Week1: Introduction to ML

ผศ.ดร.สิวะโชติ ศรีสุทธิยากร

Contents

1. What's algorithms?

อัลกอริทึมคือกระบวนการในการดำเนินงานที่มีขั้นตอนอย่างชัดเจน โดยมีวัตถุประสงค์เพื่อทำงาน/แก้ปัญหาที่กำหนดให้สำเร็จ การใช้อัลกอริทึมในการทำงานนั้นไม่ได้จำกัดเฉพาะงานทางด้านคอมพิวเตอร์ หรือสถิติและวิทยาการข้อมูลเท่านั้น แต่ในชีวิตประจำวันเราก็มีการใช้อัลกอริทึมเพื่อดำเนินงานต่าง ๆ อยู่เป็นประจำ เช่น

- การเดินทางจากบ้านไปยังร้านขายของสะดวกซื้อ งานดังกล่าวสามารถเขียนแยกแยะออกมาเป็นขั้นตอนการเดินทางโดยอาจเริ่มตั้งแต่การออกประตูบ้าน เลี้ยวขวา เดินตรงไป เมื่อพบสามแยกให้เลี้ยวขวาอีกครั้งจะพบร้านสะดวกซื้อ
- การทอดไข่เจียวที่อาจเริ่มจากการตั้งไฟ ใส่น้ำมัน ตอกไข่ ตีไข่ ใส่เครื่องปรุง ทอดไข่ และนำไข่เจียวที่ได้เสิร์ฟ

ปัจจุบันโลกได้ก้าวเข้าสู่ยุคที่ให้เครื่องจักร เช่น เครื่องคอมพิวเตอร์ทำงานบางอย่างแทนมนุษย์ได้ ซึ่งเบื้องหลังการดำเนินการของเครื่องจักรต่าง ๆ จำเป็นต้องมีอัลกอริทึมที่ใช้สำหรับควบคุมการทำงาน การพัฒนาอัลกอริทึมดังกล่าวอาจทำได้สองวิธีการ วิธีการแรกเรียกว่า rule-based algorithm ที่ผู้พัฒนาเป็นผู้กำหนดขั้นตอนวิธีการทำงานและประมวลผลทั้งหมด ส่วนวิธีการที่สองเรียกว่า machine learning-based ที่เป็นวิธีการสมัยใหม่และถูกนำมาใช้เป็นวิธีการหลักในปัจจุบัน การพัฒนาอัลกอริทึมด้วยวิธีการนี้จะนำข้อมูลที่เกี่ยวข้องมาเป็นต้นแบบเพื่อสอนให้คอมพิวเตอร์ได้เรียนรู้ผ่านอัลกอริทึมการเรียนรู้ต่าง ๆ เรียกว่า learning ซึ่งเมื่อ learning ได้การเรียนรู้จากข้อมูลที่มากเพียงพอจะได้ผลลัพธ์เป็นอัลกอริทึมหรือโมเดลที่สามารถใช้ดำเนินการตัดสินใจได้ด้วยตนเอง อัลกอริทึมประเภทนี้จะมีความยืดหยุ่นและมีประสิทธิภาพมากกว่าอัลกอริทึมแบบ rule-based

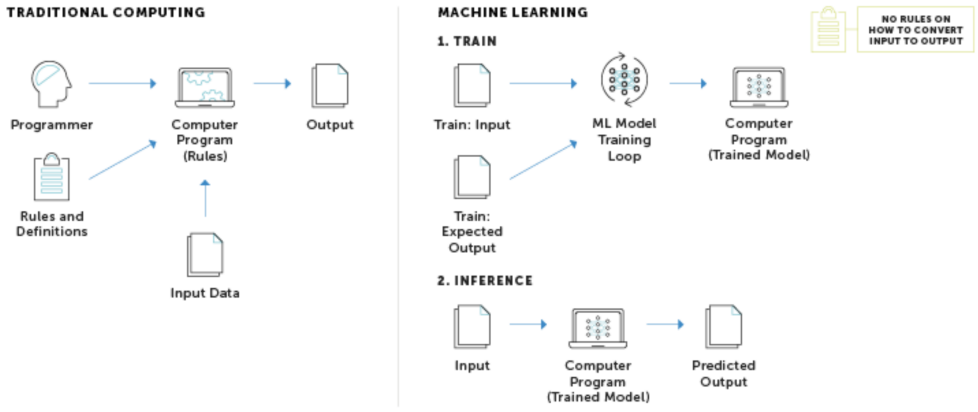


Figure 1: rule-based vs ML-based (https://www.epam.com/insights/blogs/making-ai-more-human-black-box-models-lead-to-better-decision-making)

เพื่อให้เห็นภาพชัดเจนมากขึ้นลองพิจารณาปัญหาการอ่านตัวเลขจากลายมือด้านล่าง ปัญหานี้อ้างอิงจาก (Geron, 2019)

Figure 2: Handwritting Digit Recognition

จากรูปจะเห็นว่าการเขียนกฎเกณฑ์แบบ rule-based เพื่อจำแนกตัวเลขในรูปข้างต้นทำได้ยากมากและเป็นไปแทบไม่ได้เลยที่จะมีอัลกอริทึมแบบ rule-based ที่มีประสิทธิภาพในการจำแนกตัวเลขดังกล่าว ในขณะที่อัลกอริทึมแบบ ML-based สามารถนำภาพของลายมือดังกล่าวไปให้อัลกอริทึมการเรียนรู้ของเครื่องได้เรียนรู้ และสร้างโมเดลจำแนก (classifier) เพื่อจำแนกตัวเลขจากลายมือดังกล่าวได้อย่างมีประสิทธิภาพ ดังตัวอย่างต่อไปนี้

- https://www.kaggle.com/code/pranjalrathore/digit-recognizer-minst
- https://www.kaggle.com/code/alphaghostusmc/mnist-cnnv2
- https://www.kaggle.com/code/kobakhit/digital-recognizer-in-r
- https://www.kaggle.com/code/ivoruaro/mnist-xgboost-r

2. What's ML?

การเรียนรู้ของเครื่อง (ML) เป็นศาสตร์ย่อยแขนงหนึ่งภายใต้ศาสตร์ทางด้านสถิติและวิทยาการข้อมูล ซึ่งเกี่ยวข้องกับการใช้**อัลกอริทึม (algorithms)** ในการเรียนรู้/ค้นหาความรู้จากข้อมูล แล้วนำความรู้ที่ได้มาใช้งานตั้งแต่**การบรรยายสภาพของข้อมูล (descriptive) การวินิจฉัย (diagnostic) เพื่อหาสาเหตุหรือปัจจัยที่ก่อให้เกิดผลลัพธ์ที่สนใจ การทำนาย (predictive) เพื่อสร้างโมเดลที่เรียนรู้ความสัมพันธ์ในข้อมูลเพื่อทำนายผลลัพธ์ของตัวแปรที่สนใจ ผลลัพธ์ที่ได้จากการทำนายนี้สามารถนำมาโมเดลเพื่อช่วยวางแผน/ตัดสินใจ (prescriptive)** ดำเนินการเพื่อนำไปสู่ผลลัพธ์ที่คาดหวัง

3. AI vs ML vs DL

ปัจจุบันมีการใช้คำว่า AI, ML และ DL แทนกันไปมาจนบางครั้งเหมือนว่าจะเป็นคำเดียวกัน ในความเป็นจริงทั้งสามคำดังกล่าวมิได้เป็นสิ่งเดียวกันเลยทีเดียว แต่มีทั้งส่วนที่เหมือนและแตกต่างกัน รายละเอียดมีดังนี้

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

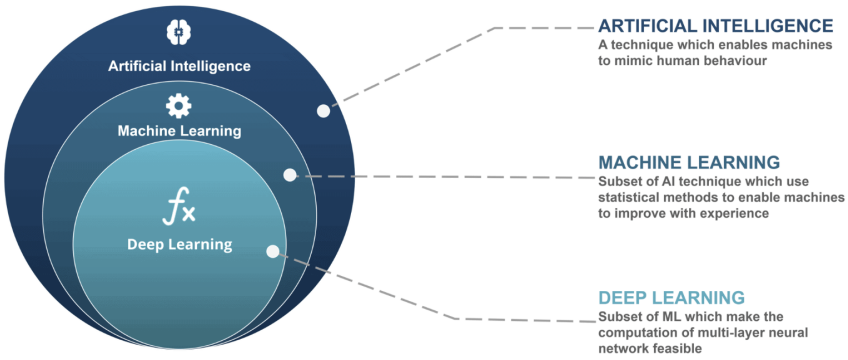


Figure 3: AI, ML และ DL (https://k21academy.com/datascience/deep-learning/dl-vs-ml/)

