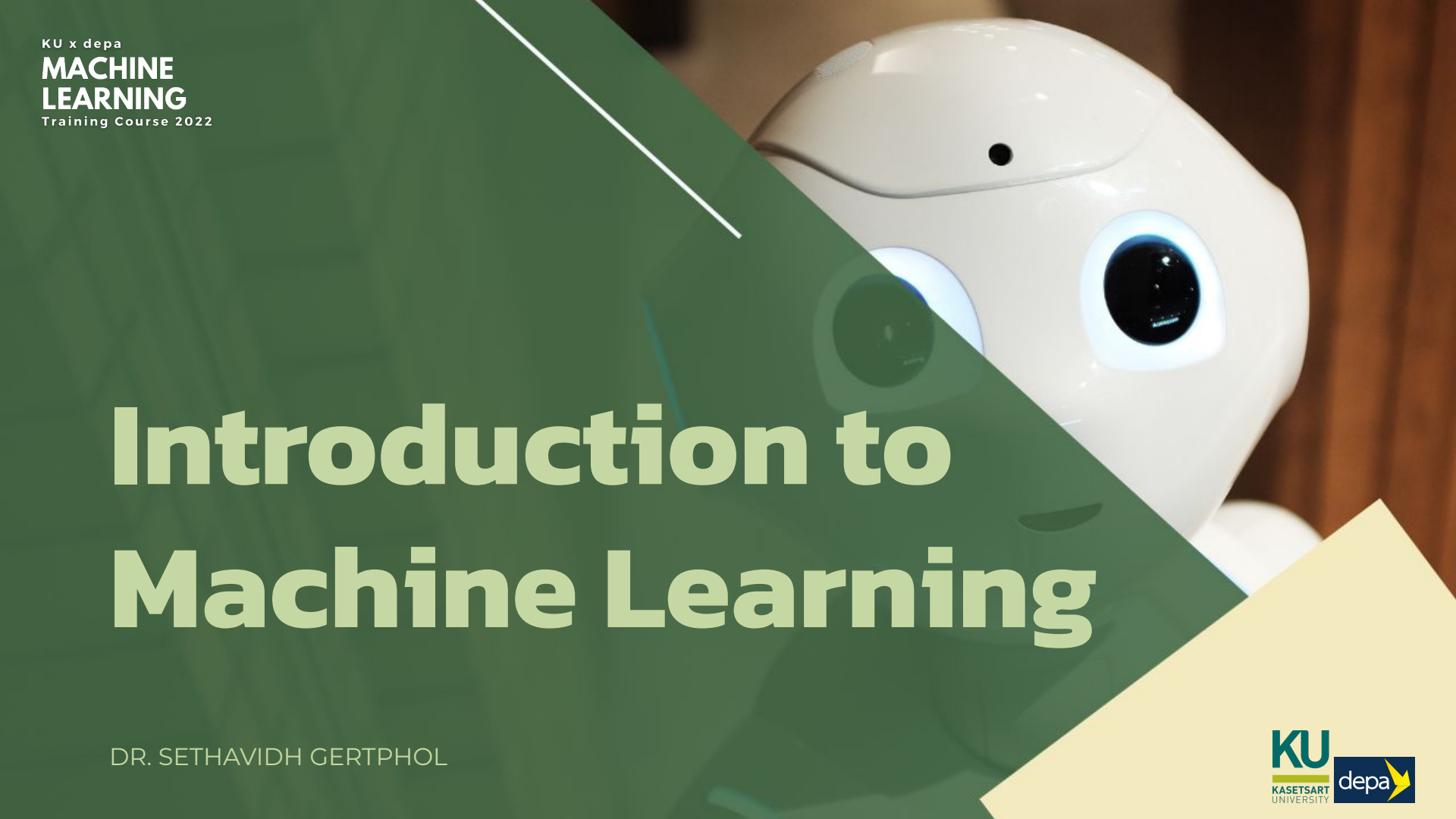
KU x depa

MACHINE LEARNING

Training Course 2022



Introduction to Machine Learning

DR. SETHAVIDH GERTPHOL

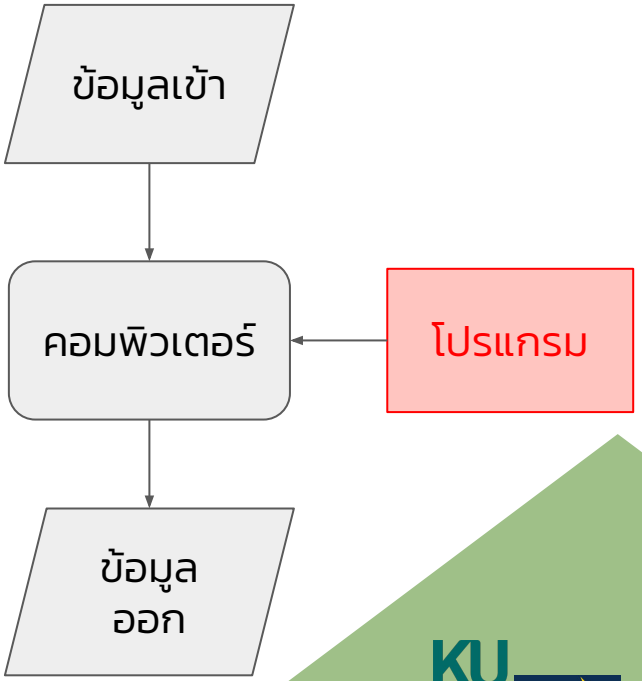
KU KASETSART UNIVERSITY depa

Today's Outline

- Data Modeling
- Model and Program
- Machine Learning Problems
- Model Evaluation Metrics
- Train Validation Test Set
- Overfitting / Underfitting

