x<-c(0,1,2,3,4,5)
y<-c(1,2,2,3,4,4)
dat<-data.frame behav = x, score = y)
# estimate linear regression model
fit_linear<-lm(y~x, data=dat)
summary(fit_linear)

##
## Call:
## lm(formula = y ~ x, data = dat)
##
## Residuals:
##      1       2       3       4       5       6 
## -0.09524  0.27619 -0.35238  0.01905  0.39048 -0.23810 
##
## Coefficients:
```

```

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.09524   0.23425   4.675  0.00948 **
## x           0.62857   0.07737   8.124  0.00125 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3237 on 4 degrees of freedom
## Multiple R-squared:  0.9429, Adjusted R-squared:  0.9286
## F-statistic: 66 on 1 and 4 DF,  p-value: 0.001249

```

## Evaluation Metrics

วัดคุณประสิทธิภาพของการพัฒนาโมเดลทำนายข้างต้นคือเพื่อทำนายคะแนนสอบด้วยพฤติกรรมการเรียนของนักเรียน การจะนำโมเดลดังกล่าวไปใช้จึงจำเป็นจะต้องตรวจสอบประสิทธิภาพการทำนายของโมเดลดังกล่าวก่อน

เนื่องจากโมเดลนี้เป็น supervised learning model ประเภท regression การตรวจสอบประสิทธิภาพการทำนายจึงควรเป็นการเปรียบเทียบความสอดคล้องกันระหว่างค่าทำนายกับค่าจริงของ score ในเชิงเทคนิคไม่เกณฑ์พิจารณาที่เรียกว่า evaluation metric สำหรับ regression model อยู่หลายตัว โดยในบทเรียนนี้จะกล่าวถึง 2 ตัวได้แก่ RMSE และ R squared รายละเอียดมีดังนี้

**Root Mean Squared Error (RMSE)** ในทางทฤษฎีค่า RMSE มีความหมายเป็นค่าคาดคะเนในการทำนายโดยเฉลี่ยของโมเดล สามารถคำนวณได้จากสูตร

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

**Coefficient of Determination (R squared)** ส่วน R squared มีความหมายเป็นสัดส่วนของความผันแปรที่ร่วมกันระหว่างค่าจริงของตัวแปรตามกับค่าทำนายของตัวแปรตามที่ได้จากการทำนาย การคำนวณค่า R squared สามารถทำได้ง่าย ๆ ด้วยการทำค่ากำลังสองของสัมประสิทธิ์สหสัมพันธ์ระหว่างค่าจริงกับค่าทำนายของตัวแปรตามดังกล่าว

$$R^2 = Corr(y, \hat{y})^2$$

จะเห็นว่า evaluation metric ทั้งสองล้วนเป็นการเปรียบเทียบความแตกต่างหรือความสอดคล้องระหว่างค่าจริงของตัวแปรตามกับค่าทำนายที่ได้จากการทำนาย การคำนวณค่าของ metric ดังกล่าวสามารถเขียนคำสั่งใน R ได้ดังนี้

```

# calculate prediction values
pred<-predict(fit_linear, newdata = dat)
pred # predicted value

```

```

##      1      2      3      4      5      6
## 1.095238 1.723810 2.352381 2.980952 3.609524 4.238095

# calculate rmse value
sqrt(mean((y-pred)^2)) #rmse

## [1] 0.264275

# calculate r squared value
cor(pred, y)^2 #rsq

## [1] 0.9428571

```

**R squared plot** R squared plot เป็นแผนภาพการกระจายที่ plot ระหว่างค่าท่านาย (แกน Y) กับค่าจริง (แกน X) และเปรียบเทียบแนวโน้มของคู่อันดับกับกล่าวกับเส้นตรงอ้างอิง  $y = x$

```

# create R squared plot
plot(y, pred, pch=16, xlab = "actual score", ylab="predicted score")
abline(a=0, b=1, lty=3, col="steelblue")

```

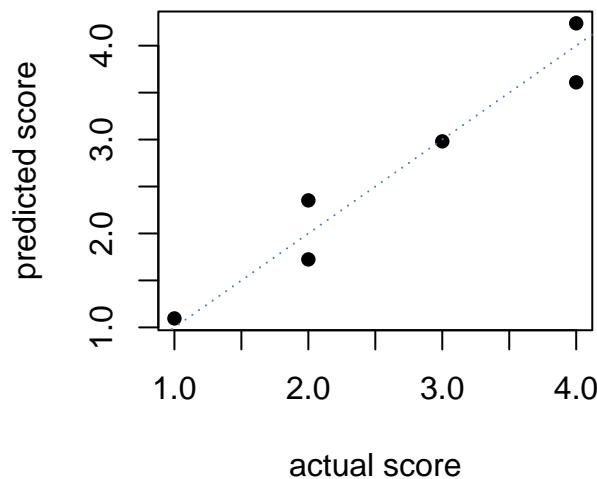


Figure 12: R Square plot

ผลการพัฒนา linear regression ข้างต้นได้ข้อสรุปว่าโมเดลท่านายที่พัฒนาขึ้นมีประสิทธิภาพสูงมากในการท่านายคะแนนของนักเรียน (ผู้อ่านเห็นด้วยกับข้อความนี้หรือไม่? เพราะเหตุใด?)

## 2.4 กิจกรรม My First Classification

กิจกรรมนี้จะแสดงตัวอย่างการสร้างโมเดลจำแนก (classification model) ஆட்சமூலத்தீடுபேர்ந்தான்போய்கிடைக்கிறது மூலம் mmreg.sav இல்லை என்பதை விடுவது ஆட்சமூலத்தீடுபேர்ந்தான்போய்கிடைக்கிறது மூலம் mmreg.sav இல்லை என்பதை விடுவது

กิจกรรมนี้จะสร้างโมเดลจำแนกผลการเรียนวิชาคณิตศาสตร์ (สอบผ่าน vs สอบตก) โดยใช้ตัวแปรอิสระได้แก่ เพศ (female) และแรงจูงใจในการเรียน (motivation) ทั้งนี้ก่อนการดำเนินการขอให้ผู้อ่านแปลงค่าของตัวแปร math ให้เป็นแบบ binary response ก่อน โดยใช้คำสั่งดังนี้

```
# importing mmreg.sav into dat
library(haven)
dat <- read_spss("mmreg.sav")
# manipulating response variable "math"
dat <- dat%>%mutate(
  math_binary = factor(ifelse(math>50,1,0),
                        labels=c("fail","pass"))
)
table(dat$math_binary)

##  
## fail pass  
## 266 334
```

### Fitting Logistic Regression Model using glm()

สำหรับอัลกอริทึมการเรียนรู้ที่จะใช้ในกิจกรรมนี้จะเลือกใช้ logistic regression ซึ่งเป็นอัลกอริทึมพื้นฐานที่เรียนในรายวิชาสถิติพื้นฐานแล้ว การ fit logistic regression บน R สามารถทำได้หลายวิธีการ โดยมีขั้นตอนการดำเนินงานที่คล้ายกับการ fit linear regression โดยจะใช้ฟังก์ชัน glm() แทน lm() คำสั่งต่อไปนี้แสดงการ fit binary logistic regression ด้วยโปรแกรม R จากஆட்சமூலத்தீடு

```
fit_logistic <- glm(math_binary ~ female + motivation,
                      data = dat,
                      family = "binomial")
summary(fit_logistic)

##  
## Call:  
## glm(formula = math_binary ~ female + motivation, family = "binomial",  
##       data = dat)  
##  
## Deviance Residuals:
```

