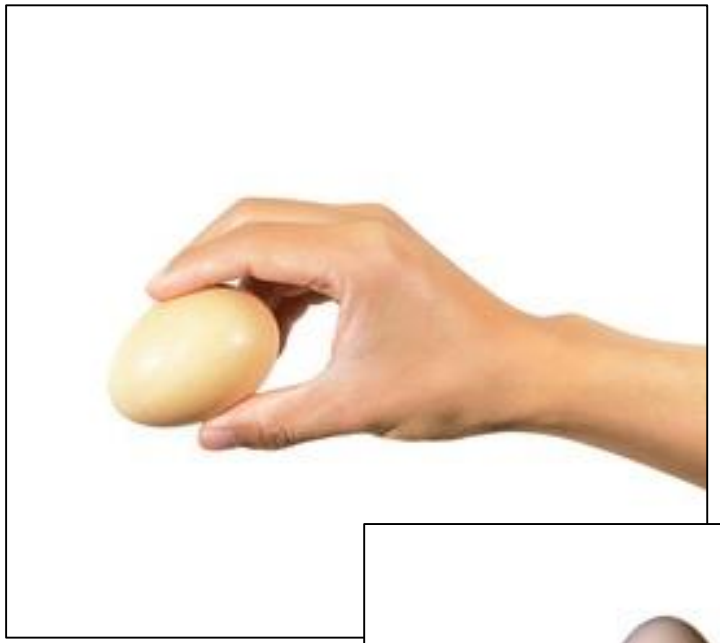
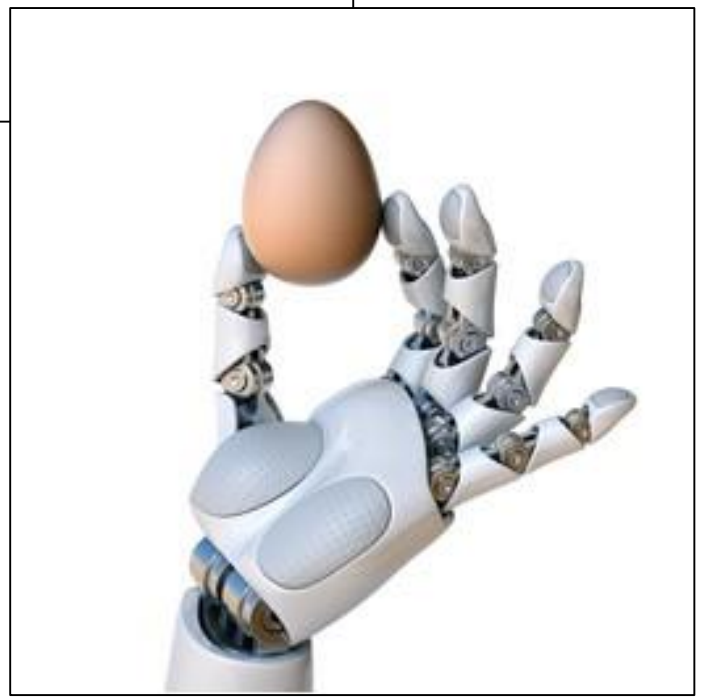