Data Modeling

- โปรแกรมคือตัวแบบในการเปลี่ยน input ให้เป็น output
  - แสดงความสัมพันธ์ระหว่าง input กับ output
- โมเดลก็คล้ายกับโปรแกรม
- ข้อแตกต่างคือ
  - โปรแกรมนั้นถูกสร้างขึ้นโดยโปรแกรมเมอร์
  - โมเดลนั้นถูกอุปนัยขึ้นจากข้อมูล
- ทางสถิติจะเรียก input เป็น ตัวแปรต้น และ output เป็น ตัวแปรตาม



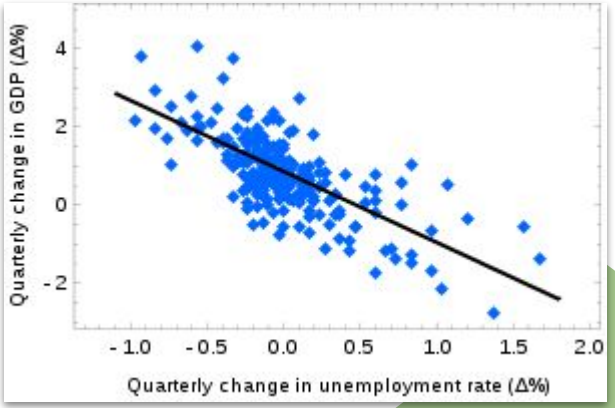
ตัวแปรต้นและตัวแปรตาม

- ค่าของตัวแปรตามจะขึ้นกับตัวแปรต้น
- ศาสตร์ต่างกันอาจจะเรียกชื่อตัวแปรต้นตัวแปรตามต่างกัน

ตัวแปรต้น	ตัวแปรตาม	ศาสตร์
Input	Output	Computer Science
Independent variable	Dependent variable	คณิตศาสตร์, สถิติ
Regressor, control variable, explanatory variable	Response variable, outcome variable	สถิติ
Feature, attribute	Label, target attribute	Machine Learning

โมเดล กับ โปรแกรม

- เราสร้างโปรแกรมเมื่อเข้าใจกฎเกณฑ์ในการเปลี่ยน input เป็น output ชัดเจน
  - ค่าจอดรถ: 15 นาทีแรกฟรี ต่อไปชั่วโมงละ 20 บาท เศษของชั่วโมงคิดเป็นหนึ่งชั่วโมง สูงสุด 8 ชั่วโมง
- เราสร้างโมเดลในกรณีที่กฎเกณฑ์ในการแปลงข้อมูลเข้าเป็นข้อมูลออกนั้นซับซ้อน ไม่ชัดเจน หรือมีความไม่แน่นอนเข้ามาเกี่ยวข้อง
  - การเปลี่ยนแปลงจำนวนผู้ว่างงานส่งผลต่อ GDP อย่างไร
  - สร้างโมเดลจากข้อมูลการเปลี่ยนแปลงของ GDP กับ การเปลี่ยนแปลงจำนวนคนว่างงานในแต่ละไตรมาส
  - %Change GDP = 0.789 – 1.654*(Change Unemployment Rate)



https://en.wikipedia.org/wiki/Okun%27s_law

ประเภทของข้อมูล

- ข้อมูลตัวเลข (numeric, continuous)
  - สามารถนำมาคำนวณได้ เช่น จำนวนประชากร
- ข้อมูลแบบประเภท (category, nominal)
  - ไม่มีความสัมพันธ์ระหว่างประเภท เช่น ชาย/หญิง หรือ ชื่อจังหวัด
- ข้อมูลแบบลำดับ (ordinal)
  - สามารถเรียงลำดับประเภทได้ เช่น ดี/ปานกลาง/พอใช้ หรือ ร้อน/อบอุ่น/เย็น/หนาว
- ข้อมูลแบบช่วง (interval)
  - มีลำดับและวัดความแตกต่างระหว่างลำดับได้เป็นช่วง เช่น ปีค.ศ.
- ตัวแปรต้นและตัวแปรตามเป็นประเภทไหนก็ได้

วิธีการสร้างโมเดล

- วิธีการทางสถิติ
  - ใช้คณิตศาสตร์ทางสถิติในการสร้างโมเดล
- วิธีการทาง Machine Learning
  - ใช้วิธีด้าน optimization ในการสร้างโมเดล
- ตอนนี้ทั้งสองวิธีเริ่มใช้รวมกันและไม่มีเส้นแบ่งที่ชัดเจนนัก

ประเภทของโมเดล

- Regression
  - หาความสัมพันธ์ระหว่างตัวแปรต้นกับตัวแปรตาม
  - ตัวแปรตามต้องเป็นประเภทตัวเลข (numeric)
  - ตัวแปรต้นอาจมีมากกว่าหนึ่งตัว แต่ไม่จำเป็นต้องเป็นประเภท numeric ทุกตัว
  - เช่น โมเดลทำนายอุณหภูมิของเมือง ด้วยอุณหภูมิของเมืองใกล้ ๆ
- Classification
  - ทำนายว่าตัวอย่าง(ใหม่)จัดอยู่ในประเภทใด
  - ตัวแปรตามต้องเป็นประเภท category
  - เช่น โมเดลทำนายพันธุ์ดอกไอริส ด้วยความยาวและความกว้างกลีบจริงและกลีบเลี้ยง
- Regression กับ Classification ถือเป็น Supervised Learning