```

##      Min       1Q    Median       3Q      Max
## -1.4960 -1.2290   0.8893   1.0177   1.4228
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.2931    0.1951  -1.502   0.133
## female      -0.2674    0.1691  -1.582   0.114
## motivation  1.0166    0.2468   4.119  3.8e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 824.05 on 599 degrees of freedom
## Residual deviance: 805.30 on 597 degrees of freedom
## AIC: 811.3
##
## Number of Fisher Scoring iterations: 4

```

สมการลดด้อยแบบ logistic ในข้างต้นสามารถเขียนได้ดังนี้

$$P(\text{math} = 1 | \text{female}, \text{motivation}) = \frac{\exp(-0.2931 + -0.2674\text{female} + 1.0166\text{motivation})}{1 + \exp(-0.2931 + -0.2674\text{female} + 1.0166\text{motivation})}$$

แสดงว่าค่าที่คำนวณของสมการข้างต้นเป็นค่าความน่าจะเป็นที่จะสอบผ่านเมื่อกำหนดค่าของตัวแปรอิสระ การจะใช้มีเดลดังกล่าวเพื่อจำแนกว่านักเรียนแต่ละคนมีแนวโน้มจะสอบผ่านหรือตกจะต้องแปลงค่าความน่าจะเป็นดังกล่าวให้เป็นประเภทหรือค่าสังเกตของตัวแปรตามแบบจัดประเภท การแปลงค่าความน่าจะเป็นนี้สามารถทำได้โดยกำหนดคะแนนจุดตัดของค่าความน่าจะเป็นสำหรับแบ่งประเภทของหน่วยข้อมูลระหว่างสอบผ่านและสอบตก (ใช้เทคนิคเรียกว่าค่า threshold)

สมมุติว่ากำหนดค่า threshold = 0.5 สามารถคำนวณค่าที่คำนวณผลการสอบรายวิชาคณิตศาสตร์ได้ดังนี้

```

# calculate predicted probabilities
pred_prob <- predict(fit_logistic,
                      type="response",
                      newdata = dat)

summary(pred_prob)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##  0.3634  0.5106  0.5958  0.5567  0.6121  0.6734

```

```

# calculate predicted response values
pred_val <- factor(ifelse(pred_prob>0.5,"pass","fail"))

# assign predicted response values into dat dataframe
dat$pred_prob <- pred_prob
dat$pred_val <- pred_val
table(dat$pred_val)

##

## fail pass
## 120 480

```

## Evaluation Metrics

Evaluation metric สำหรับ classification model มีหลายตัว ในหัวข้อนี้จะกล่าวถึงบางตัวก่อน

**Confusion Matrix** confusion matrix ที่เป็นเมตริกซ์เปรียบเทียบระหว่างค่าทำนายกับค่าจริงดังด้านล่าง

		Actual	
		Yes	No
Prediction	Yes	True positive	False positive
	No	False negative	True negative

Figure 13: confusion matrix

การคำนวณ confusion matrix ใน R สามารถดำเนินการได้หลายวิธีการ วิธีการหนึ่งคือการใช้ฟังก์ชัน `table()` เพื่อแจกแจงความถี่ระหว่างค่าทำนายกับค่าจริง ดังนี้

```

table(dat$pred_val, dat$math_binary)

##

##      fail pass
##  fail    65   55
##  pass   201  279

```

จาก confusion matrix ที่คำนวณได้ข้างต้นสามารถแปลผลได้อย่างไร

**Accuracy (Hit Rate)** บาง package อาจเรียกว่า balance accuracy ดัชนีตัวนี้เป็นค่าปัจจุบันที่ความแม่นยำในการทำนายโดยตรงของโมเดลโดยมีค่าเท่ากับสัดส่วนของเดสที่ทำนายถูกทั้งหมดเทียบกับเดสทั้งหมดที่ทำนาย ดังนี้

$$Accuracy = \frac{TP + TN}{Total}$$

**Sensitivity and Specificity** sensitivity เป็นประสิทธิภาพของโมเดลในด้านความไวหรืออำนาจในการตรวจจับเดสที่  $y = 1$  ส่วน specificity เป็นประสิทธิภาพของโมเดลในการทำนายเดสที่  $y = 0$  ดัชนีทั้งสองตัวสามารถคำนวณได้ดังนี้

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{FP + TN}$$

**Prevalence** เป็นค่าความชุกของเดสที่  $y = 1$  ในกรณีที่ค่า prevalence มีค่าต่ำ ปัจจุบันไม่สามารถทำนายจากเกิดปัญหา Imbalance class กล่าวคือโมเดลมีชุดข้อมูลฝึกหัดสำหรับเดส  $y = 1$  ที่อาจน้อยเกินไป

$$Prevalence = \frac{TP + TN}{Total}$$

**Precision** เป็นความน่าเชื่อถือของผลการทำนายเดส  $y=1$  สามารถคำนวณได้จาก

$$Precision = \frac{TP}{TP + FP}$$

การคำนวณดัชนีข้างต้นด้วยมือค่อนข้างวุ่นวายใน R มี package หลายตัวที่ช่วยคำนวณดัชนีข้างต้นได้ ในที่นี้จะให้พังก์ชัน confusionMatrix() จาก package caret ดังนี้

```
# install.package("caret")
library(caret)
confusionMatrix(data = dat$pred_val,
                 reference = dat$math_binary,
                 positive = "pass")

## Confusion Matrix and Statistics
##
##          Reference
## Prediction fail pass
##      fail     65    55
##      pass    201   279
```