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

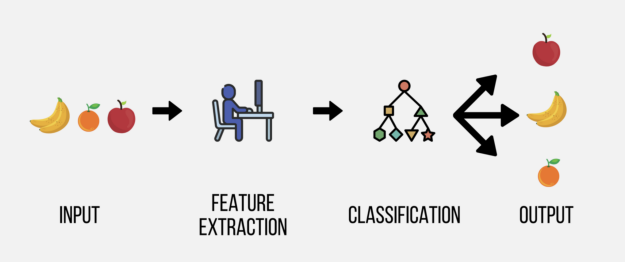


Figure 4: ML (https://www.advancinganalytics.co.uk/blog/2021/12/15/understanding-the-difference-between-ai-ml-and-dl-using-an-incredibly-simple-example)

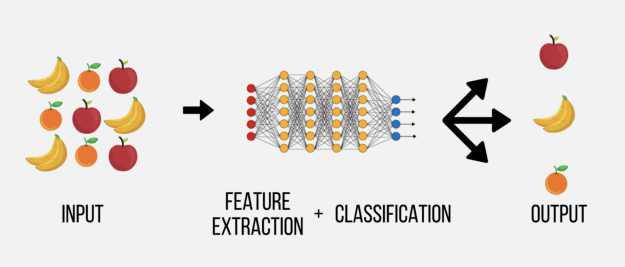


Figure 5: DL (https://www.advancinganalytics.co.uk/blog/2021/12/15/understanding-the-difference-between-ai-ml-and-dl-using-an-incredibly-simple-example)

4. Types of ML

เทคนิคการเรียนรู้ของเครื่องอาจจำแนกได้เป็น 3 ประเภท ตามวัตถุประสงค์หรือความสามารถของอัลกอริทึมการเรียนรู้ ได้แก่

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

4.1 Supervised Learning

ผู้วิเคราะห์จะใช้ supervised learning เมื่อมีวัตถุประสงค์ที่ต้องการสร้างโมเดลทำนาย/จำแนกค่าสังเกตของตัวแปรตามด้วยข้อมูลค่าสังเกตของตัวแปรอิสระ โดย supervised learning เป็นกลุ่มของอัลกอริทึมที่จะเรียนรู้รูปแบบความสัมพันธ์ระหว่างตัวแปรอิสระกับตัวแปรตาม และใช้รูปแบบความสัมพันธ์ที่เรียนรู้จากข้อมูลในอดีตนี้ในการทำนายข้อมูลที่ไม่ทราบค่าที่จะเกิดขึ้นในอนาคต เช่น

- การทำนายสถานะการเป็นหนี้ของลูกค้า (ลูกหนี้ชั้นดี ลูกหนี้ปกติ ลูกหนี้เสีย) โดยอิงกับข้อมูลส่วนตัว ข้อมูลที่เกี่ยวข้องกับเครดิตทางการเงิน และข้อมูลพฤติกรรมการดำเนินชีวิต
- ผู้พัฒนาคอสเรียนออนไลน์ใช้ supervised learning เพื่อทำนายผลการเรียนของนักเรียน หรือแนวโน้มการ drop out ของนักเรียนในคอสเรียน โดยอิงจากพฤติกรรมการเรียนที่แสดงในระบบการเรียนรู้ออนไลน์

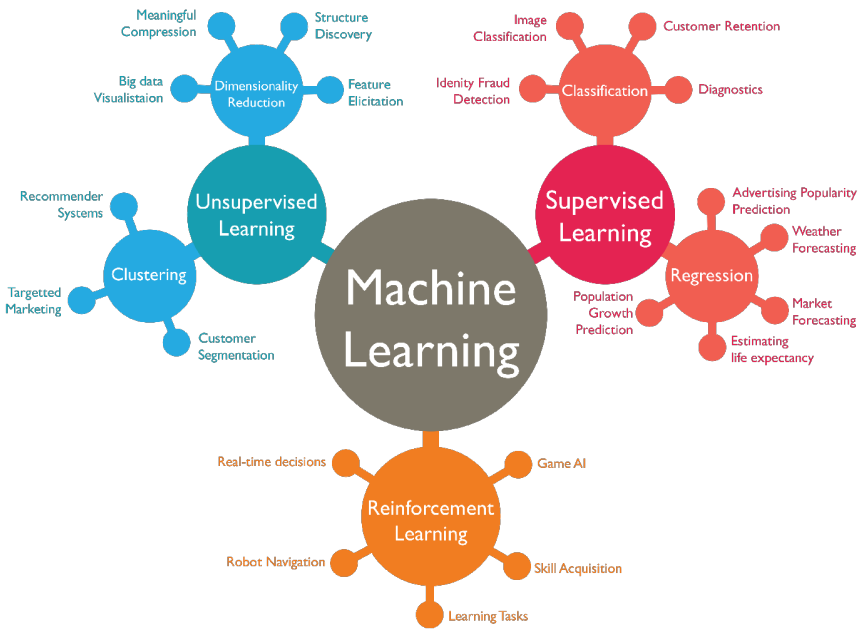


Figure 6: ประเภทของ ML

- การพัฒนาระบบวินิจฉัยความยึดมั่นผูกพันของนักเรียนด้วยการรู้จำใบหน้าโดยใช้การเรียนรู้เชิงลึก

Figure 7: ลักษณะของ ML ประเภท supervised learning (https://3.bp.blogspot.com/-occLtedKtRw/W8RVv5QyIlI/AAAAAAAAAEBg/fdvwBPGxdfQ1izWa_l95-SW4kgYSMgAsgCLcBGAs/s1600)

การที่จะใช้ supervised learning ได้นั้นผู้วิเคราะห์ยังจำเป็นต้องมีชุดข้อมูลต้นแบบที่ภายในชุดข้อมูลประกอบด้วยข้อมูลของตัวแปรตามหรือผลลัพธ์ที่ต้องการทำนาย และตัวแปรอิสระหรือข้อมูลที่จะใช้เป็นตัวทำนายผลลัพธ์ที่ต้องการดังกล่าว ในเชิงเทคนิคจะเรียกชุดข้อมูลต้นแบบดังกล่าวว่า **ชุดข้อมูลฝึกหัด (training dataset)** นอกจากนี้ supervised learning ยังจำแนกเป็นประเภทย่อยได้อีก 2 ประเภทตามลักษณะของตัวแปรตาม ได้แก่ regression และ classification

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

4.2 Unsupervised Learning

ภาษาไทยอาจใช้คำว่าการเรียนรู้แบบไม่มีการชี้นำ การเรียนรู้ประเภทนี้มีความแตกต่างจาก supervised learning กล่าวคือชุดข้อมูลฝึกหัดไม่จำเป็นต้องมีค่าสังเกตของตัวแปรตาม และวัตถุประสงค์ของการใช้ unsupervised learning คือการสร้างหรือสกัดสารสนเทศออกมาจากข้อมูล ซึ่งอาจจำแนกได้เป็น การจัดกลุ่ม (clustering) และการหาความสัมพันธ์ (association)

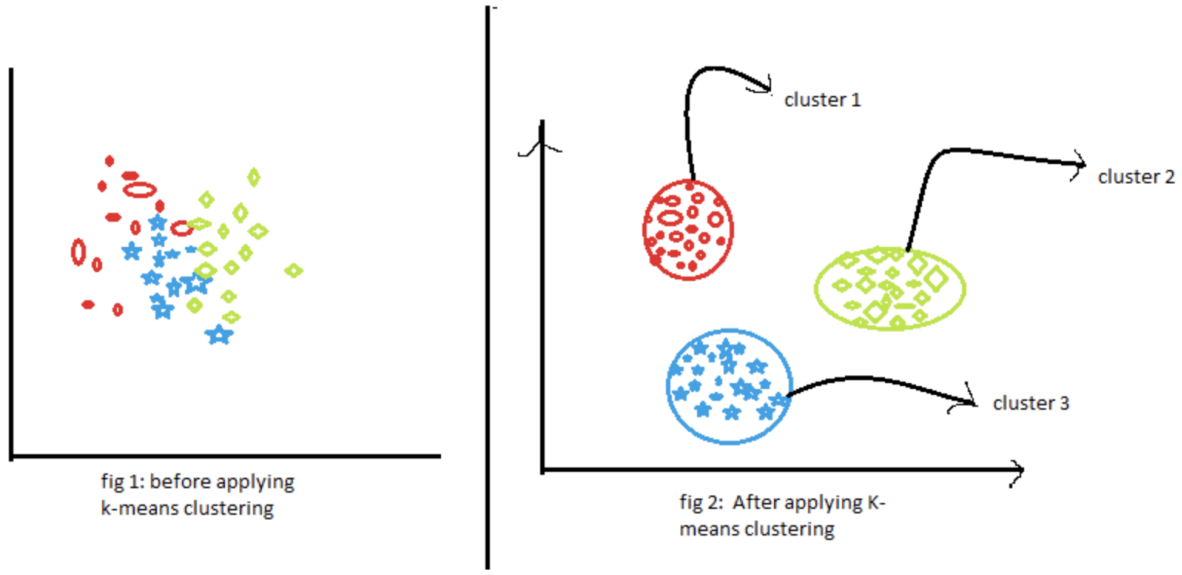


Figure 8: ลักษณะของการ clustering

4.3 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

5. ตัวอย่างการประยุกต์ใช้ ML ทางการศึกษา (การบ้าน)