  คือตัวอย่างต้องมีผลเฉลย (ตัวแปรตาม)

ตัวอย่างข้อมูล

คอลัมน์ผลเฉลย (label)

	city	country	latitude	longitude	temperature
41	Bradford	United Kingdom	53.80	-1.75	8.39
198	Trikala	Greece	39.56	21.77	16.00
70	Daugavpils	Latvia	55.88	26.51	5.38
175	Salzburg	Austria	47.81	13.04	4.62
124	Limoges	France	45.83	1.25	10.32
83	Eskisehir	Turkey	39.79	30.53	11.11
29	Bialystok	Poland	53.15	23.17	6.07
121	Kryvyy Rih	Ukraine	47.93	33.34	8.61
181	Sivas	Turkey	39.75	37.03	8.05

regression

	sepal length	sepal width	petal length	petal width	iris
91	6.1	3.0	4.6	1.4	Iris-versicolor
89	5.5	2.5	4.0	1.3	Iris-versicolor
115	6.4	3.2	5.3	2.3	Iris-virginica
24	4.8	3.4	1.9	0.2	Iris-setosa
76	6.8	2.8	4.8	1.4	Iris-versicolor
136	6.3	3.4	5.6	2.4	Iris-virginica
138	6.0	3.0	4.8	1.8	Iris-virginica
62	6.0	2.2	4.0	1.0	Iris-versicolor
29	4.7	3.2	1.6	0.2	Iris-setosa

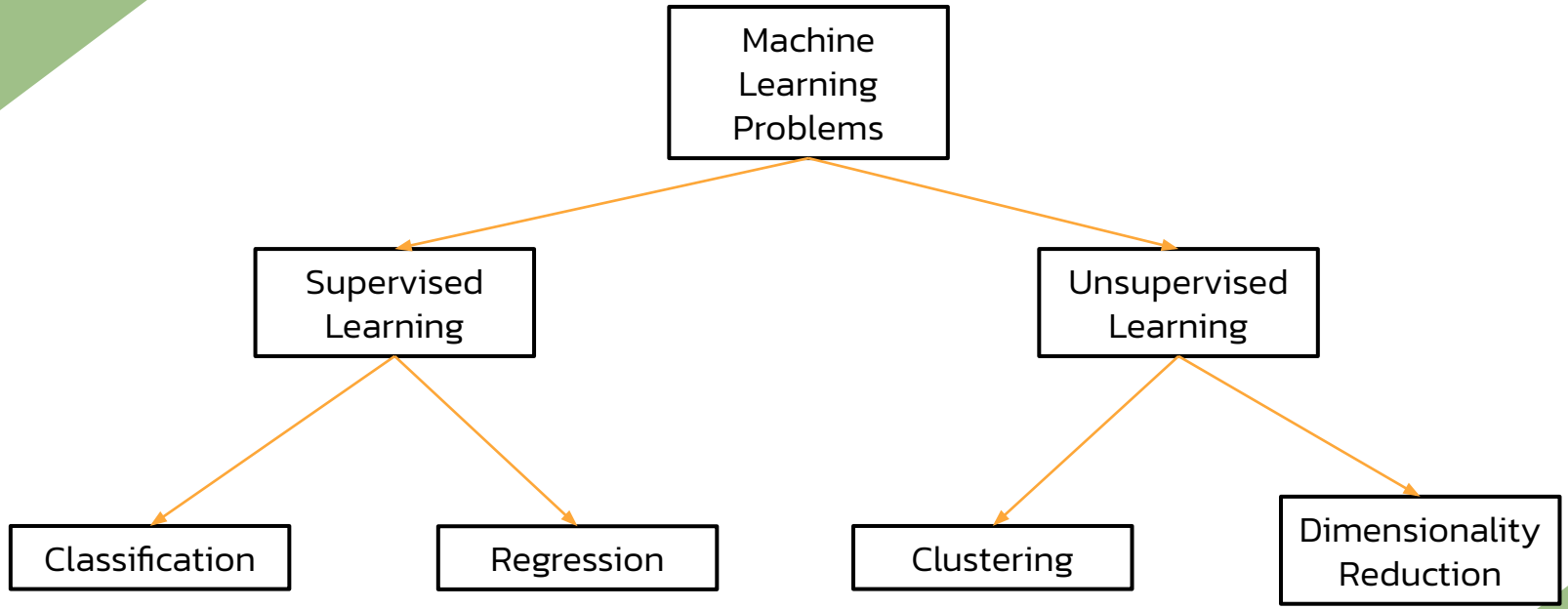
classification

ประเภทของโมเดล

- Clustering
  - ใช้จัดข้อมูลให้เป็นกลุ่ม โดยตัวอย่างในกลุ่มเดียวกันจะมีความ"เหมือนกัน" มากกว่าตัวอย่างนอกกลุ่ม
  - ความเหมือนกันนิยามได้หลากหลายวิธี
  - เช่น จัดกลุ่มคะแนนสอบให้เป็น 7 กลุ่มเพื่อตัดเกรด, จัดกลุ่มลูกค้าตามประเภทและราคาสินค้าที่ซื้อ
- Dimensionality reduction
  - ใช้ลดจำนวนตัวแปรต้น (คอลัมน์) ของชุดข้อมูล โดยยังคงปริมาณสารสนเทศไว้
  - ลดขนาดของโมเดล เพิ่มความเร็วในการเรียนรู้และทำนาย
- ทั้ง Clustering และ Dimensionality reduction จัดเป็น **Unsupervised Learning**