```

## 
##          Accuracy : 0.5733
##          95% CI : (0.5326, 0.6133)
##      No Information Rate : 0.5567
##      P-Value [Acc > NIR] : 0.2177
##
##          Kappa : 0.0844
##
##  Mcnemar's Test P-Value : <2e-16
##
##          Sensitivity : 0.8353
##          Specificity : 0.2444
##          Pos Pred Value : 0.5812
##          Neg Pred Value : 0.5417
##          Prevalence : 0.5567
##          Detection Rate : 0.4650
##          Detection Prevalence : 0.8000
##          Balanced Accuracy : 0.5398
##
##          'Positive' Class : pass
##

```

### คำถามท้ายบท

- ผลการพัฒนา binary logistic regression ในข้างต้นผู้อ่านคิดว่าโมเดลที่พัฒนาขึ้นมีประสิทธิภาพในการทำนายเป็นอย่างไร?
- จากระบวนการที่ใช้พัฒนาทั้ง linear regression model และ logistic regression model ผู้อ่านคิดว่าเป็นกระบวนการที่เหมาะสมสมสำหรับการพัฒนาโมเดลทำนายแล้วหรือไม่ หรือมีข้อสังเกตอะไร

## บทที่ 3 กระบวนการพัฒนา Machine Learning Models

กระบวนการพัฒนา machine learning models อาจจำแนกได้เป็นสองส่วน ส่วนแรกคือส่วนของการพัฒนาโมเดล และส่วนที่สองคือส่วนของการนำโมเดลไปใช้งาน

ส่วนการพัฒนาโมเดลมีกระบวนการพัฒนาแสดงไว้ในรูป 14 จากรูปจะเห็นว่าเริ่มตั้งแต่การเก็บรวบรวมข้อมูล การจัดการข้อมูล จากนั้นจะมีการแบ่งชุดข้อมูลออกเป็นส่วน training dataset และ test dataset ข้อมูลส่วนที่เป็น training dataset จะถูกนำมาพัฒนาการเรียนรู้/ความสามารถในการทำนายของโมเดล ผ่านอัลกอริทึมต่าง ๆ จากนั้นจะเลือกอัลกอริทึมที่มีประสิทธิภาพสูงที่สุด (ประเมินจาก test dataset) ไปใช้งาน ทั้งนี้การจะนำโมเดลทำนายไปใช้งานได้นั้นจะต้องมีบันทึกโมเดลที่พัฒนาขึ้นเพื่อนำไปเรียกใช้ต่อไป

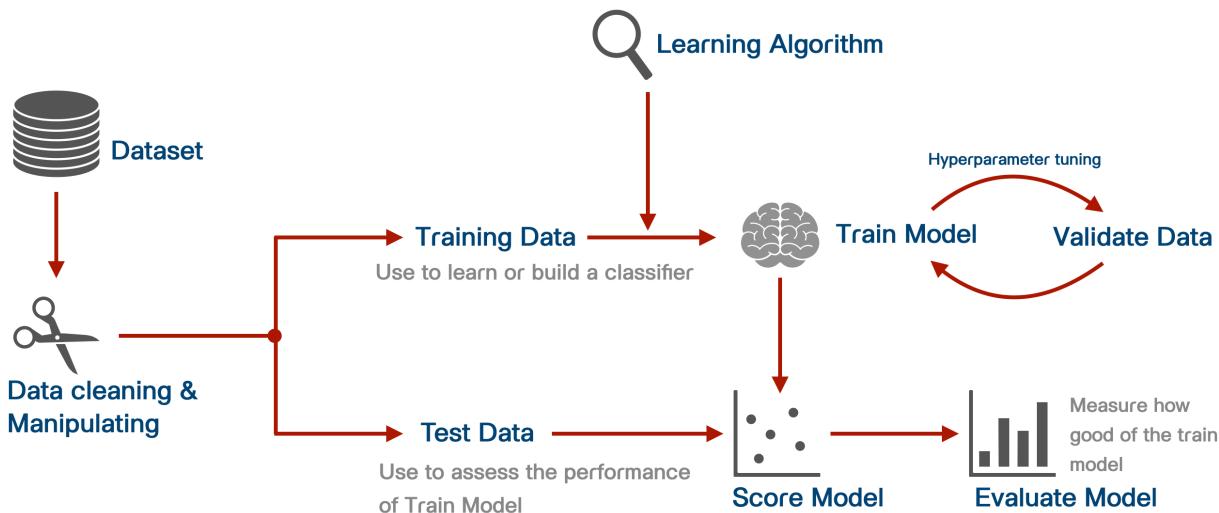


Figure 14: Modeling Process

ส่วนของการนำโมเดลไปใช้งาน ต้องทำความเข้าใจว่า ML model ที่พัฒนาขึ้นนั้นเป็นเหมือนสมองซึ่งต้องเข้าไปอยู่ในโปรแกรมหรือ application สักตัวหนึ่ง โดยโปรแกรมดังกล่าวจะมีหน้าที่หลัก ๆ เช่นมี interface สำหรับรับข้อมูลจากผู้ใช้ เพื่อนำไปประมวลผลด้วยโมเดลการทำนายที่เราสร้างขึ้น และรายงานผลการทำนาย/จำแนก ที่ได้จากโมเดลให้แก่ผู้ใช้ ทั้งนี้โปรแกรมดังกล่าวอาจจะมีล้วนอื่น ๆ ที่ยอมให้ผู้ใช้เลือกหรือปรับแต่งการทำงานของโปรแกรมได้ ทั้งนี้ขึ้นอยู่กับวัตถุประสงค์การใช้งาน ตัวอย่างด้านล่างแสดงโปรแกรมทำนายผลการเรียนของนิสิต KruRooTeller

เนื้อหาส่วนที่เหลือของบทเรียนนี้จะกล่าวถึงกระบวนการในส่วนของการพัฒนา supervised learning models โดยจะกล่าวถึงโมเดลที่จำเป็นก่อน จากนั้นจึงกล่าวถึงกระบวนการพัฒนาโมเดลดังกล่าว รายละเอียดมีดังนี้



### 3.1 Bias and Variance in ML models

พิจารณากราฟ 16 แสดงการ fit โมเดลท่านาย 3 แบบกับชุดข้อมูลฝึกหัดชุดหนึ่ง จะเห็นว่าแต่ละโมเดลมีความสามารถในการเรียนรู้ความสัมพันธ์ที่เกิดขึ้นในชุดข้อมูลแตกต่างกัน ความแตกต่างระหว่างค่าจริงของตัวแปรตามในชุดข้อมูลฝึกหัดกับค่าท่านายที่ได้จากโมเดล เรียกว่า **ความลำเอียง (bias)** จากกราฟ 16 ผู้อ่านคิดว่าโมเดลใดที่มีประสิทธิภาพในการท่านายสูงที่สุด เพราะเหตุใด ?

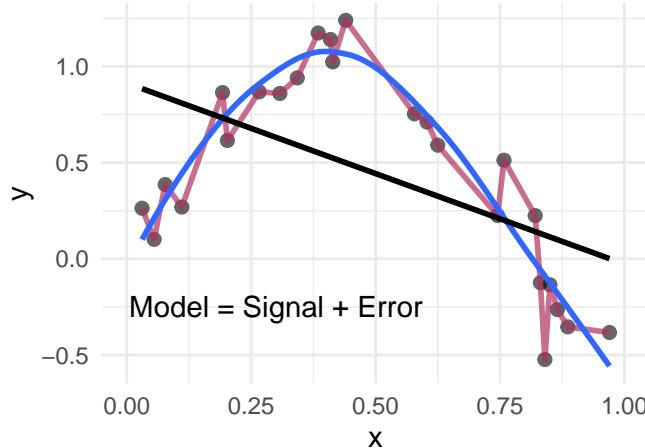


Figure 16: regression model on training dataset

พิจารณากราฟ 17 ผู้วิเคราะห์ได้นำโมเดลท่านายทั้ง 3 แบบ ที่พัฒนาจากชุดข้อมูลฝึกหัดมาใช้ท่านายข้อมูลใหม่ที่ไม่ได้ในชุดข้อมูลใหม่ (หรือชุดข้อมูลที่ไม่ได้ใช้ในขั้นการพัฒนาโมเดล) กับค่าท่านายของโมเดล เรียกว่า **ความแปรปรวน (variance)** ความแปรปรวนของโมเดลทั้ง 3 เป็นอย่างไร?

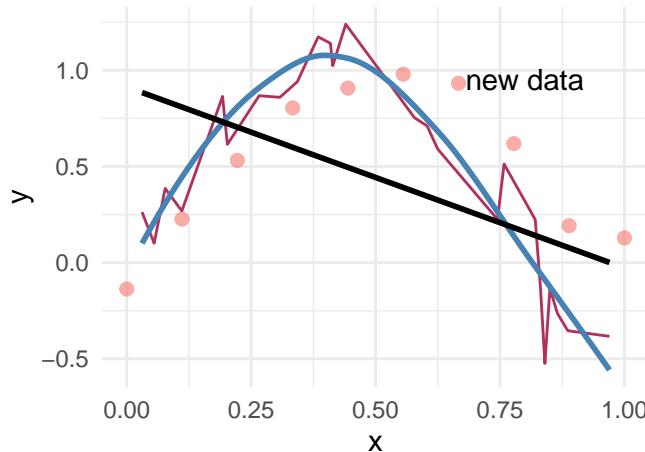


Figure 17: regression model on new dataset

จากตัวอย่างข้างต้นจะเห็นว่า ถึงแม้จะสามารถพัฒนาโมเดลท่านายให้สามารถเรียนรู้ความสัมพันธ์ภายในชุดข้อมูลฝึกหัดได้เป็นอย่างดี (มีความลำเอียงต่ำที่สุดแล้ว) แต่ก็ไม่ใช่เงื่อนไขเพียงพอที่จะสรุปได้ว่าโมเดลท่านายดังกล่าวจะมีประสิทธิภาพในการท่านายได้ดีในกรณีทั่วไป การตรวจสอบอีกชั้นหนึ่งคือตรวจสอบความแปรปรวนของโมเดลท่านาย โดยนำโมเดลดังกล่าว

ไปทำนายชุดข้อมูลที่ไม่เคยได้เรียนรู้มาก่อน โมเดลที่มีทั้งความลำเอียงและความแปรปรวนต่ำจึงเป็นโมเดลที่มีประสิทธิภาพที่จะนำไปใช้ในการนี้ทั่วไป

ในเชิงอุดมคติ ผู้วิเคราะห์ต้องการให้ทั้งความลำเอียง และความแปรปรวนมีค่าต่ำที่สุดเท่าที่จะสามารถตัดได้ แต่ในความเป็นจริงความคลาดเคลื่อนทั้งสองไม่ควบคุมให้ต่ำที่สุดพร้อมกันได้ (เพราะอะไร?) รูป 18 ด้านล่างแสดงความสัมพันธ์ระหว่างความลำเอียง และความแปรปรวน ซึ่งจะเห็นว่ามีการแปรผันซึ่งกันและกัน โมเดลที่มีความลำเอียงสูงมีแนวโน้มที่จะมีความแปรปรวนต่ำ และในทางกลับกันโมเดลที่มีความลำเอียงต่ำจะมีแนวโน้มที่มีความแปรปรวนสูง ดังนั้นวัตถุประสงค์ของการพัฒนาโมเดลจึงเป็นการหาจุดที่ดีที่สุดที่ทำให้ความคลาดเคลื่อนทั้งสองอยู่ในจุดที่ต่ำที่สุดเท่าที่จะเป็นไปได้