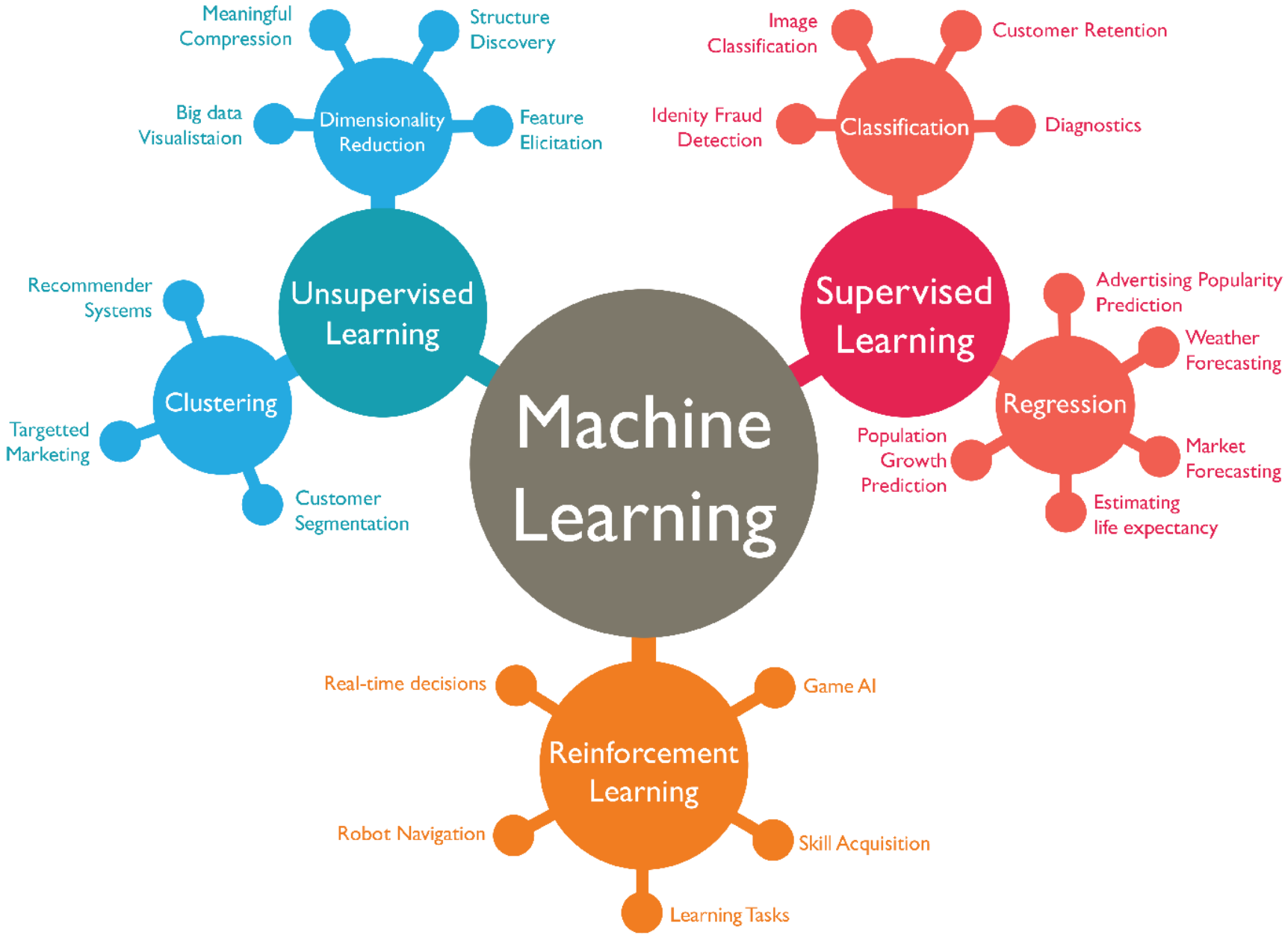
เทคนิคหรืออัลกอริทึ่ม
ถูกสร้างมาเพื่อแก้ปัญหาบางอย่างเสมอ