  คือตัวอย่างไม่มีผลเฉลย

ประเภทของโมเดล

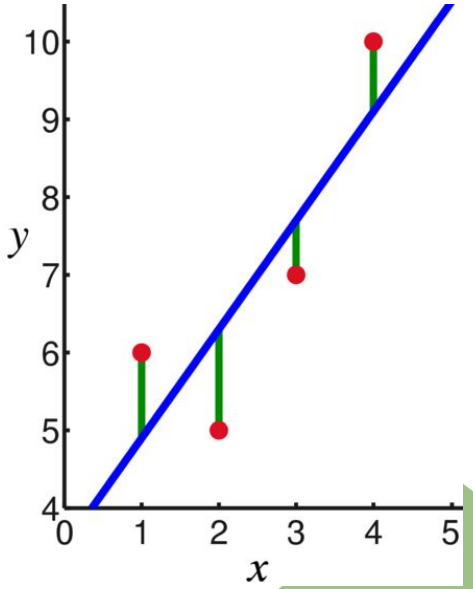


Evaluation Metrics for Supervised Learning Models

- การสร้างโมเดลจากข้อมูลนั้นเป็นใช้วิธีการทางอุปนัย (induction) เพื่อ"เหมารวม" (generalization) สิ่งประเภทนั้นทั้งหมดจากตัวอย่างที่มี
  - ไม่สามารถ"พิสูจน์"ได้ว่าโมเดลที่ได้นั้นถูกต้อง
- จึงต้องใช้มาตรวัดความใกล้เคียงของผลการทำนายของโมเดลเทียบกับเฉลย/คำตอบ
- ตอนนี้จะกล่าวถึงการวัดความถูกต้องของโมเดลระหว่างการสร้างโมเดล
  - วัดจากผลเฉลยของชุดข้อมูลที่มีอยู่แล้ว
  - ดังนั้นจะใช้กับ Model แบบ Supervised Learning เท่านั้น

Regression Model Metrics

- ผลเฉลยและผลทำนายที่ได้จากโมเดลเป็นค่าตัวเลข ดังนั้นใช้การคำนวณหามาตรวัดได้
- Error: ผลต่างระหว่างผลเฉลยและค่าทำนาย
  - ถ้านำ Error ของหลายการทำนายมาบวกกันอาจมีค่าที่หักล้างกันไป
- Absolute Error: ค่าสัมบูรณ์ของผลต่างระหว่างค่าทำนายและผลเฉลย
  - ผลต่างจะเป็นค่าบวกเสมอ
- Squared Error: ค่ายกกำลังสองของผลต่างระหว่างค่าทำนายและผลเฉลย
  - ผลต่างจะเป็นค่าบวกเสมอ ผลต่างที่มากจะทำให้ค่า squared error ยิ่งมากตามไปด้วย
- Percentage Error: Error หารด้วยผลเฉลย คิดเป็นเปอร์เซ็นต์



Regression Model Metrics

- **Mean Absolute Error (MAE)**: นำ Absolute error ของทุกการทำนายมาหาค่าเฉลี่ย
- **Mean Squared Error (MSE)**: นำ Squared Error ของทุกการทำนายมาหาค่าเฉลี่ย
- **Root Mean Squared Error (RMSE)**: นำ Squared Error ของทุกการทำนายมาหาค่าเฉลี่ย แล้วถอดรากที่สอง
  - หน่วยของมาตรวัดจะเป็นหน่วยเดียวกันกับข้อมูล
- **Mean Absolute Percentage Error (MAPE)**: นำ Percentage error มาหาค่า Absolute แล้วเฉลี่ยกัน

R-squared

- มาตรวัดว่าโมเดลอธิบายความแปรปรวนของผลเฉลยได้มากแค่ไหน
- ค่า R2 โดยทั่วไปอยู่ระหว่าง 0-1
  - ถ้า R2 = 0.7 : ความแปรปรวน 70% ของผลเฉลยอธิบายได้ด้วยโมเดล
- เช่น ความแปรปรวนของอุณหภูมิของเมืองในยุโรปจากชุดข้อมูลคือ 12.68
  - นั่นคือถ้าให้เดาค่าอุณหภูมิของเมืองโดยไม่มีข้อมูลอื่นเลย ความแปรปรวนของคำตอบจะอยู่ที่ 12.68
- พอสร้างโมเดลเสร็จ ค่า R2 เป็น 0.7
  - ข้อมูลเพิ่มเติมและวิธีการที่ใช้สร้างโมเดลนั้นอธิบายความแปรปรวนไปได้ 70%
    คือ 12.68*0.7 = 8.88 ดังนั้นยังมีความแปรปรวนอีก 3.5 ที่ไม่ครอบคลุมด้วยโมเดล