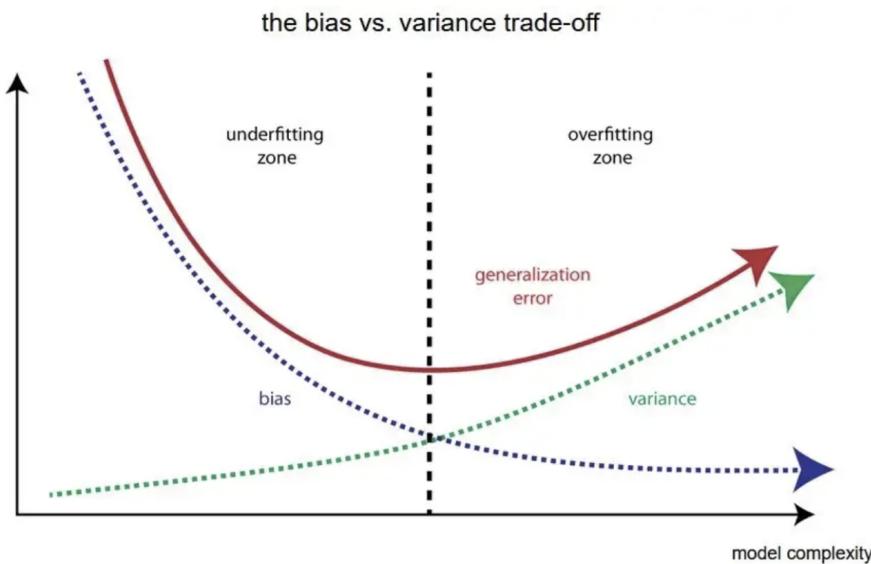


Figure 18: bias and variance trace-off

### 3.2 Underfitting, Overfitting และ Good fit models

หากจำแนกโมเดลทำนายที่ถูกพัฒนาขึ้นตามประสิทธิภาพการทำนายของโมเดล อาจจำแนกได้เป็น 3 ประเภท ดังในรูป 19 ได้แก่

- underfitting models คือโมเดลที่มีความลำเอียงสูง
- overfitting models คือโมเดลที่มีความแปรปรวนสูง
- good fit models คือโมเดลที่สามารถสมดุลความลำเอียงและความแปรปรวนให้มีค่าต่ำที่สุดเท่าที่จะเป็นไปได้

### 3.3 Training, validation, and Test Dataset

จาก concept ข้างต้นจะเห็นว่าในกระบวนการพัฒนาโมเดลผู้วิเคราะห์จะให้ความสำคัญกับประสิทธิภาพในการทำนายของโมเดลเฉพาะด้านความลำเอียงไม่ได้ ยังต้องคำนึงถึงด้านความแปรปรวนด้วย การพัฒนาโมเดลการเรียนรู้ของเครื่องจึงจะมีแค่ชุดข้อมูลฝึกหัดไม่ได้ ยังต้องมีชุดข้อมูลอีกชุดหนึ่งเพื่อเอาไว้ตรวจสอบความแปรปรวนของโมเดลด้วย ในเชิงเทคนิคเรียกชุดข้อมูลนี้ว่า ชุดข้อมูลทดสอบ (test dataset)

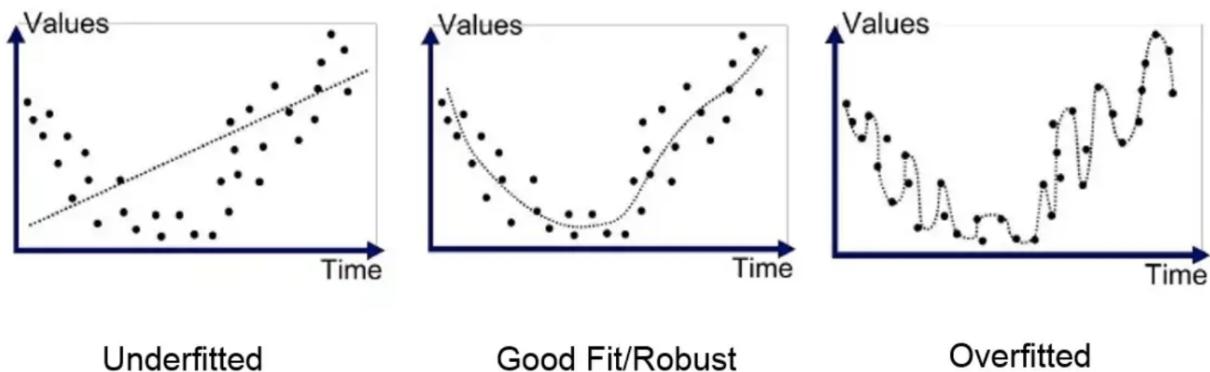


Figure 19: underfitting, good fit, and overfitting model

ภายในอัลกอริทึม supervised learning จะมีส่วนประกอบหลัก ๆ ได้แก่ อัลกอริทึม พารามิเตอร์ และ ไฮเปอร์พารามิเตอร์

- **อัลกอริทึม** เป็นส่วนของการเรียนรู้ของสำหรับแต่ละการเรียนรู้ของเครื่องที่ใช้ในการเรียนรู้หรือสกัดสารสนเทศจากข้อมูลในชุดข้อมูลฝึกหัด
- **พารามิเตอร์ (parameters)** ส่วนที่ทำให้การเรียนรู้ของเครื่อง fit กับข้อมูล กล่าวง่าย ๆ คือค่าของพารามิเตอร์ที่เปลี่ยนแปลงไป จะทำให้รูปแบบการเรียนรู้มีการเปลี่ยนไป ค่าพารามิเตอร์นี้สามารถประมาณได้จากข้อมูลด้วยวิธีการทางสถิติ/คลนิตศาสตร์ ตัวอย่างของพารามิเตอร์ เช่น ใน linear regression model มีพารามิเตอร์คือ สัมประสิทธิ์จุดตัดแกน และสัมประสิทธิ์ความชัน เป็นต้น อย่างไรก็ตามบางอัลกอริทึมไม่ได้มีพารามิเตอร์ของโมเดล เช่น K-NN เป็นต้น
- **ไฮเปอร์พารามิเตอร์ (Hyperparameters)** เป็นพารามิเตอร์ประเภทหนึ่งในอัลกอริทึมการเรียนรู้ของเครื่อง พารามิเตอร์ประเภทนี้ไม่สามารถประมาณค่าจากข้อมูลโดยตรงด้วยวิธีการทางสถิติ แต่จะใช้การกำหนด/ปรับแต่งค่าโดยตัวผู้ใช้เครื่อง ในการเขิงเทคนิคเรียกวิธีการปรับแต่งค่าดังกล่าวว่า **hyperparameter tuning** การปรับแต่งค่าของ hyperparameter ดังกล่าวจะให้วิธีการทดลองกำหนดค่า hyperparameter จำนวนหนึ่งให้กับอัลกอริทึม จากนั้นเลือกใช้ค่า hyperparameter ที่ทำให้ค่าประสิทธิภาพของโมเดลทำงานสูงที่สุด ทั้งนี้การพิจารณาประสิทธิภาพดังกล่าวจะพิจารณาบนชุดข้อมูลอีกชุดหนึ่งที่เรียกว่า **validation dataset**

จากที่กล่าวข้างต้นจะเห็นว่าในกระบวนการพัฒนาโมเดลการเรียนรู้ของเครื่อง ต้องการชุดข้อมูลทั้งหมดจำนวน 3 ชุด ได้แก่ training, validation และ test dataset โดยที่ training และ validation dataset เป็นชุดข้อมูลที่ใช้ในระยะพัฒนาการเรียนรู้ของโมเดลให้มีประสิทธิภาพสูงสุด ส่วน test dataset เป็นชุดข้อมูลที่ใช้ตรวจสอบประสิทธิภาพด้านความเป็นนัยทั่วไป แต่จะไม่ได้มีส่วนเกี่ยวข้องกับระยะการพัฒนาการเรียนรู้ของโมเดล

### 3.4 Data Partitioning

ในทางปฏิบัติผู้วิเคราะห์มักมีข้อมูลต้นฉบับเพียงชุดเดียวเท่านั้นแต่ละใช้การแบ่งส่วนข้อมูลโดยใช้วิธีการสุ่มตัวอย่าง (random sampling) เพื่อสร้าง training, validation และ test dataset รูปด้านล่างแสดงลักษณะการแบ่งส่วนข้อมูล

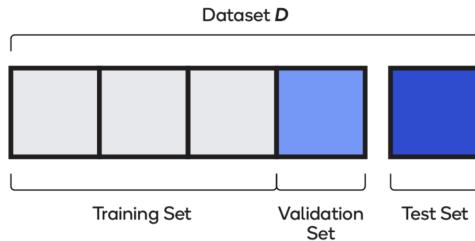


Figure 20: training, validation และ testing dataset

โดยปกติการแบ่งส่วนข้อมูลดังกล่าวไม่ได้มีกฎเกณฑ์ตายตัวว่าควรแบ่งส่วนใดอย่างละเอียดเท่าไหร่ โดยปกติผู้วิเคราะห์มักกำหนดสัดส่วนระหว่าง training + validation dataset กับ test dataset เป็น 80 : 20, 75 : 25, 70 : 30, 60: 40 หรือ 50: 50 ขึ้นอยู่กับว่าชุดข้อมูลต้นฉบับที่มีนั้นมีขนาดใหญ่มากเพียงใด นอกจากนี้การแบ่งส่วนข้อมูลด้วยวิธีการสุ่มตัวอย่างอาจจำแนกเป็น 2 วิธีการ วิธีการแรกคือการสุ่มตัวอย่างแบบง่าย (simple random sampling: SRS) และวิธีการที่สองคือการสุ่มตัวอย่างแบบชั้นภูมิ (stratified random sampling)

#### ชุดข้อมูล mpg

ชุดข้อมูลที่ใช้เป็นตัวอย่างในหัวข้อนี้จะใช้ dataset mpg ที่เป็นชุดข้อมูลตัวอย่างซึ่งถูกติดตั้งมาพร้อมกับการติดตั้งโปรแกรม R อยู่แล้ว ผู้วิเคราะห์สามารถเรียกดูข้อมูลภายในชุดข้อมูลดังกล่าวได้โดยใช้คำสั่งพื้นฐานดังนี้ ๆ เช่น head(), str(), glimpse() หรือ summary() เป็นต้น