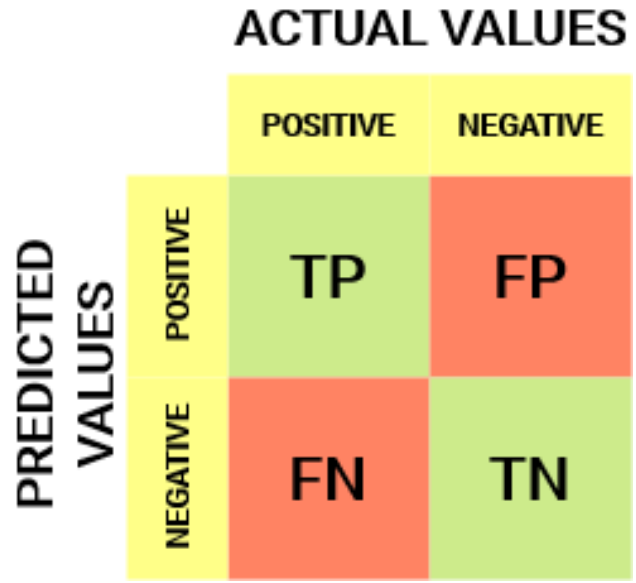


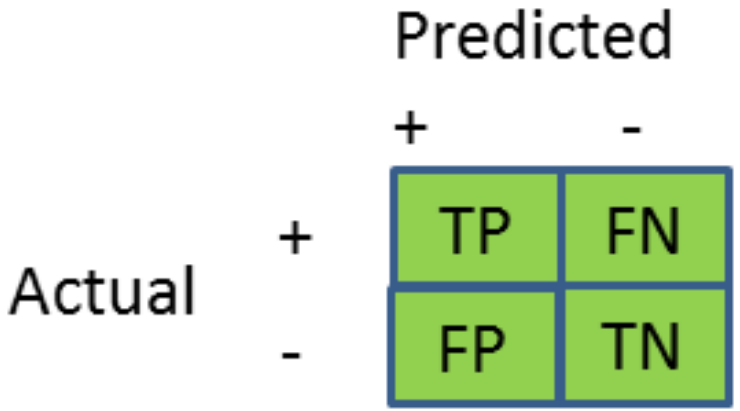

Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

Python +
Machine Learning

วัดประสิทธิภาพ Model จาก Confusion Matrix

True Positive (TP) คือ สิ่งทำนายว่า “จริง” และ มีค่าเป็น “ จริง ”
True Negative (TN) คือ สิ่งทำนายว่า “ไม่จริง” และ มีค่า “ ไม่จริง ”
False Positive (FP) คือ สิ่งทำนายว่า “จริง” แต่ มีค่าเป็น “ไม่จริง”
False Negative (FN) คือ สิ่งทำนายว่า “ไม่จริง” แต่ มีค่าเป็น “จริง”





Accuracy

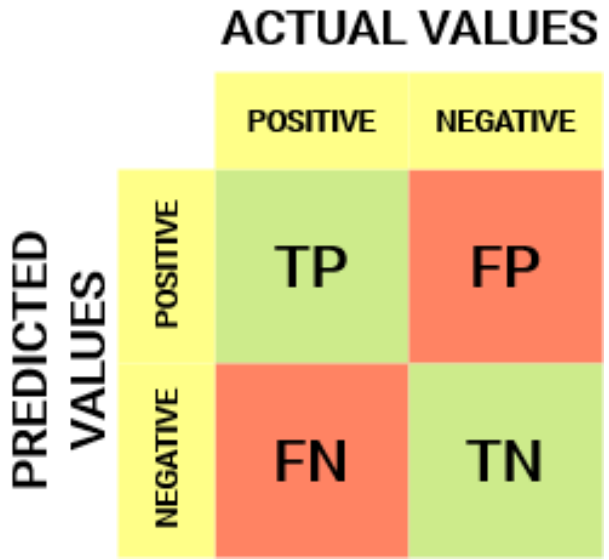
จำนวนครั้งที่ทายถูก หารด้วยจำนวนครั้งที่ทายทั้งหมด หมายความว่าทายแม่นแค่ไหนแบบรวม ๆ

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}}$$

Precision

จำนวนครั้งที่ทายว่า Positive แล้วถูก หารด้วยจำนวนครั้งที่ทายว่า Positive ทั้งหมด หมายความว่าใช้กระสุนเปลืองแค่ไหน

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



Recall

จำนวนครั้งที่ทายว่า Positive แล้วถูก หารด้วยจำนวน Positive ทั้งหมดในข้อมูล หมายความว่าเก็บหมดแค่ไหน