Classification Model Metrics

- การวัดประสิทธิภาพด้านการแยกแยะของโมเดล มาตรวัดง่ายสุดคือความแม่นยำ (accuracy)
- สมมติว่าตัวอย่างมี 2 classes คือ Yes และ No
- นิยามของความแม่นยำ

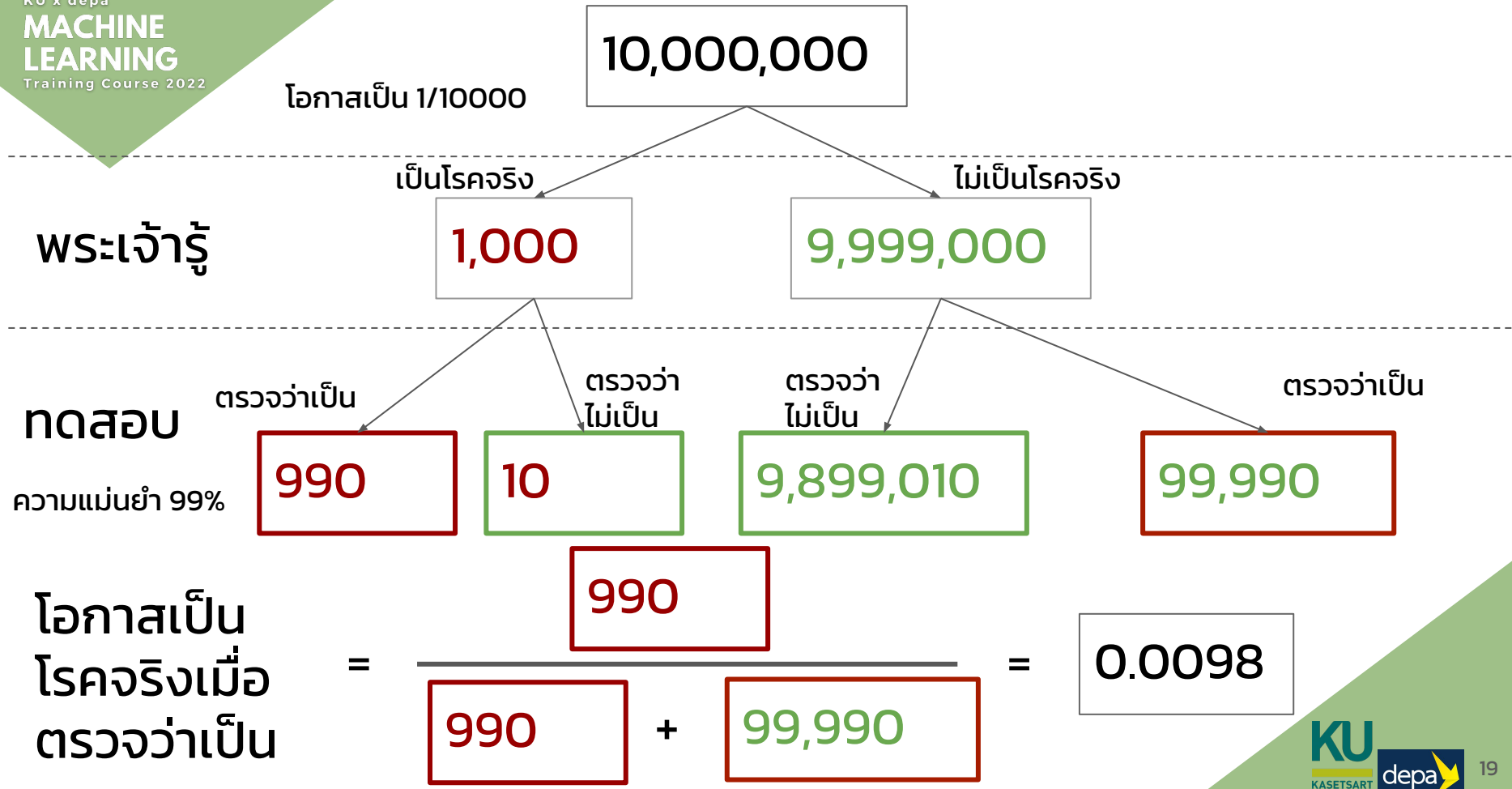
$$\text{ความแม่นยำ} = \frac{\text{จำนวนตัวอย่างที่เป็น Yes และโมเดลทำนายว่าเป็น Yes} + \text{จำนวนตัวอย่างที่เป็น No และ โมเดลทำนายว่าเป็น No}}{\text{จำนวนตัวอย่างทั้งหมด}}$$

ปัญหาของความแม่นยำ

- โมเดลที่แม่นยำ 90% นั้นเป็นโมเดลที่ดีหรือไม่
- เทียบกับอะไร?
- สมมติว่ามีข้อมูล 450 ตัวอย่าง เป็น Class NO 407 ตัวอย่างและ Class YES อีก 43 ตัวอย่าง
- โมเดลโง่ๆ ทำนายว่าเป็น NO เสมอ จะทำนายถูก 407 จาก 450 ตัวอย่าง คิดเป็น 90.44% ดีกว่าโมเดลแรก
- เรียกโมเดลโง่ๆ แบบนี้ว่า Dummy Model เอาไว้ใช้เป็น มาตรฐานการเปรียบเทียบ (baseline)