```
library(dplyr)
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ year cyl trans     drv   cty   hwy fl class
##   <chr>        <chr>  <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5) f       18    29 p     compact
## 2 audi         a4      1.8  1999     4 manual(m5) f      21    29 p     compact
## 3 audi         a4      2.0  2008     4 manual(m6) f      20    31 p     compact
## 4 audi         a4      2.0  2008     4 auto(av)   f      21    30 p     compact
## 5 audi         a4      2.8  1999     6 auto(l5)   f      16    26 p     compact
## 6 audi         a4      2.8  1999     6 manual(m5) f      18    26 p     compact
```

## การแบ่งข้อมูลด้วยการสุ่มอย่างง่าย

การแบ่งด้วย simple random sampling เป็นการแบ่งโดยสุ่มข้อมูลตามจำนวนที่กำหนดออกมาเป็นชุดๆข้อมูล training dataset หรือ test dataset โดยการสุ่มดังกล่าวมีข้อสมมุติว่าหน่วยข้อมูลทุกหน่วยในชุดข้อมูลต้นฉบับมีโอกาสที่จะถูกสุ่มขึ้นมาเท่ากันทั้งหมด การแบ่งข้อมูลด้วยวิธีการนี้ใน R สามารถทำได้หลายวิธี แต่ในบทความนี้จะใช้วิธีที่อยู่ภายใต้ framework ของ tidyverse การแบ่งข้อมูลด้วยวิธีการดังกล่าวมีสองขั้นตอน ขั้นแรก คือการสร้างกรอบของการแบ่งข้อมูลออกเป็น training และ test data สามารถทำได้ด้วยฟังก์ชัน `initial_split()` จาก package `rsample` ารถกิจกรรมที่สำคัญที่จะต้องระบุในฟังก์ชันได้แก่ `data` และ `prop` ขั้นที่สอง คือการแบ่งข้อมูลตามกรอบในขั้นตอนแรก โดยจะใช้ฟังก์ชัน `training()` เพื่อแบ่งชุด training dataset ออกมานำและใช้ฟังก์ชัน `testing()` เพื่อแบ่งชุดข้อมูล test dataset อกมา

```
# import rsample
library(rsample)

# generate sampling frame
mpg_split1 <- initial_split(data = mpg, prop = 0.75)

mpg_split1

## <Training/Testing/Total>
## <175/59/234>

# create training and test dataset
train_srs <- mpg_split1 %>% training()
test_srs <- mpg_split1 %>% testing()
```

## การแบ่งข้อมูลด้วยการสุ่มแบบชั้นภูมิ

การแบ่งชุดข้อมูลด้วยการสุ่มแบบชั้นภูมิสามารถทำได้ด้วยฟังก์ชัน `initial_split()` เช่นเดียวกัน แต่จะต้องมีการระบุ `strata` เพื่อระบุตัวแปรตามหรือตัวแปรเกณฑ์ที่จะใช้แบ่งชั้นภูมิก่อนการสุ่มตัวอย่าง และ `breaks` ใช้กำหนดจำนวนอันตรภาคชั้นของตัวแปรตามหรือตัวแปรเกณฑ์ที่จะใช้แบ่งชั้นภูมิหากตัวแปรดังกล่าวเป็นตัวแปรเชิงปริมาณ ค่าเริ่มต้นของ `strata` นี้กำหนดให้ `breaks = 4` ตัวอย่างต่อไปนี้แสดงการแบ่งชุดข้อมูล `training` และ `test` ด้วยการสุ่มแบบชั้นภูมิ

```
mpg_split2 <- initial_split(data = mpg,
                                prop = 0.75,
                                strata = "hwy",
                                breaks = 5)