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

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

Table 1: ข้อมูลคะแนนสอบและพฤติกรรมการเรียนของนักเรียน

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

6. Introduction to Supervised Learning 1 (regression models)

หัวข้อนี้ประกอบด้วยกิจกรรมและเนื้อหาที่สำคัญเกี่ยวกับการพัฒนา supervised learning รายละเอียดมีดังนี้

6.1 กิจกรรม 1 : Rule-based algorithm

ข้อมูลในตาราง 1 ประกอบด้วยคะแนนสอบ (score) กับคะแนนพฤติกรรมการเรียนของนักเรียน (behav) ลองดำเนินการดังนี้

1. พิจารณาความสัมพันธ์เบื้องต้นระหว่าง `score` กับ `behav`
2. ลองสร้างโมเดลทำนายคะแนนสอบ (หาสมการเส้นตรง)โดย (1) ลองใช้วิธีการลากเส้นด้วยมือ (2) ลองคำนวณหาสมการเส้นตรงด้วยวิธีการทางคณิตศาสตร์
3. สมการเส้นตรงที่ได้จากวิธีการทั้งสองเป็นอย่างไร และมีประสิทธิภาพในการทำนายเป็นอย่างไร

6.2 กิจกรรม 2 : ติดตั้ง R

กิจกรรมนี้มีวัตถุประสงค์คือให้ผู้เรียนได้ทำความรู้จักกับเครื่องมือของนักวิทยาการข้อมูลที่สามารถใช้ประมวลผลการเรียนรู้ของเครื่อง ขอให้ผู้เรียนติดตั้ง R และ Rstudio โดยดาวน์โหลด R ที่นี่ และดาวน์โหลด Rstudio ที่นี่

6.3 กิจกรรม 3 : ML-based using Linear regression model

จากตัวอย่างข้อมูลในกิจกรรม 1 เราจะสร้างโมเดลทำนายใหม่ด้วยอัลกอริทึมการเรียนรู้ของเครื่อง ทั้งนี้จะให้ใช้อัลกอริทึมที่ทุกคนรู้จักกันดีตั้งแต่ในรายวิชาสถิติพื้นฐาน คือ linear regression model

โมเดลการวิเคราะห์การถดถอยเชิงเส้น (linear regression) เป็นโมเดลเชิงสถิติ (statistical model) ที่ใช้สำหรับทำนายแนวโน้มค่าสังเกตของตัวแปรตามที่ไม่ทราบค่าโดยอิงกับค่าสังเกตของตัวแปรอิสระที่ทราบค่า การเรียนรู้ของ linear regression จะพยายามสร้างสมการเส้นตรงที่ดีที่สุด (best linear equation) ที่สามารถใช้เป็นตัวแทนความสัมพันธ์ตามธรรมชาติระหว่างตัวแปรตามกับตัวแปรอิสระที่พบในชุดข้อมูล ในเชิงเทคนิคการหาสมการเส้นตรงดังกล่าวจะเป็นการแก้สมการหรือเฟ้นหาค่าของพารามิเตอร์ภายในสมการเส้นตรง ได้แก่ พารามิเตอร์จุดตัดแกน y และพารามิเตอร์ความชัน ที่ทำให้สมการเส้นตรงมีความคลาดเคลื่อนในการทำนายต่ำที่สุด

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
## -0.09524  0.27619 -0.35238  0.01905  0.39048
##        6
## -0.23810
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept)  1.09524    0.23425   4.675
## x            0.62857    0.07737   8.124
##             Pr(>|t|)
## (Intercept)  0.00948 **
## x            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
```

การใช้งานโมเดล `fit_linear` จะเน้นไปในการทำนายมากกว่าการอธิบายความสัมพันธ์แบบที่ทำให้ data analysis ดังนั้นก่อนที่จะนำโมเดลทำนายไปใช้ผู้วิเคราะห์จำเป็นต้องตรวจสอบให้มั่นใจว่าโมเดลที่พัฒนาขึ้นมีประสิทธิภาพในการทำนาย ทั้งนี้มีเกณฑ์การพิจารณาที่เรียกว่า evaluation metric ได้หลายเกณฑ์ที่สามารถใช้ประเมินประสิทธิภาพดังกล่าวได้ในที่นี้จะกล่าวเกณฑ์ที่มักใช้สำหรับประเมิน regression model ได้แก่ RMSE (root mean squared error) และค่า R-squared

Root Mean Squared Error (RMSE)

ในทางทฤษฎีค่า RMSE มีความหมายเป็นค่าคลาดเคลื่อนในการทำนายโดยเฉลี่ยของโมเดล สามารถคำนวณได้จากสูตร

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

Coeffcient of Determination (R squared)

ส่วน R squared มีความหมายเป็นสัดส่วนของความผันแปรที่ร่วมกันระหว่างค่าจริงของตัวแปรตามกับค่าทำนายของตัวแปรตามที่ได้จากโมเดลทำนาย การคำนวณค่า R squared สามารถทำได้ง่าย ๆ ด้วยการหาค่ากำลังสองของสัมประสิทธิ์สหสัมพันธ์ระหว่างค่าจริงกับค่าทำนายของตัวแปรตามดังกล่าว

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

จะเห็นว่า evalulation metric ทั้งสองล้วนเป็นการเปรียบเทียบความแตกต่างหรือความสอดคล้องระหว่างค่าจริงของตัวแปรตามกับค่าทำนายที่ได้จากโมเดลทำนาย การคำนวณค่าของ metric ดังกล่าวสามารถเขียนคำสั่งใน R ได้ดังนี้

```
# calculate prediction values
pred<-predict(fit_linear)
pred # predicted value
```

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

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

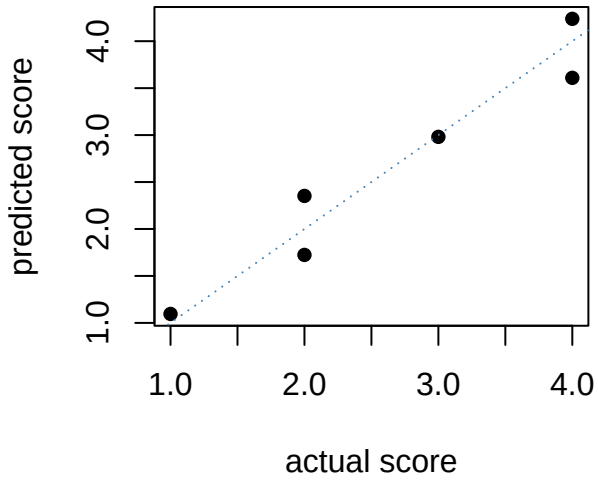
```
## [1] 0.264275
```

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