ปัญหาที่ 2 ของความแม่นยำ

- สมมติว่ามีการตรวจหาโรคที่แม่นยำ 99% ถ้าสุ่มคนขึ้นมา 1 คนแล้วพาไปตรวจโรคด้วยวิธีนี้ แล้วผลการตรวจบอกว่าเป็นโรค โอกาสที่คนนั้นจะเป็นโรคจริงนั้นมีกี่ %
  - 99%
  - 1%
  - ข้อมูลไม่พอที่จะตอบ
- ขึ้นกับความพบยากของโรคด้วย
- สมมติต่อว่าโรคนั้นพบใน 1 คน จาก 10000 คน


10,000,000
โอกาสเป็น 1/10000
เป็นโรคจริง
ไม่เป็นโรคจริง
พระเจ้ารู้
1,000
9,999,000
ทดสอบ
ตรวจว่าเป็น
ตรวจว่า ไม่เป็น
ตรวจว่า ไม่เป็น
ตรวจว่าเป็น
ความแม่นยำ 99%
990
10
9,899,010
99,990
โอกาสเป็น โรคจริงเมื่อ ตรวจว่าเป็น
=
990
990
+
99,990
=
0.0098

Confusion Matrix

- N: จำนวนตัวอย่างทั้งหมด
- TN (True Negative): ทำนายว่าไม่ใช่และตัวอย่างนั้นไม่ใช่จริงๆ
- TP (True Positive): ทำนายว่าใช่และตัวอย่างนั้นใช่จริงๆ
- FN (False Negative): ทำนายว่าไม่ใช่แต่ตัวอย่างนั้นใช่
  - Type II Error
- FP (False Positive): ทำนายว่าใช่แต่ตัวอย่างนั้นไม่ใช่
  - Type I Error
- Confusion Matrix แสดงจำนวนของ TN, TP, FN, FP ทั้งหมด

N = 450	Predicted NO	Predicted YES
Actual NO	TN =400	FP = 7
Actual YES	FN =17	TP = 26

Accuracy: for what fraction of all instances is the classifier's prediction correct (for either positive or negative class)?

	Predicted negative	Predicted positive	
True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
			$N = 450$

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP}$$

$$= \frac{400+26}{400+26+17+7}$$

$$= 0.95$$

Classification error (1 – Accuracy): for what fraction of all instances is the classifier's prediction incorrect?

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
			$N = 450$
	Predicted negative	Predicted positive	

$$\text{ClassificationError} = \frac{FP + FN}{TN + TP + FN + FP}$$

$$= \frac{7+17}{400+26+17+7}$$

$$= 0.060$$

Recall, or True Positive Rate (TPR): what fraction of all positive instances does the classifier correctly identify as positive?

	Predicted negative	Predicted positive	
True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
			$N = 450$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$= \frac{26}{26+17}$$

$$= 0.60$$

Recall is also known as:

- True Positive Rate (TPR)
- Sensitivity
- Probability of detection

Precision: what fraction of positive predictions are correct?

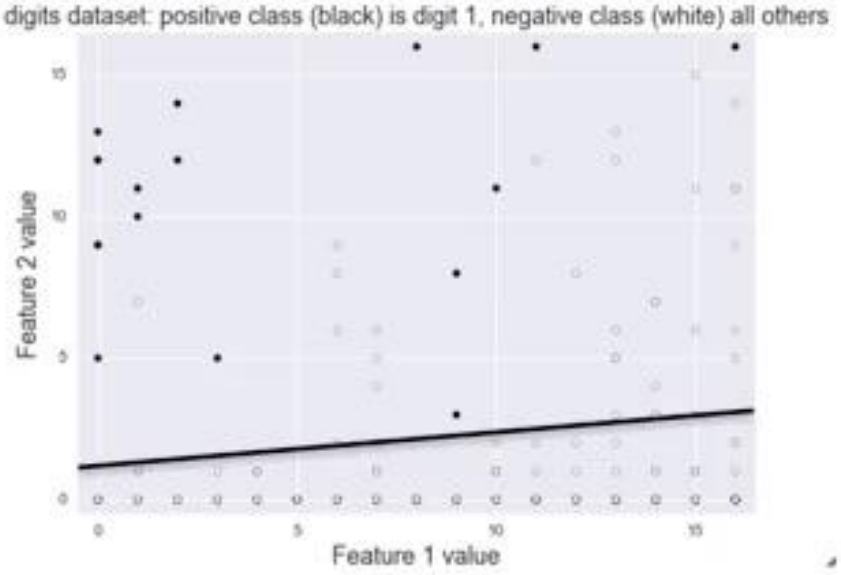
True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
			$N = 450$
	Predicted negative	Predicted positive	