train_str <- mpg_split2 %>% training()
test_str <- mpg_split2 %>% testing()
```

อุปด้านล่างแสดงการเปรียบเทียบการแจกแจงของตัวแปรตามระหว่างชุดข้อมูลต้นฉบับ (full dataset), `training` และ `test` dataset ที่แบ่งด้วยวิธีการสุ่มตัวอย่างแบบง่าย และแบบชั้นภูมิ

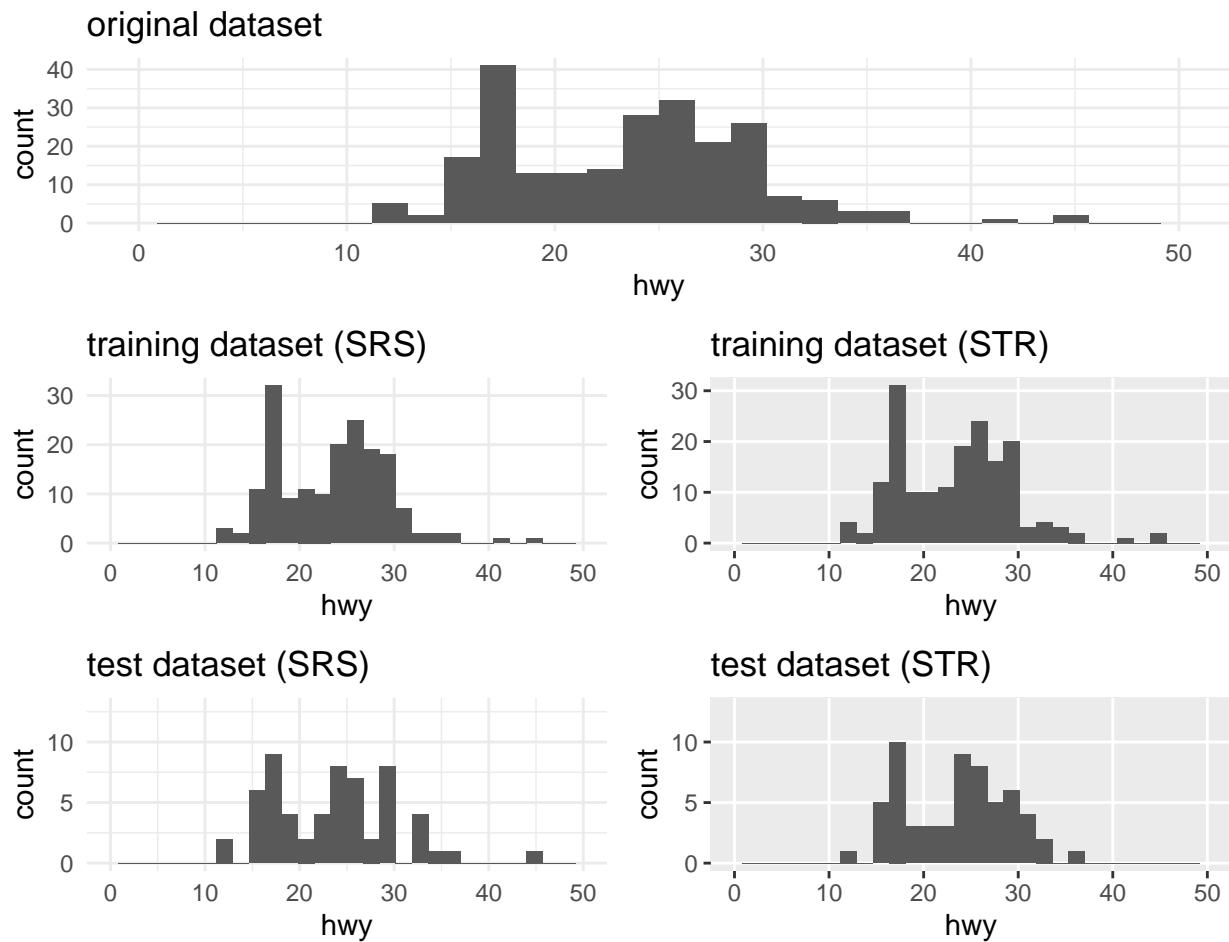


Figure 21: เปรียบเทียบระหว่าง SRS กับ STR

### 3.5 Tidymodels Framework

ปัจจุบันมีเครื่องมือที่ช่วยให้ผู้ใช้เคราะห์สามารถพัฒนา machine model ได้หลายตัว บทเรียนนี้จะกล่าวถึงการใช้โปรแกรม R เพื่อพัฒนา ML model ดังกล่าว ทั้งนี้ต้องทำความเข้าใจก่อนว่า การทำงานบน R แม้จะเป็นปัญหาเดียวกัน ทุดข้อมูลเดียวกัน แต่ผู้ใช้เคราะห์ต่างคนกันก็มีทางที่จะดำเนินการด้วยวิธีการที่แตกต่างกันได้ (ใน Python หรือโปรแกรมอื่น ๆ ก็เช่นเดียวกัน) วิธีการหนึ่งใน R ที่สามารถ modeling ได้ง่ายและมีประสิทธิภาพคือการใช้ **tidymodels framework** ดังรูป 22

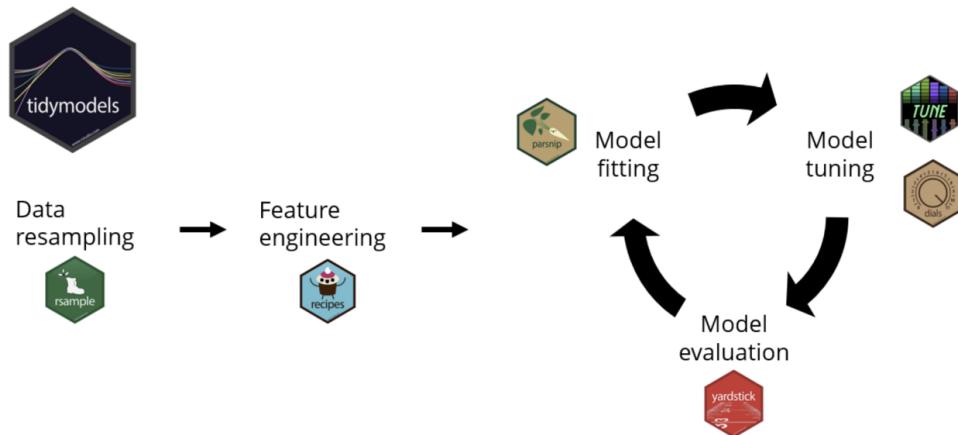


Figure 22: tidymodel framework

- **package-rsample** ใช้ในงาน resampling ข้อมูล เช่นการสร้าง training/validation/test dataset การสร้าง cross-validation dataset หรือการสร้าง bootstrap dataset ซึ่งได้กล่าวการใช้งานเบื้องต้นไปแล้ว
- **package-recipes** ใช้แปลง/แก้ปัญหาที่เกิดขึ้นในข้อมูลของตัวแปรที่ใช้ในการพัฒนาโมเดล ขั้นตอนนี้เรียกว่า feature engineering
- **package-parsnip** ใช้ fit machine learning กับข้อมูล
- **package-TUNE** และ **package-dials** มีฟังก์ชันที่อำนวยความสะดวกในการ fine tune hyperparameter ของโมเดลเพื่อเพิ่มประสิทธิภาพการทำนายของโมเดลให้สูงที่สุด
- **package-yardstick** มีฟังก์ชันของ metric ที่ใช้ประเมินประสิทธิภาพของโมเดลทำนาย

tidymodels ถูกพัฒนาขึ้นโดยได้รับการออกแบบให้สามารถทำขั้นตอนการพัฒนาโมเดลได้ง่าย โดยใช้ไวยกรณ์ของภาษาในลักษณะเดียวกัน และถูกออกแบบโดยเน้นใช้กับ supervised learning เป็นหลัก ผู้ใช้งานไม่จำเป็นต้องติดตั้งทุก package ในขั้นต้นด้วยตนเอง แต่ติดตั้งเพียง package-tidymodels ก็สามารถใช้งานทุก package ภายใต้ framework ดังกล่าวได้แล้ว โดยการพิมพ์คำสั่งต่อไปนี้