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

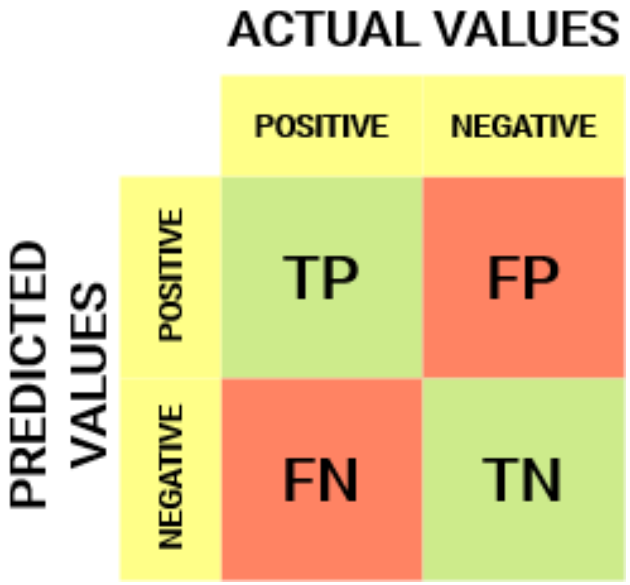
F1-Score

Harmonic mean ของ Precision และ Recall

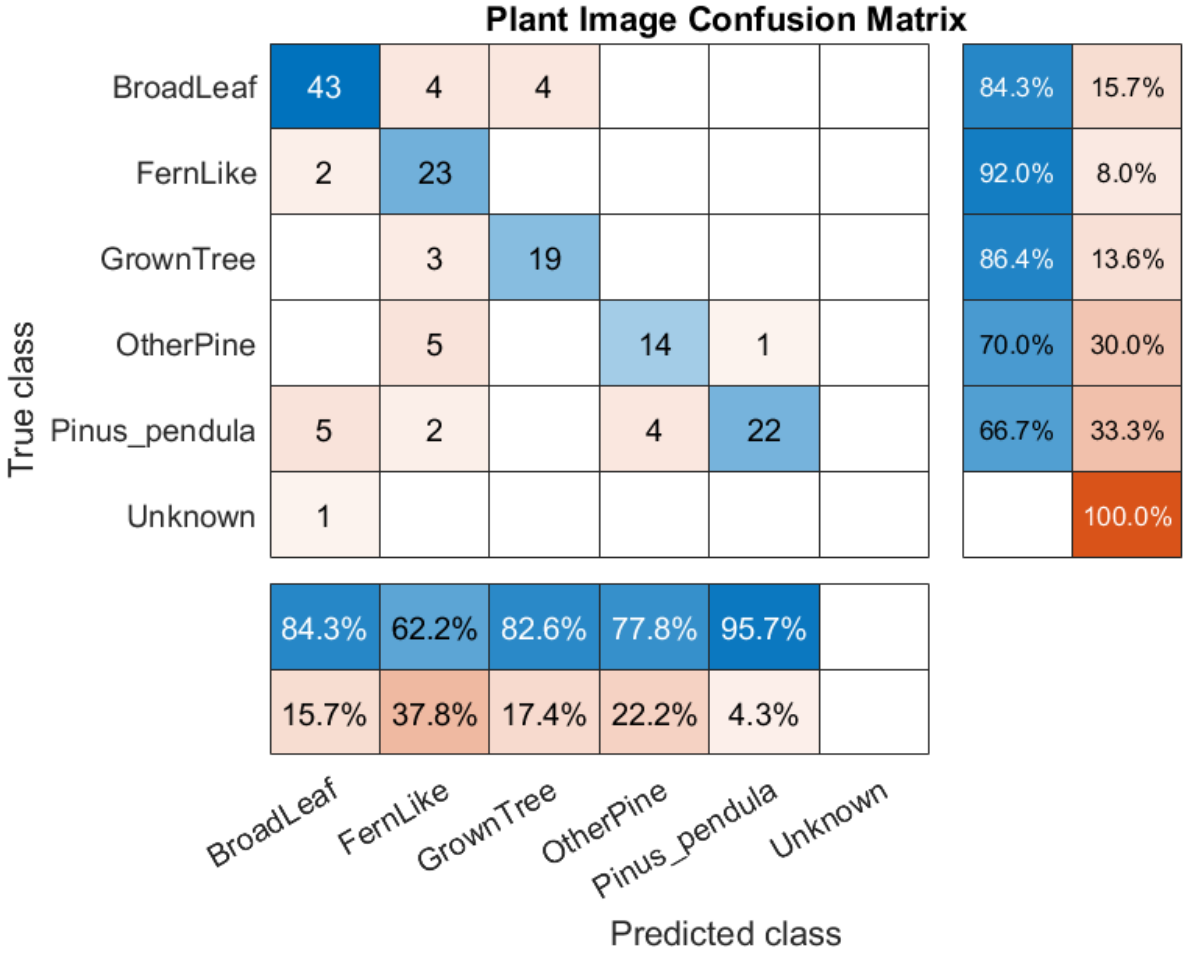
$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