```
## [1] 0.9428571
```

R squared plot

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



6.4 Bias and Variance in ML models

- พิจารณารูป 9 แสดงการ fit โมเดลทำนาย 3 แบบกับชุดข้อมูลฝึกหัดชุดหนึ่ง จะเห็นว่าแต่ละโมเดลมีความสามารถในการเรียนรู้ความสัมพันธ์ที่เกิดขึ้นในชุดข้อมูลแตกต่างกัน
- ความแตกต่างระหว่างค่าจริงของตัวแปรตามในชุดข้อมูลฝึกหัดกับค่าทำนายที่ได้จากโมเดล เรียกว่า **ความลำเอียง (bias)**
- **จากรูป 9 ผู้อ่านคิดว่าโมเดลใดที่มีประสิทธิภาพในการทำนายสูงที่สุดเพราะเหตุใด ?**
- พิจารณารูป 10 ผู้วิเคราะห์ได้นำโมเดลทำนายทั้ง 3 แบบ ที่พัฒนาจากชุดข้อมูลฝึกหัดมาใช้ทำนายข้อมูลใหม่ที่โมเดลทั้ง 3 ไม่เคยได้เรียนรู้มาก่อน **ผลการทำนายที่ได้เป็นอย่างไร ?**
- ความแตกต่างระหว่างค่าจริงของตัวแปรตามในชุดข้อมูลใหม่ (หรือชุดข้อมูลที่ไม่ได้ใช้ในขั้นการพัฒนาโมเดล) กับค่าทำนายของโมเดล เรียกว่า **ความแปรปรวน (variance)**

จากตัวอย่างข้างต้นทำให้ได้ข้อสรุปว่า การวัดประสิทธิภาพของโมเดลทำนายอย่างน้อยผู้วิเคราะห์จะต้องพิจารณาความคลาดเคลื่อนของโมเดล 2 ด้าน ได้แก่ ความลำเอียง (biased) และความแปรปรวน (variance) โดยที่ (1) ความลำเอียงเป็นตัวชี้วัดที่บอกผู้วิเคราะห์ว่าสามารถสอนให้โมเดลเรียนรู้ความสัมพันธ์ระหว่างตัวแปรหรือสารสนเทศภายในชุดข้อมูลฝึกหัดได้ดีมากน้อยแค่ไหน ความลำเอียงสามารถประมาณได้จากความคลาดเคลื่อนระหว่างค่าจริงของตัวแปรตามกับค่าทำนายภายใต้ชุดข้อมูลฝึกหัด และ (2) ความแปรปรวน (variance) ใช้เป็นตัวชี้วัดความเป็นนัยทั่วไป หรือผู้วิเคราะห์สามารถนำโมเดลทำนายที่พัฒนาขึ้นไปใช้กับข้อมูลที่โมเดลไม่รู้จักในประชากรได้ดีมากน้อยแค่ไหน

ในเชิงอุดมคติ ผู้วิเคราะห์ต้องการให้ทั้งความลำเอียง และความแปรปรวนมีค่าต่ำที่สุดเท่าที่จะสามารถต่ำได้ แต่ในความเป็นจริงความคลาดเคลื่อนทั้งสองไม่ควบคุมให้ต่ำที่สุดพร้อมกันได้ **(เพราะอะไร?)** รูป 11 ด้านล่างแสดงความสัมพันธ์ระหว่าง

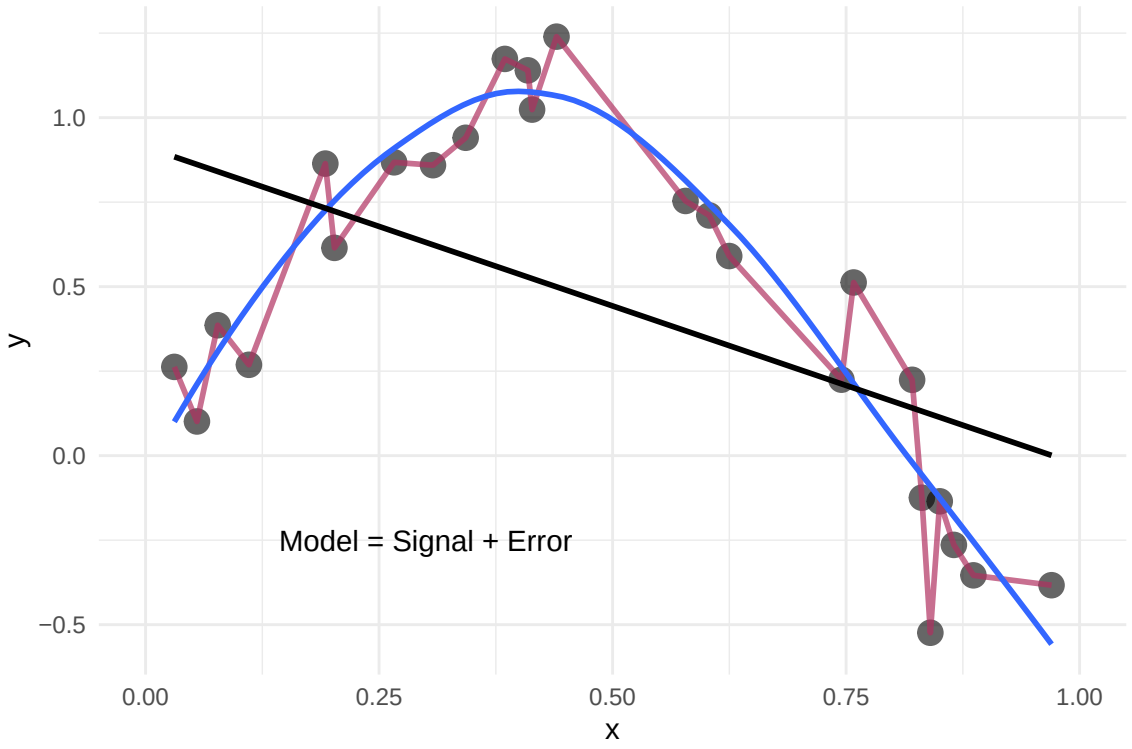


Figure 9: regression model on training dataset

ความลำเอียง และความแปรปรวน ซึ่งจะเห็นว่ามีการแปรผกผันซึ่งกันและกัน โมเดลที่มีความลำเอียงสูงมีแนวโน้มที่จะมีความแปรปรวนต่ำ และในทางกลับกันโมเดลที่มีความลำเอียงต่ำจะมีแนวโน้มที่มีความแปรปรวนสูง **ดังนั้นวัตถุประสงค์ของการพัฒนาโมเดลจึงเป็นการหาจุดที่ดีที่สุดที่ทำให้ความคลาดเคลื่อนทั้งสองอยู่ในจุดที่ต่ำที่สุดเท่าที่จะเป็นไปได้**

6.5 Underfitting, Overfitting และ Good fit models

หากจำแนกโมเดลทำนายที่ถูกพัฒนาขึ้นตามประสิทธิภาพการทำนายของโมเดล อาจจำแนกได้เป็น 3 ประเภท ดังในรูป 12 ได้แก่

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

6.6 Training, validation, and Test Dataset

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

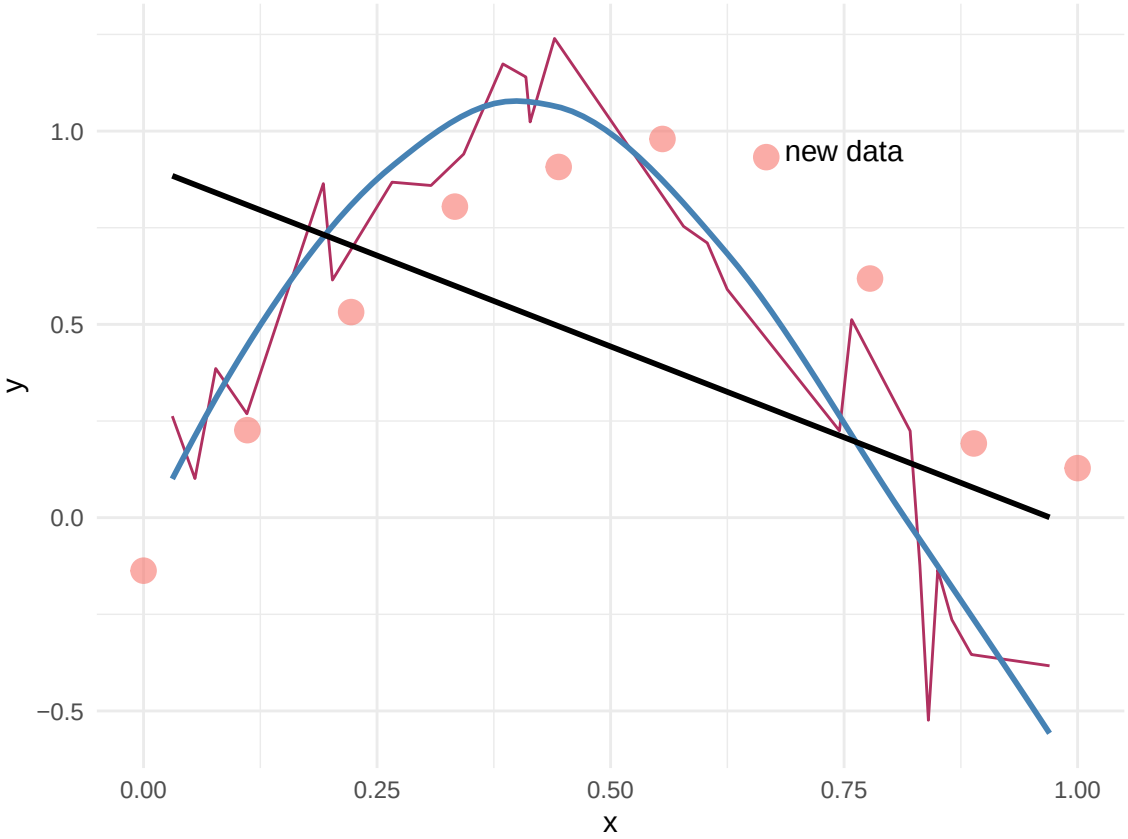


Figure 10: regression model on new dataset

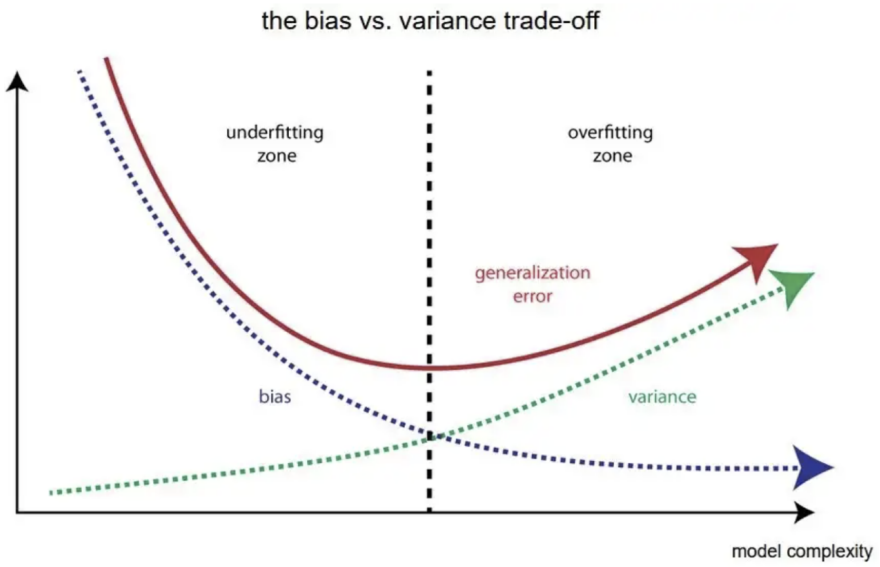


Figure 11: bias and variance trace-off

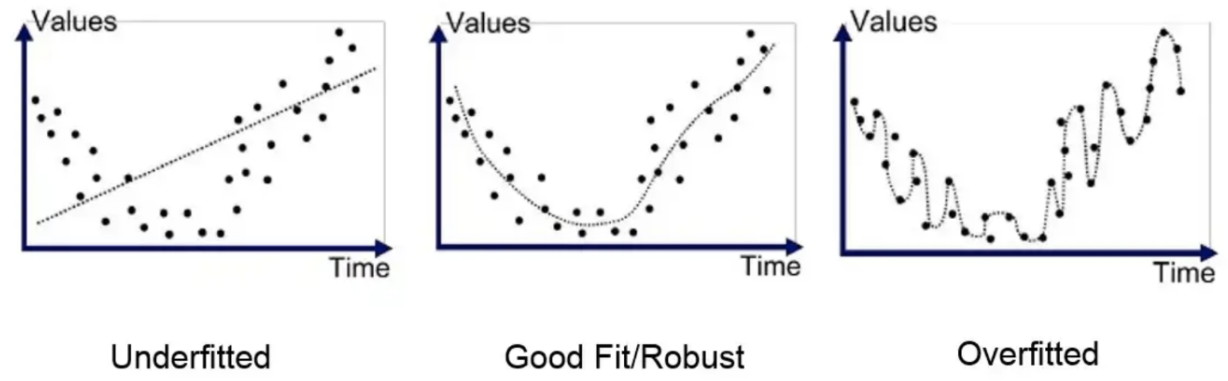


Figure 12: underfitting, good fit, and overfitting model

ภายในอัลกอริทึม supervised learning จะมีส่วนประกอบหลัก ๆ ได้แก่ อัลกอริทึม พารามิเตอร์ และ ไฮเปอร์พารามิเตอร์

- **อัลกอริทึม** เป็นส่วนของวิธีการเรียนรู้ของสำหรับแต่ละการเรียนรู้ของเครื่องที่ใช้ในการเรียนรู้หรือสกัดสารสนเทศจากข้อมูลในชุดข้อมูลฝึกหัด
- **พารามิเตอร์ (parameters)** ส่วนที่ทำให้การเรียนรู้ของเครื่อง fit กับข้อมูล กล่าวง่าย ๆ คือค่าของพารามิเตอร์ที่เปลี่ยนแปลงไป จะทำให้รูปแบบการเรียนรู้มีการเปลี่ยนไป ค่าพารามิเตอร์นี้สามารถประมาณได้จากข้อมูลด้วยวิธีการทางสถิติ/คณิตศาสตร์ ตัวอย่างของพารามิเตอร์เช่น ใน linear regression model มีพารามิเตอร์คือ สัมประสิทธิ์จุดตัดแกน และสัมประสิทธิ์ความชัน เป็นต้น อย่างไรก็ตามบางอัลกอริทึมไม่ได้มีพารามิเตอร์ของโมเดล เช่น K-NN เป็นต้น
- **ไฮเปอร์พารามิเตอร์ (Hyperparameters)** เป็นพารามิเตอร์ประเภทหนึ่งในอัลกอริทึมการเรียนรู้ของเครื่อง พารามิเตอร์ประเภทนี้ไม่สามารถประมาณค่าจากข้อมูลโดยตรงด้วยวิธีการทางสถิติ แต่จะใช้การกำหนด/ปรับแต่งค่าโดยตัวผู้วิเคราะห์เอง **ในเชิงเทคนิคเรียกการปรับแต่งค่าดังกล่าวว่า hyperparameter tuning** การปรับแต่งค่าของ hyperparameter ดังกล่าวจะใช้วิธีการทดลองกำหนดค่า hyperparameter จำนวนหนึ่งให้กับอัลกอริทึม จากนั้นเลือกใช้ค่า hyperparameter ที่ทำให้ค่าประสิทธิภาพของโมเดลทำนายสูงที่สุด ทั้งนี้การพิจารณาประสิทธิภาพดังกล่าวจะพิจารณาบนชุดข้อมูลอีกชุดหนึ่งที่เรียกว่า **validation dataset**

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

6.7 Data Partitioning

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

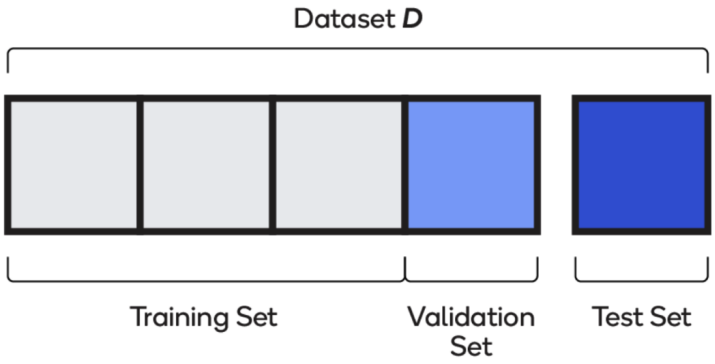


Figure 13: training, validation และ testing dataset

โดยปกติการแบ่งส่วนข้อมูลดังกล่าวไม่ได้มีกฎเกณฑ์ตายตัวว่าควรแบ่งส่วนใดอย่างละเท่าไหร่ 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)

6.7.1 ชุดข้อมูล `mpg`

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

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