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

$$= \frac{26}{26+7}$$

$$= 0.79$$

Low Precision, High Recall

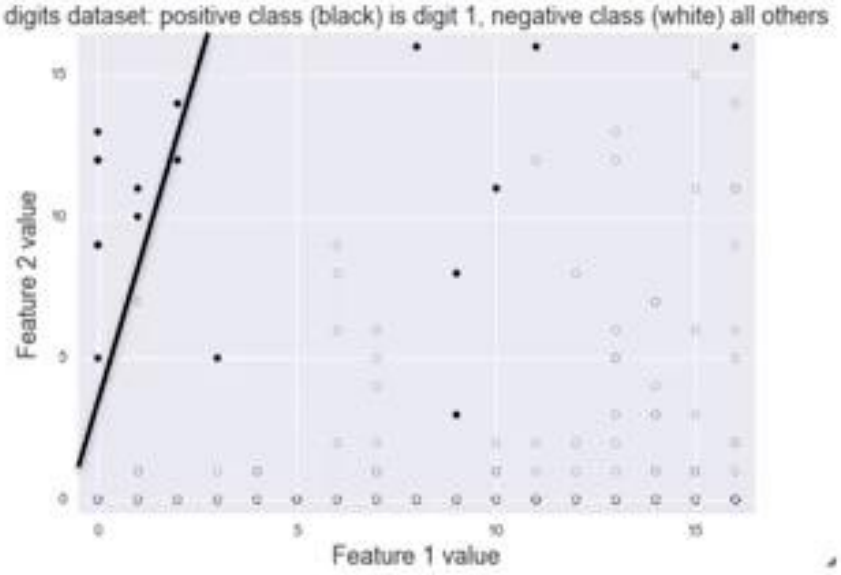


TN = 408	FP = 27
FN = 0	TP = 15

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{15}{42} = 0.36$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{15}{15} = 1.00$$

High Precision, Lower Recall



TN = 435	FP = 0
FN = 8	TP = 7

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{7}{7} = 1.00$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{7}{15} = 0.47$$

Recall v.s. Precision

- งานที่เน้น Recall
  - คืองานที่ต้องการตรวจจับตัวอย่างที่เป็น YES ให้ได้เยอะที่สุด
  - ความเสียหายจากการปล่อยกรณีที่เป็น YES หลุดไปนั้นสูงมาก
  - เช่น ตรวจจับเซลล์มะเร็ง เครื่องตรวจจับเพลิงไหม้
  - ยอมให้ตรวจจับกรณี NO เป็น YES ได้ คัดกรองอีกทีโดยผู้เชี่ยวชาญ
- งานที่เน้น Precision
  - คืองานที่ต้องการผลการทำนาย YES ให้ถูกต้องที่สุด
  - ความเสียหายจากการทำนาย NO เป็น YES นั้นสูงมาก
  - เช่น Search Engine, Recommendation System, งานที่โต้ตอบกับผู้ใช้โดยตรง
  - ยอมให้กรณีที่เป็น YES จริงหลุดรอดการตรวจจับไปได้

F-score: generalizes F1-score for combining precision & recall into a single number

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} = \frac{(1 + \beta^2) \cdot TP}{(1 + \beta^2) \cdot TP + \beta \cdot FN + FP}$$

β allows adjustment of the metric to control the emphasis on recall vs precision:

- **Precision-oriented users: $\beta = 0.5$ (false positives hurt performance more than false negatives)**
- **Recall-oriented users: $\beta = 2$ (false negatives hurt performance more than false positives)**

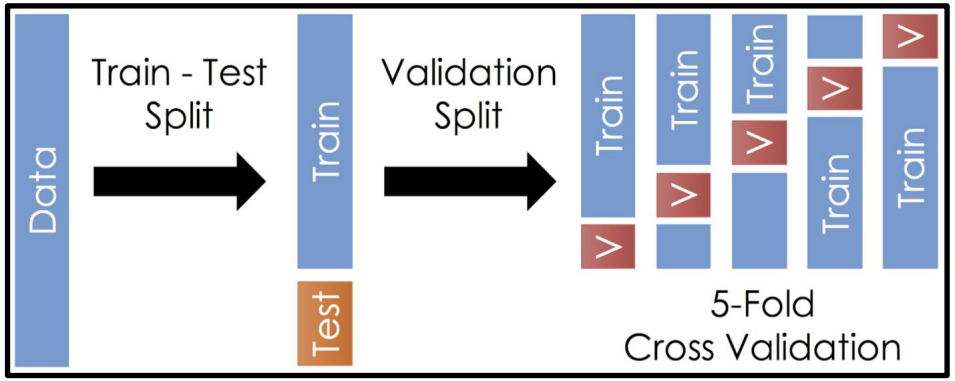
Model Evaluation for Supervised Learning

- Model Evaluation คือ การวัดประสิทธิภาพโมเดลด้วยเทคนิคต่างๆ เพื่อทำให้มั่นใจว่าโมเดลสามารถทำงานได้ดีเมื่อนำไปใช้กับข้อมูลในอนาคต
  - ปัญหาคือเราไม่มีข้อมูลในอนาคตมาทดสอบ
- Hold Out (Train-test split)
  - เทคนิคการแบ่งข้อมูลเป็นสองชุด เรียกว่าชุดฝึก (Training Set) เพื่อใช้สร้างโมเดล และชุดทดสอบ (Test Set) เพื่อวัดประสิทธิภาพของโมเดล
  - โมเดลที่สร้างจากชุดฝึกจะไม่เคยเห็นข้อมูลจากชุดทดสอบ ดังนั้นจึงพอประมาณได้ว่าข้อมูลชุดทดสอบเป็นข้อมูลในอนาคต

k-fold Cross-validation

Train/Test Split, Cross-Validation, & You | by Hector Ian Martinez | Medium

- ปัญหา: ผู้สร้างโมเดลอาจ"จูน"โมเดลไปเรื่อยๆ จนใช้งานกับ Test Set ได้ดี แต่อาจใช้กับข้อมูลจริงในอนาคตได้ไม่ดี
- การทำ k-fold Cross-validation จะแก้ปัญหานี้ โดย