```
install.packages("tidymodels") # ดาวน์โหลดและติดตั้ง tidymodels
library(tidymodels) # เรียกใช้ tidymodels
```

## Fitting Linear Regression using parsnip



การ fit machine learning model กับข้อมูลด้วย R ในยุคเริ่มแรกค่อนข้างมีความยากลำบากพอสมควร เพราะ R ไม่ได้มี package ที่เป็น framework รวมสำหรับการ fit ML model ดังกล่าว การที่จะ fit ML model ในงานหนึ่ง ๆ ผู้ใช้เคราะห์อาจจะต้องยุ่งเกี่ยวกับ package จำนวนมาก เช่น

- package rpart สำหรับ fit decision tree
- package glmnet สำหรับ fit regularized regression model
- package knn สำหรับ fit K-NN model

โดย package ที่แตกต่างกันมักมีแนวคิดและไวยกรรมในการเขียนคำสั่งที่แตกต่างกัน ทำให้เป็นอุปสรรคต่อการทำงานโดยเฉพาะการทำข้าวในอนาคต จากปัญหานี้ tidyverse จึงมีการพัฒนา package parsnip ขึ้นเพื่อเป็น interface สำหรับใช้ package ใน R ที่เกี่ยวข้องกับการ fit supervised learning ทั้งนี้ parsnip ได้ถูกออกแบบมาให้การลั่งงานทั้งหมดอยู่ภายใต้ไวยกรรมแบบเดียวกัน ปัจจุบันการ fit ML models ใน R จึงดำเนินการได้ง่ายขึ้นอย่างมาก

ขั้นตอนการ fit ML models ด้วย parsnip มี 2 ขั้นตอน ได้แก่ การระบุโมเดล และการประมวลผล รายละเอียดมีดังนี้

**การระบุโมเดล (model specification)** การระบุโมเดลใน parsnip มีลักษณะของ 3 ส่วนที่จำเป็นได้แก่

- **model type** หรืออัลกอริทึมการเรียนรู้ของเครื่องที่ผู้ใช้เคราะห์จะใช้ในการทำงาน
- **engine** หรือ package ของ R ที่จะใช้สำหรับประมวลผล model type ที่เลือก
- **mode** สำหรับกำหนดว่าปัญหาที่ทำงานด้วยอยู่นี่เป็น regression หรือ classification

รายละเอียดว่าผู้ใช้เคราะห์สามารถกำหนด model type, engine และ mode แบบใดได้บ้างและต้องกำหนดอย่างไร สามารถศึกษาได้จาก <https://www.tidymodels.org/find/parsnip/> รูป 23 ด้านล่างแสดงคันหาสำหรับอัลกอริทึม linear regression จากผลการค้นหาในรูปด้านล่างจะเห็นว่าการ fit linear regression ด้วย parsnip สามารถทำได้ด้วย model type คือ `linear_reg()` เมื่อพิจารณาในคอลัมน์ engine จะเห็นว่าการ fit linear regression มี engine จำนวนมากที่สามารถใช้เพื่อประมาณค่าพารามิเตอร์ของโมเดลได้ engine ดังกล่าวจะมี ๑ แล้วคือ package ต่าง ๆ ของ R ที่ใช้ประมวลผล mode type ที่เลือกไว้ได้ ผู้อ่านจะเห็นว่า model type แบบ `linear_reg` มี engine ที่สามารถใช้ประมวลผลได้จำนวนมาก ซึ่งมีความเหมือนและความแตกต่างกัน เนื้อหาส่วนนี้มีความละเอียดพอสมควรจึงจะกล่าวถึงในบท regression model ต่อไป

ในคุณมีข้างต้นยังมีเครื่องมือให้ค้นหาการกำหนดอัลกอริทึมที่ของฟังก์ชัน model type ในข้างต้น จากรูป 24 จะเห็นรายละเอียดในการกำหนดอัลกอริทึมที่ของฟังก์ชัน `linear_reg()` เมื่อกำหนด engine ในลักษณะต่าง ๆ

## EXPLORE MODELS

Show 5 entries Search: linear regression

TITLE	MODEL TYPE	PACKAGE	MODE	ENGINE
All	All	All	All	All
Linear regression	linear_reg	parsnip	regression	brulee, gee, glm, glmer, glmnet, gls, h2o, keras, lm, lme, lmer, spark, stan, stan_glm

Figure 23: parsnip manual

Show 5 entries Search: linear\_reg

MODEL TYPE	ENGINE	PARSNIP	ORIGINAL
All	All	All	All
linear_reg	glmnet	penalty	lambda
linear_reg	glmnet	mixture	alpha
linear_reg	spark	penalty	reg_param
linear_reg	spark	mixture	elastic_net_param
linear_reg	keras	penalty	penalty

Figure 24: ໃນພັງກ່ຽວຂ້ອງ model type

ความหมายของการกำหนดอาร์กิวเม้นท์แต่ละค่าสามารถศึกษาได้จากคู่มือของฟังก์ชัน `linear_reg()` ซึ่งสามารถกด hyperlink จากคู่มือได้เลย (คู่มือ `linear_reg()`)

เอกสารเพิ่มเติมเกี่ยวกับ package parsnip

- <https://cran.r-project.org/web/packages/parsnip/parsnip.pdf>
- <https://cran.r-project.org/web/packages/parsnip/vignettes/parsnip.html>

สมมุติว่าผู้ใช้เครื่องที่ต้องการพัฒนาโมเดลการเรียนรู้ของเครื่องด้วยอัลกอริทึม linear regression โดยมีตัวแปรตามคือ `hwy` และตัวแปรอิสระเพียง 1 ตัวได้แก่ `cty` ตัวอย่างคำสั่งต่อไปนี้แสดงการกำหนดโมเดลการเรียนรู้ด้วย parsnip ดังกล่าว

```
lm_model <- linear_reg() %>%          # model type
  set_engine("lm") %>%    # model engine
  set_mode("regression") # model mode
```

การประมวลผล เมื่อกำหนดโมเดลการเรียนรู้แล้วขั้นตอนถัดไปคือการนำ model specification ดังกล่าว ไปดำเนินการประมวลผล โดยส่งผ่านไปยังฟังก์ชัน `fit()` ซึ่งมีอาร์กิวเม้นท์สำคัญ 2 ตัวได้แก่ model formula และ training dataset ที่จะใช้สำหรับฝึกหัดโมเดล

การเขียน model formula จะเขียนอยู่ในรูปของ  $y \sim x_1 + x_2 + x_3 + \dots$  โดยที่  $y$  คือตัวแปรตาม ส่วน  $x_1, x_2, x_3, \dots$  คือตัวแปรอิสระภายในชุดข้อมูลฝึกหัด และสัญลักษณ์  $\sim$  หมายความว่า “regress on” ในกรณีที่ต้องการใช้ตัวแปรที่เหลือในชุดข้อมูลทั้งหมดเป็นตัวแปรที่สามารถเขียน model formula ล้าน ๆ ได้ดังนี้ ‘ $y \sim .$ ’ ตัวอย่างต่อไปนี้แสดงการส่งผ่าน model specification `lm_model` ไปยังต้นไปประมวลผล

```
fit_lm <- lm_model %>%
  fit(hwy ~ cty,    # model formula
       data = train_str) # training dataset

fit_lm

## parsnip model object
##
##
## Call:
## stats::lm(formula = hwy ~ cty, data = data)
##
## Coefficients:
## (Intercept)      cty
##             1.139     1.326
```

การเรียกดูค่าประมาณพารามิเตอร์ของ ML model อย่างไรก็ตาม tidymodels มีฟังก์ชัน tidy() ซึ่งช่วยสร้างตารางสรุปผลลัพธ์จากการประมาณค่าพารามิเตอร์หรือการเรียนรู้ของโมเดลที่ใช้หอโยนรูปแบบเดียวกัน ดังนี้

```
tidy(fit_lm)

## # A tibble: 2 x 5
##   term      estimate std.error statistic p.value
##   <chr>     <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 1.14     0.537     2.12 3.51e- 2
## 2 cty         1.33     0.0309    42.9  7.32e-94
```

ภายใต้ framework ของ tidyverse จะใช้ฟังก์ชันใน package parsnip เพื่อ fitting model ทำนายดังกล่าว package ดังกล่าว จุดเด่นของ parsnip คือถูกออกแบบมาเพื่อเป็น interface สำหรับ fit supervised learning model ที่มีรูปแบบการใช้คำสั่งเป็นໄวยกรณ์แบบเดียว

**Prediction** ผู้ใช้เคราะห์สามารถนำโมเดลที่ผ่านการ train เรียบร้อยแล้วไปใช้หาค่าทำนาย โดยส่งผ่านโมเดลที่ train แล้ว (ในที่นี่คือ fit\_lm) ไปยังฟังก์ชัน predict() ที่มีอาร์กิวเม้นท์สำคัญคือ new\_data ตัวอย่างด้านล่างแสดงนำ fit\_lm ไปทำนายตัวแปร hwy ในชุดข้อมูลทดสอบ ผลลัพธ์ที่ได้จากการทำนายจะเป็นตารางแบบ tibble ที่แต่ละ row คือค่าทำนายของหน่วยข้อมูลใน row เดียวกันกับใน test\_str ดังนี้