```
## # A tibble: 6 x 11
##   manufa~1 model displ  year   cyl trans drv
##   <chr>    <chr> <dbl> <int> <int> <chr> <chr>
## 1 audi     a4      1.8  1999     4 auto~ f
## 2 audi     a4      1.8  1999     4 manu~ f
## 3 audi     a4      2    2008     4 manu~ f
## 4 audi     a4      2    2008     4 auto~ f
## 5 audi     a4      2.8  1999     6 auto~ f
## 6 audi     a4      2.8  1999     6 manu~ f
## # ... with 4 more variables: cty <int>,
## #   hwy <int>, fl <chr>, class <chr>, and
## #   abbreviated variable name 1: manufacturer
```

```
glimpse(mpg)
```

```
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi",~
## $ model        <chr> "a4", "a4", "a4", "a4",~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8~
## $ year         <int> 1999, 1999, 2008, 2008,~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4,~
## $ trans        <chr> "auto(l5)", "manual(m5)~
## $ drv          <chr> "f", "f", "f", "f", "f"~
## $ cty          <int> 18, 21, 20, 21, 16, 18,~
## $ hwy          <int> 29, 29, 31, 30, 26, 26,~
## $ fl           <chr> "p", "p", "p", "p", "p"~
## $ class        <chr> "compact", "compact", "~
```

6.7.2 การแบ่งข้อมูลด้วยการสุ่มอย่างง่าย

การแบ่งด้วย simple random sampling เป็นการแบ่งโดยสุ่มข้อมูลตามจำนวนที่กำหนดออกมาเป็นชุดข้อมูล training dataset หรือ test dataset โดยการสุ่มดังกล่าวมีข้อสมมุติว่าหน่วยข้อมูลทุกหน่อยในชุดข้อมูลต้นฉบับมีโอกาสที่จะถูกสุ่มขึ้นมาเท่ากันทั้งหมด การแบ่งข้อมูลด้วยวิธีการนี้ใน R สามารถทำได้หลายวิธี แต่ในบทความนี้จะใช้วิธีที่อยู่ภายใต้ framework ของ tidymodels การแบ่งข้อมูลด้วยวิธีการดังกล่าวมีสองขั้นตอน

- ขั้นแรกคือการสร้างกรอบของการแบ่งข้อมูลออกเป็น training และ test data สามารถทำได้ด้วยฟังก์ชัน `initial_split()` ของ package rsample อาร์กิวเมนท์สำคัญที่จะต้องระบุในฟังก์ชันได้แก่ `data` และ `prop`
- ขั้นที่สองคือการแบ่งข้อมูลตามกรอบในขั้นตอนแรก โดยจะใช้ฟังก์ชัน `training()` เพื่อแบ่งชุด training dataset ออกมา และใช้ฟังก์ชัน `testing()` เพื่อแบ่งชุดข้อมูล test dataset ออกมา