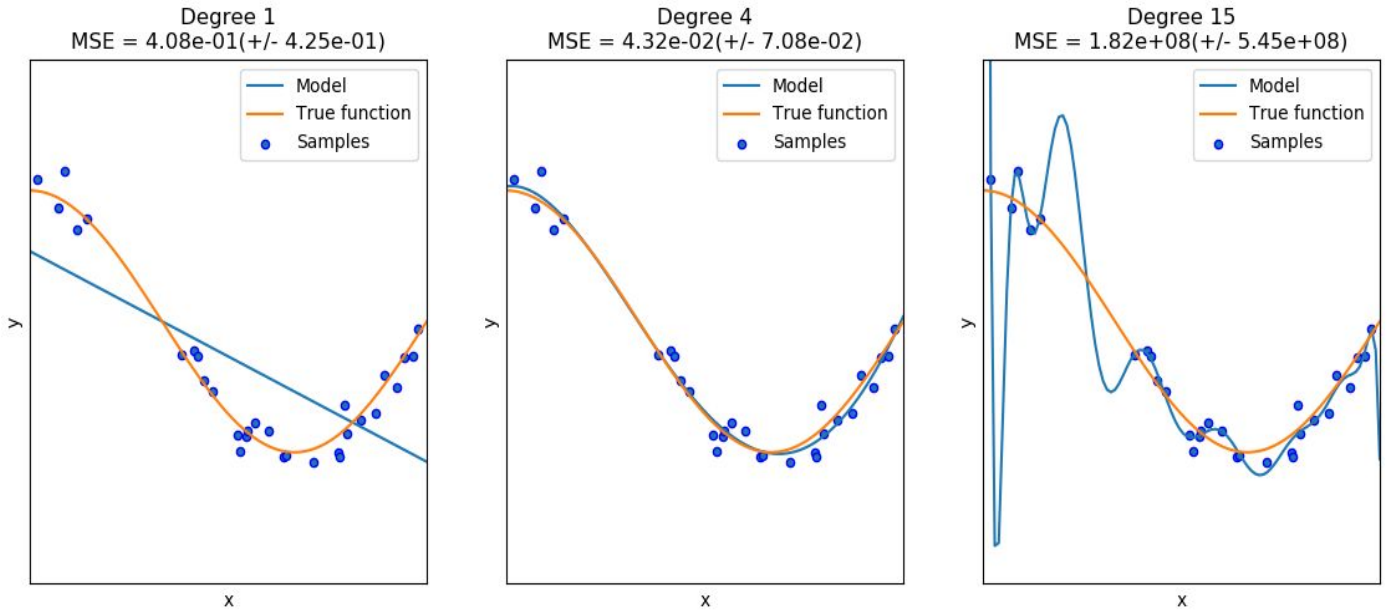


  - แบ่งข้อมูลเป็น k ชุด จะสร้างโมเดลจาก k-1 ชุดและใช้ Validation อีก 1 ชุด (นิยมใช้ k = 10)
  - สร้างโมเดลและทดสอบ k รอบ แต่ละรอบจะเปลี่ยน Training กับ Validation Set ไปเรื่อยๆ ไม่ซ้ำกัน
  - เมื่อทดสอบครบ k ครั้ง จะคำนวณค่าเฉลี่ยผลลัพธ์ของประสิทธิภาพที่ได้
  - สร้างโมเดลด้วยวิธีที่ดีที่สุดโดยใช้ข้อมูลทั้งหมด แล้วค่อยนำไปทดสอบกับ Test Set แค่ครั้งเดียว

Overfitting / Underfitting

- Overfitting:
  - โมเดลที่สร้างได้ มีความอ่อนไหวต่อข้อมูลมากเกินไป
  - เปลี่ยนค่าบางค่าของข้อมูลเข้าเล็กน้อยจะทำให้คำทำนายเปลี่ยนไปอย่างมาก
  - เกิดจากการที่โมเดลเรียนรู้ noise ของข้อมูลเข้าไปด้วย
  - เช็คได้จากค่าของมาตรวัดความถูกต้องสูงมากเมื่อทดสอบกับ training set แต่ต่ำมากเมื่อทดสอบกับ test set
- Underfitting
  - โมเดลที่สร้างได้เก็บ pattern ของข้อมูลได้ไม่ครบถ้วน ไม่ละเอียดพอ
  - ค่าของมาตรวัดความถูกต้องของโมเดลมีค่าต่ำมากทั้งการทดสอบกับ training และ test set

Overfitting / Underfitting



Source: http://scikit-learn.org/stable/_images/sphx_glr_plot_underfitting_overfitting_001.png

Hyperparameter tuning

- Model parameters: ค่าที่เป็นผลลัพธ์ของการเรียนรู้ ส่งผลต่อการทำงานของโมเดล
  - เช่น ค่า slope และ intercept ของ linear regression
- Hyperparameter: ค่าที่กำหนดกระบวนการเรียนรู้ ส่งผลให้การเรียนรู้ของโมเดลเปลี่ยนไป
  - ค่า hyperparameter ที่ต่างกันอาจทำให้ได้โมเดลที่ต่างกัน แม้จะใช้ข้อมูลชุด train ที่เหมือนกัน
  - เช่น จำนวน neighbors ของ K-Nearest Neighbors
- การปรับค่า hyperparameters อาจลดหรือเพิ่มการ overfit หรือทำให้โมเดลที่ได้มีความแม่นยำมากขึ้น