```
hwy_pred <- fit_lm %>%
  predict(new_data = test_str)

hwy_pred
```

```
## # A tibble: 60 x 1
##       .pred
##   <dbl>
## 1 29.0
## 2 25.0
## 3 21.0
## 4 22.4
## 5 19.7
## 6 26.3
## 7 25.0
## 8 23.7
## 9 23.7
## 10 23.7
## # ... with 50 more rows
```

เมื่อได้ค่าที่นายในชุดข้อมูลทดสอบมาแล้ว ขั้นตอนถัดไปคือการประเมินประสิทธิภาพของโมเดลที่นายทำ โดยทั่วไปผู้ใช้เคราะห์มักจะรวมค่าที่นายได้ (ในที่นี้คือ hwy\_pred) ไปไว้อยู่ภายใต้ชุดข้อมูลทดสอบ การดำเนินการนี้สามารถทำได้หลายวิธีการ ขึ้นอยู่กับว่าตนจะดำเนินการแบบนี้ ในตัวอย่างนี้จะใช้ฟังก์ชัน bind\_cols()

```
test_results <- test_str %>%
  select(hwy, cty) %>%
  bind_cols(hwy_pred)

test_results

## # A tibble: 60 x 3
##       hwy     cty .pred
##   <int> <int> <dbl>
## 1     29     21  29.0
## 2     26     18  25.0
## 3     25     15  21.0
## 4     26     16  22.4
## 5     19     14  19.7
## 6     27     19  26.3
## 7     26     18  25.0
## 8     26     17  23.7
## 9     24     17  23.7
## 10    24     17  23.7
## # ... with 50 more rows
```

### Evaluating models using yardstick

การคำนวณค่าประสิทธิภาพของโมเดลที่นายแบบ regression ภายใต้ framework tidyverse สามารถทำได้ง่าย ๆ โดยใช้ฟังก์ชันจาก package yardstick ได้แก่ rmse() และ rsq() ตัวอย่างด่อไปนี้แสดงการเปลี่ยนคำสั่งเพื่อคำนวณ metric ทั้งสอง

```
test_results %>%
  rmse(truth = hwy, estimate = .pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>        <dbl>
## 1 rmse    standard     1.64
```

```

test_results %>%
  rsq(truth = hwy, estimate = .pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>    <chr>        <dbl>
## 1 rsq      standard     0.913

```

นอกจาก metric ที่เป็นค่าสถิติแล้วยังมี metric ที่เป็น visualization ด้วย เช่นในกรณีของ regression model สามารถใช้ R squared plots เพื่อประเมินความสอดคล้องกันระหว่างค่าจริงของตัวแปรตามกับค่าที่นายได้ การสร้าง R squared plot ใน R สามารถทำได้หลายวิธี ทั้งการใช้ฟังก์ชัน plot() ของ package graphic เมื่อกับตัวอย่างในบทที่ 2 นอกจากนี้ยังสามารถใช้ package ggplot2 เพื่อสร้างแผนภาพดังกล่าวได้เหมือนกัน

```

# create R squared plot using graphic package
plot(x = test_results$.pred,
      y = test_results$hwy,
      pch = 16,
      xlab = "predicted value",
      ylab = "actual value")
abline(a=1,b=1, lty=3, col="steelblue")

```

ในกรณีที่ต้องการใช้ ggplot2 สามารถเขียนคำสั่งได้ดังนี้

```

# create R squared plot using ggplot2 package
library(ggplot2)
test_results %>% ggplot() + # create 2D plane
  geom_point(aes(x = .pred, # create scatter plot
                 y = hwy)) +
  geom_abline(intercept=1, slope=1, linetype=3, col="steelblue") +
  coord_obs_pred() +
  theme(text=element_text(size = 10))

```

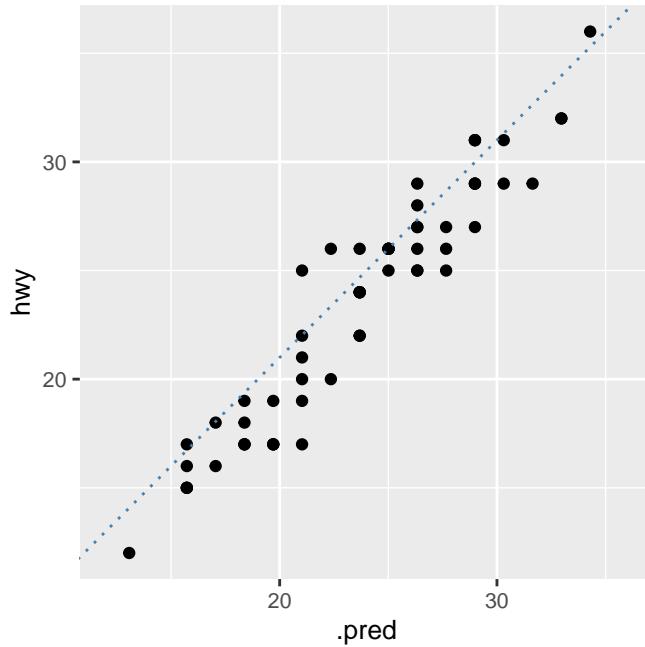


Figure 25: R squared plot via ggplot2

คำถาน เรายได้เรียน linear regression มาพอสมควรแล้ว โดยหากจำได้จะทราบว่า linear regression เป็นโมเดลทางสถิติ ที่มีข้อตกลงเบื้องต้นที่ค่อนข้างเข้มงวด ได้แก่ independence, homoscedasticity, normality, no multicollinearity, no outlier, ... คำถานคือในการพัฒนา ML model ดังกล่าวจำเป็นมั้ยที่จะต้องตรวจสอบข้อตกลงเบื้องต้นดังกล่าว เพราะอะไร?

### 3.6 กิจกรรมพัฒนา regression model ด้วย tidymodel framework

1. นำเข้าข้อมูลจากไฟล์ `TeacherSalaryData.csv`
2. สำรวจข้อมูลจากชุดข้อมูลดังกล่าว แล้วตอบคำถาม
  - ชุดข้อมูลนี้มีตัวแปรทั้งหมดกี่ตัว
  - มีหน่วยข้อมูลทั้งหมดกี่หน่วย
  - หาค่าสถิติพื้นฐานของตัวแปรเชิงปริมาณในชุดข้อมูล
  - อาจารย์มหาวิทยาลัยส่วนใหญ่มีตำแหน่งวิชาการอะไร
3. แบ่งส่วนข้อมูลที่นำเข้าออกเป็นสองส่วน ได้แก่ training และ test dataset โดยกำหนดให้สัดส่วนระหว่างชุดข้อมูล ทั้งหมดเป็น 80 : 20
4. กำหนดให้ตัวแปรตามคือ `salary` (เงินเดือนของอาจารย์มหาวิทยาลัย) ลงพัฒนา supervised learning model 2 โมเดล โดยตัวแรกให้ใช้ linear regression model ที่ใช้ `lm` เป็น engine และตัวที่สองให้ใช้ decision tree ที่ใช้ `rpart` เป็น engine ทั้งนี้ให้ใช้ตัวแปรอิสระทุกด้วยในชุดข้อมูลเป็นตัวแปรทำนาย
5. เปรียบเทียบประสิทธิภาพในการทำนายของโมเดลทั้งสอง ผลที่ได้เป็นอย่างไร

## สรุป

บทเรียนนี้ผู้อ่านได้เห็นภาพของกระบวนการพัฒนา regression models ซึ่งเป็น supervised learning ประเภทหนึ่งโดย เป็นการดำเนินงานภายใต้ tidymodels framework เกือบทั้งหมด โดยยังขาดในส่วนของการจัดการข้อมูลหรือที่เรียกว่า feature engineering และส่วนการปรับแต่งค่า hyperparameters ของโมเดลทำนาย ซึ่งจะกล่าวรายละเอียดทั้งหมดในบทเรียน ต่อไป