```
# 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()
```

6.7.3 การแบ่งข้อมูลด้วยการสุ่มแบบชั้นภูมิ

การแบ่งชุดข้อมูลด้วยการสุ่มแบบชั้นภูมิสามารถทำได้ด้วยฟังก์ชัน `initial_split()` เช่นเดียวกัน แต่จะต้องมีการระบุอาร์กิวเมนท์ของฟังก์ชันเพิ่มเติมได้แก่ `strata` เพื่อระบุตัวแปรตามหรือตัวแปรเกณฑ์ที่จะใช้แบ่งชั้นภูมิก่อนการสุ่มตัวอย่าง และ `breaks` ใช้กำหนดจำนวนอันตรภาคชั้นของตัวแปรตามหรือตัวแปรเกณฑ์ที่จะใช้แบ่งชั้นภูมิหากตัวแปรดังกล่าวเป็นตัวแปรเชิงปริมาณ ค่าเริ่มต้นของอาร์กิวเมนท์นี้กำหนดให้ `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()
```

รูปด้านล่างแสดงการเปรียบเทียบการแจกแจงของตัวแปรตามระหว่างชุดข้อมูลต้นฉบับ (full dataset), training และ test dataset ที่แบ่งด้วยวิธีการสุ่มตัวอย่างแบบง่าย และแบบชั้นภูมิ

6.8 Modeling Process

จาก concept ที่กล่าวในหัวข้อ 6.4 - 6.6 จึงมีการออกแบบกระบวนการพัฒนา ML model ไว้ดังรูป 15

6.9 Tidymodels Framework

ปัจจุบันมีเครื่องมือที่ช่วยให้ผู้วิเคราะห์สามารถพัฒนา machine model ได้หลายตัว บทเรียนนี้จะกล่าวถึงการใช้โปรแกรม R เพื่อพัฒนา ML model ดังกล่าว ทั้งนี้ต้องทำความเข้าใจก่อนว่า การทำงานบน R แม้จะเป็นปัญหาเดียวกัน ชุดข้อมูลเดียวกัน แต่ผู้วิเคราะห์ต่างคนกันก็มีทางที่จะดำเนินการด้วยวิธีการที่แตกต่างกันได้ (ใน Python หรือโปรแกรมอื่น ๆ ก็เช่นเดียวกัน) วิธีการหนึ่งใน R ที่สามารถ modeling ได้ง่ายและมีประสิทธิภาพคือการใช้ **tidymodels framework** รายละเอียดดังรูป 16

- **package-rsample** ใช้ในงาน resampling ข้อมูล เช่นการสร้าง training/validation/test dataset การสร้าง cross-validation dataset หรือการสร้าง bootstrape dataset ซึ่งได้กล่าวการใช้งานเบื้องต้นไปแล้วใน 6.7 Data Partitioning
- **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

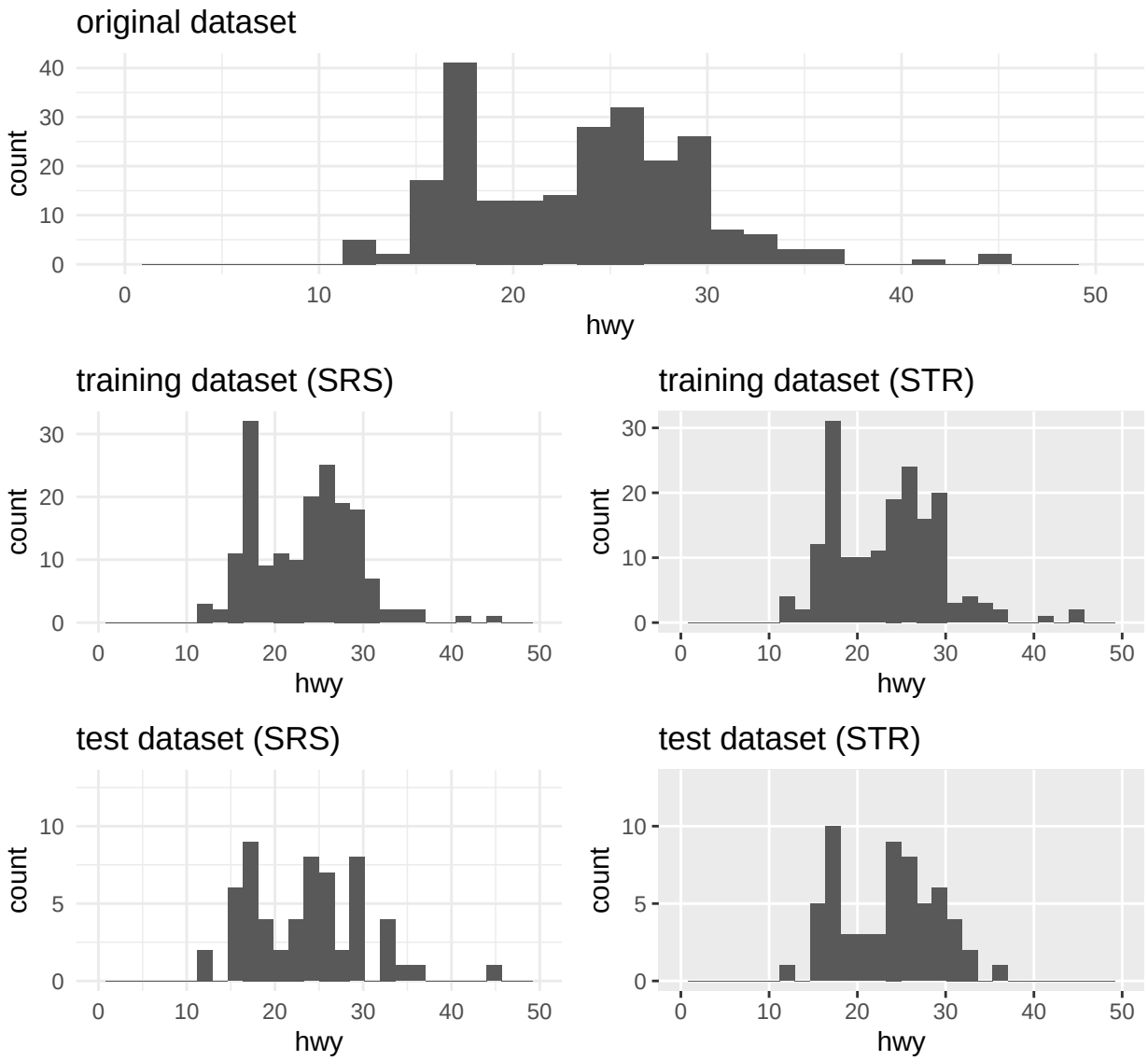


Figure 14: เปรียบเทียบระหว่าง SRS กับ STR

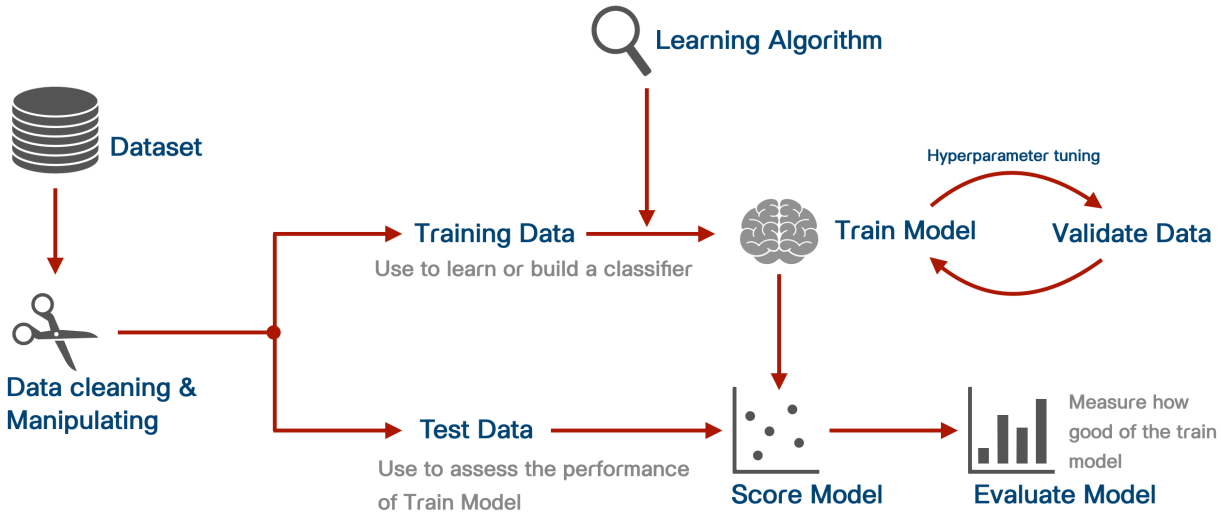


Figure 15: Modeling Process

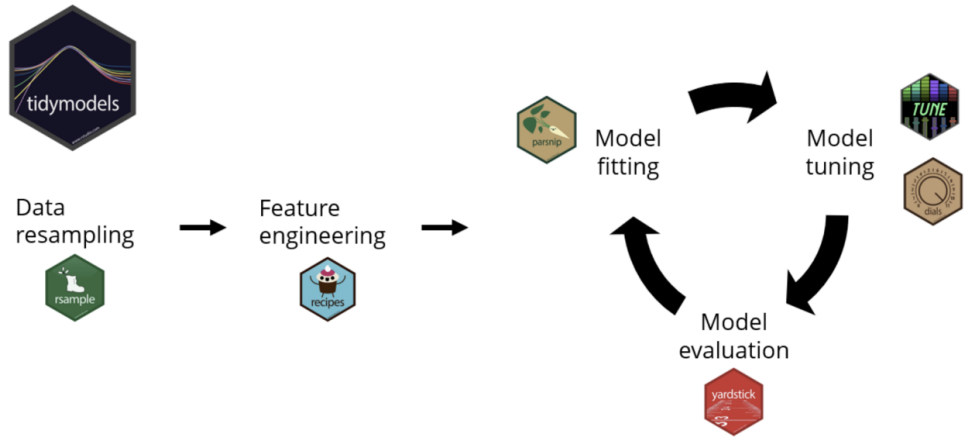


Figure 16: tidymodel framework

tidymodels ถูกพัฒนาขึ้นโดยได้รับการออกแบบให้สามารถทำซ้ำกระบวนการพัฒนาโมเดลได้ง่าย โดยใช้ไวยกรณ์ของภาษาในลักษณะเดียวกัน และถูกออกแบบโดยเน้นใช้กับ supervised learning เป็นหลัก ผู้ใช้งานไม่จำเป็นต้องติดตั้งทุก package ในข้างต้นด้วยตนเอง แต่ติดตั้งเพียง package-tidymodels ก็สามารถใช้งานทุก package ภายใต้ framework ดังกล่าวได้แล้ว โดยการพิมพ์คำสั่งต่อไปนี้

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

6.10 Fitting and Evaluating ML models via tidymodels framework

จากชุดข้อมูล `mpg` ที่ได้ดำเนินการแบ่งส่วนข้อมูลเป็นชุดข้อมูลฝึกหัด และชุดข้อมูลทดสอบแล้ว ซึ่งอยู่ในส่วน Data resampling ของรูป 15 แล้ว นอกจากนี้ชุดข้อมูลดังกล่าวถูกจัดเตรียมมาเรียบร้อยจึงยังไม่ต้องดำเนินการในส่วน feature engineering เนื้อหาในหัวข้อนี้จึงจะนำชุดข้อมูลทั้งสองมาพัฒนาและตรวจสอบประสิทธิภาพของโมเดลตาม tidymodels framework โดยใช้ package parsnip และ yardstick ตามลำดับ

6.10.1 Fitting models using parsnip package

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 ผู้วิเคราะห์อาจจะต้องยุ่งเกี่ยวกับ 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 ที่แตกต่างกันมักมีแนวคิดและไวยกรณ์การเขียนคำสั่งที่แตกต่างกัน ทำให้เป็นอุปสรรคต่อการทำงานโดยเฉพาะการทำซ้ำในอนาคต จากปัญหานี้ tidymodels จึงมีการพัฒนา package parsnip ขึ้นเพื่อเป็น interface สำหรับใช้ package ใน R ที่เกี่ยวข้องกับการ fit supervised learning ทั้งนี้ parsnip ได้ถูกออกแบบมาให้การสั่งงานทั้งหมดอยู่ภายใต้ไวยกรณ์แบบเดียวกัน 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 มีส่วนประกอบ 3 ส่วนที่จำเป็นได้แก่

- **model type** หรืออัลกอริทึมการเรียนรู้ของเครื่องที่ผู้วิเคราะห์จะใช้ในการทำงาน
- **engine** หรือ package ของ R ที่จะใช้สำหรับประมวลผล model type ที่เลือก
- **mode** สำหรับกำหนดว่าปัญหาที่ทำงานด้วยอยู่นี้เป็น regression หรือ classification

รายละเอียดว่าผู้วิเคราะห์สามารถกำหนด model type, engine และ mode แบบใดได้บ้างและต้องกำหนดอย่างไร สามารถศึกษาได้จาก https://www.tidymodels.org/find/parsnip/ รูปด้านล่างแสดงค้นหาสำหรับอัลกอริทึม linear regression

จากผลการค้นหาในรูปด้านล่างจะเห็นว่าการ fit linear regression ด้วย parsnip สามารถทำได้ด้วย model type คือ `linear_reg()` เมื่อพิจารณาในคอลัมน์ engine จะเห็นว่าการ fit linear regression มี engine จำนวนมากที่สามารถใช้เพื่อประมาณค่าพารามิเตอร์ของโมเดลได้ engine ดังกล่าวจริง ๆ แล้วคือ package ต่าง ๆ ของ R ที่ใช้ประมวลผล mode type ที่เลือกไว้ได้ ผู้อ่านจะเห็นว่า model type แบบ `linear_reg` มี engine ที่สามารถใช้ประมวลผลได้จำนวนมาก ซึ่งมีความเหมือนและความแตกต่างกัน เนื้อหาส่วนนี้มีความละเอียดและลึกมาก จึงขอไม่กล่าวถึงในที่นี้

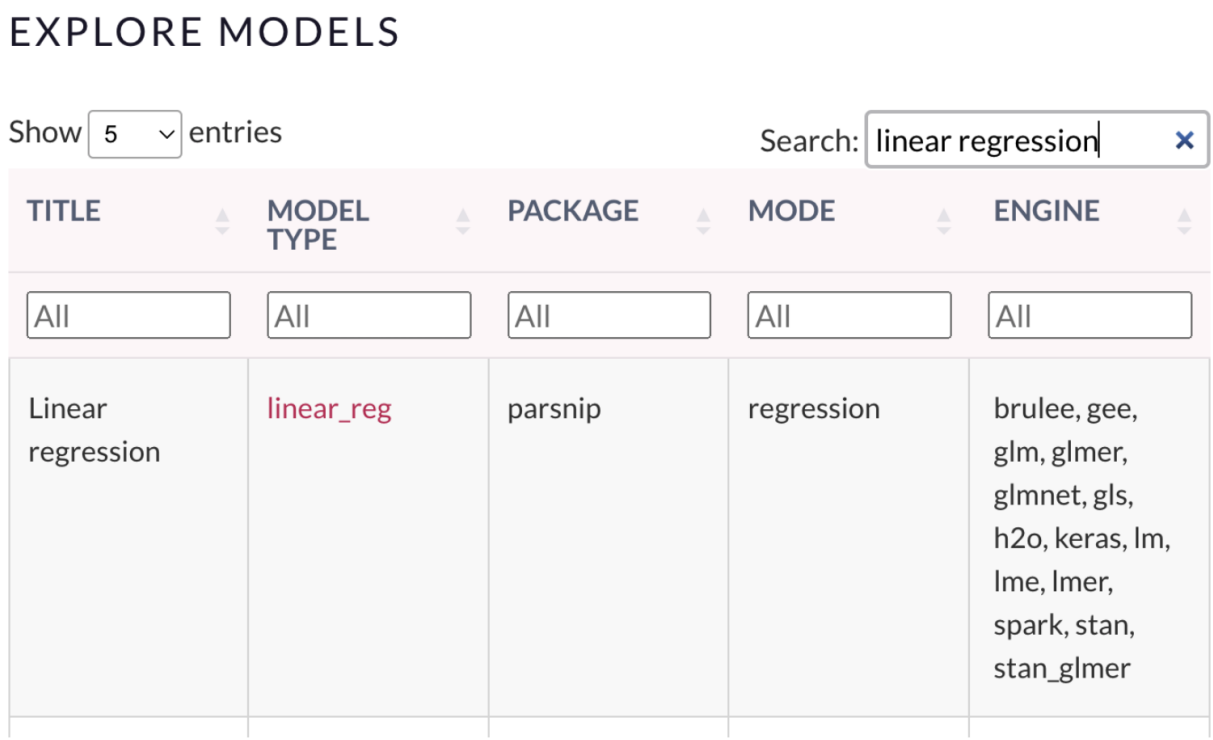


Figure 17: parsnip manual

ในคู่มือข้างต้นยังมีเครื่องมือให้ค้นหาการกำหนดอาร์กิวเมนท์ของฟังก์ชัน model type ในข้างต้น จากรูปด้านล่างจะเห็นรายละเอียดในการกำหนดอาร์กิวเมนท์ของฟังก์ชัน `linear_reg()` เมื่อกำหนด engine ในลักษณะต่าง ๆ

ความหมายของการกำหนดอาร์กิวเมนท์แต่ละค่าสามารถศึกษาได้จากคู่มือของฟังก์ชัน `linear_reg()` ซึ่งสามารถกด hyper-link จากคู่มือข้างต้นเข้าไปศึกษาได้เลย (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

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 18: argument ในฟังก์ชัน model type

สมมุติว่าผู้วิเคราะห์ต้องการพัฒนาโมเดลการเรียนรู้ของเครื่องด้วยอัลกอริทึม 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()` ซึ่งมีอาร์กิวเมนท์สำคัญ 2 ตัวได้แก่ model formula และ training dataset ที่จะใช้สำหรับฝึกหัดโมเดล

การเขียน model formula จะเขียนอยู่ในรูปของ `y ~ x1+x2+x3+…` โดยที่ `y` คือตัวแปรตาม ส่วน `x1, x2, x3 ,…` คือตัวแปรอิสระภายในชุดข้อมูลฝึกหัด และสัญลักษณ์ `~` หมายความว่า “regress on” ในกรณีที่ต้องการใช้ตัวแปรที่เหลือในชุดข้อมูลทั้งหมดเป็นตัวแปรทำนาย สามารถเขียน model formula สั้น ๆ ได้ดังนี้ ‘y ~ .‘ ตัวอย่างต่อไปนี้แสดงการส่งผ่าน model specification `lm_model` ในข้างต้นไปประมวลผล

```
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()` ซึ่งช่วยสร้างตารางสรุปผลลัพธ์จากการประมาณค่าพารามิเตอร์หรือการเรียนรู้ของโมเดลทำนายที่ใช้ให้อยู่ในรูปแบบเดียวกัน ดังนี้

```
tidy(fit_lm)
```

```
## # A tibble: 2 x 5
##   term        estim~1 std.e~2 stati~3  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
## # ... with abbreviated variable names
## #   1: estimate, 2: std.error, 3: statistic
```

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

6.10.2 Prediction

ผู้วิเคราะห์สามารถนำโมเดลที่ผ่านการ train เรียบร้อยแล้วไปใช้หาค่าทำนาย โดยส่งผ่านโมเดลที่ train แล้ว (ในที่นี้คือ `fit_lm`) ไปยังฟังก์ชัน `predict()` ที่มีอาร์กิวเมนท์สำคัญคือ `new_data` ตัวอย่างด้านล่างแสดงนำ `fit_lm` ไปทำนายตัวแปร `hwy` ในชุดข้อมูลทดสอบ

```
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
```

ผลลัพธ์ที่ได้จากการทำนายจะเป็นตารางแบบ tibble ที่แต่ละ row คือค่าทำนายของหน่วยข้อมูลใน row เดียวกันกับใน `test_str`

เมื่อได้ค่าทำนายในชุดข้อมูลทดสอบมาแล้ว ขั้นตอนถัดไปคือการประเมินประสิทธิภาพของโมเดลทำนาย โดยทั่วไปผู้วิเคราะห์มักจะรวมค่าทำนายที่ได้ (ในที่นี้คือ `hwy_pred`) ไปไว้อยู่ภายในชุดข้อมูลทดสอบ การดำเนินการนี้สามารถทำได้หลายวิธีการขึ้นอยู่กับว่าถนัดจะดำเนินการแบบนี้ 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
```

6.10.3 Evaluating models using yardstick package

ประสิทธิภาพในการทำนายของโมเดลสามารถประเมินได้หลายลักษณะ ในเชิงเทคนิคจะเรียกเกณฑ์ที่ใช้สำหรับประเมินประสิทธิภาพของโมเดลการเรียนรู้ของเครื่องว่า **evaluation metric**

Metric ดังกล่าวอาจจำแนกเป็น 2 ประเภท ตามประเภทของ supervised learning ได้แก่ metric สำหรับประเมินประสิทธิภาพของ regression model และ classification model ในหัวข้อนี้จะกล่าวถึง metric สำหรับ regression model ก่อน

สำหรับ metric ที่นิยมใช้ประเมินประสิทธิภาพของ regression model ได้แก่ RMSE (root mean squared error) และ R squared (coefficient of determination) การคำนวณค่าประสิทธิภาพดังกล่าวสามารถทำได้โดยใช้ฟังก์ชันจาก package yardstick ได้แก่ `rmse()` และ `rsq()` ตามลำดับ ตัวอย่างต่อไปนี้แสดงการเขียนคำสั่งเพื่อคำนวณ metric ทั้งสอง

ในทางทฤษฎีค่า RMSE มีความหมายเป็นค่าคลาดเคลื่อนในการทำนายโดยเฉลี่ยของโมเดล สามารถคำนวณได้จากสูตร

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

ส่วน R squared มีความหมายเป็นสัดส่วนของความผันแปรที่ร่วมกันระหว่างค่าจริงของตัวแปรตามกับค่าทำนายของตัวแปรตามที่ได้จากโมเดลทำนาย การคำนวณค่า R squared สามารถทำได้ง่าย ๆ ด้วยการหาค่ากำลังสองของสัมประสิทธิ์สหสัมพันธ์ระหว่างค่าจริงกับค่าทำนายของตัวแปรตามดังกล่าว

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

```
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 เพื่อประเมินความสอดคล้องกันระหว่างค่าจริงของตัวแปรตามกับค่าทำนายได้ การสร้าง R squared plot ใน R สามารถทำได้หลายวิธี ทั้งการใช้ package graphic ซึ่งเป็น package พื้นฐานสำหรับสร้าง visualization ใน R ดังนี้

```
# 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")
```

หรือใช้ ggplot2 ซึ่งเป็น package หนึ่งภายใต้ tidyverse framework

```
# 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))
```

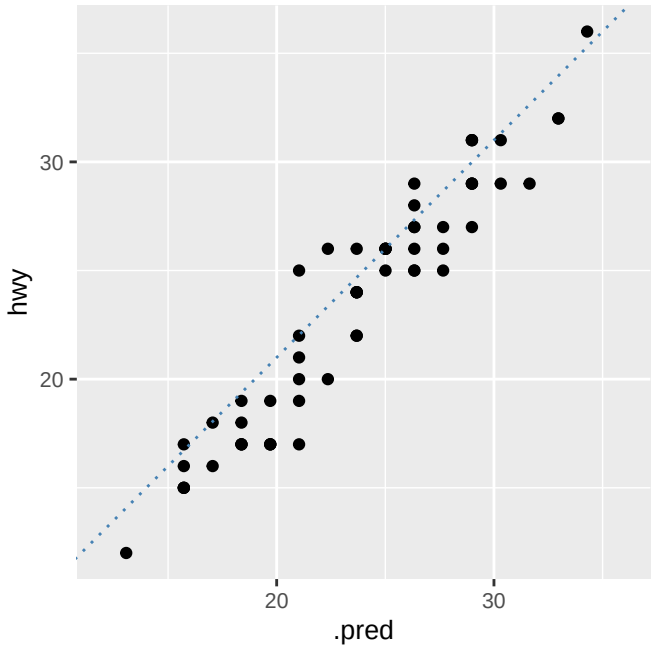


Figure 19: R squared plot via ggplot2

คำถาม

linear regression มีโมเดลทางสถิติที่มีข้อตกลงเบื้องต้นที่ค่อนข้างเข้มงวด ได้แก่ independence, homoscedasticity, normality, no multicollinearity, no outlier, ... คำถามคือในการพัฒนา ML model ดังกล่าวจำเป็นมั้ยที่จะต้องตรวจสอบข้อตกลงเบื้องต้นดังกล่าว เพราะอะไร?

6.11 กิจกรรม 4 : พัฒนา regression model ด้วย tidymodel framework

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