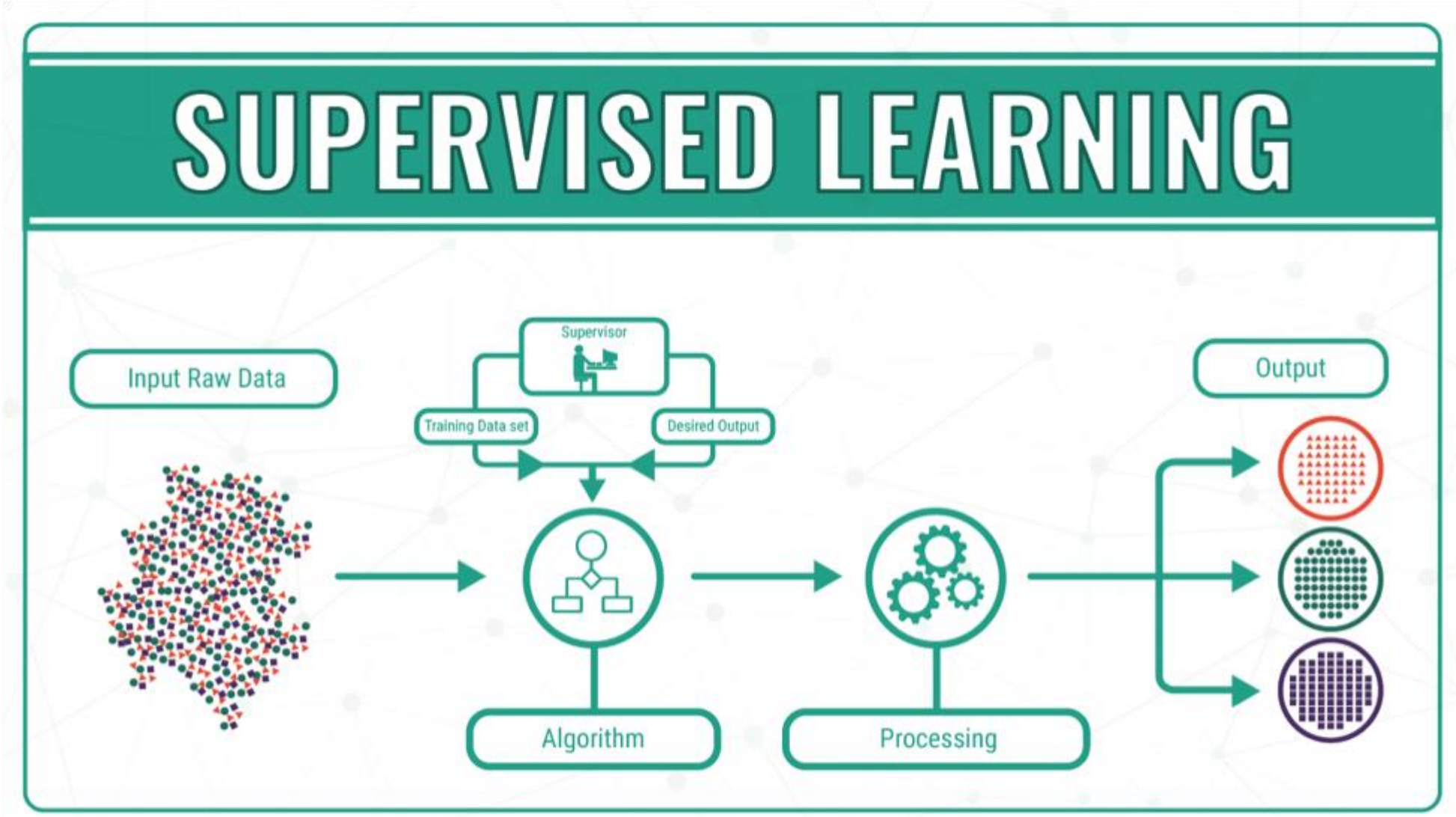
วัดประสิทธิภาพ Model จาก Confusion Matrix

2 กลุ่ม



หลายกลุ่ม




SUPERVISED LEARNING
Input Raw Data
Supervisor
Training Data set
Desired Output
Algorithm
Processing
Output


LINEAR
REGRESSION

แถว	x	y
1	1	6
2	2	7
3	3	8
4	4	9
5	5	?
6	6	?

คิดว่า

- ค่า **y** แถวที่ 5 = **?**
- ค่า **y** แถวที่ 6 = **?**

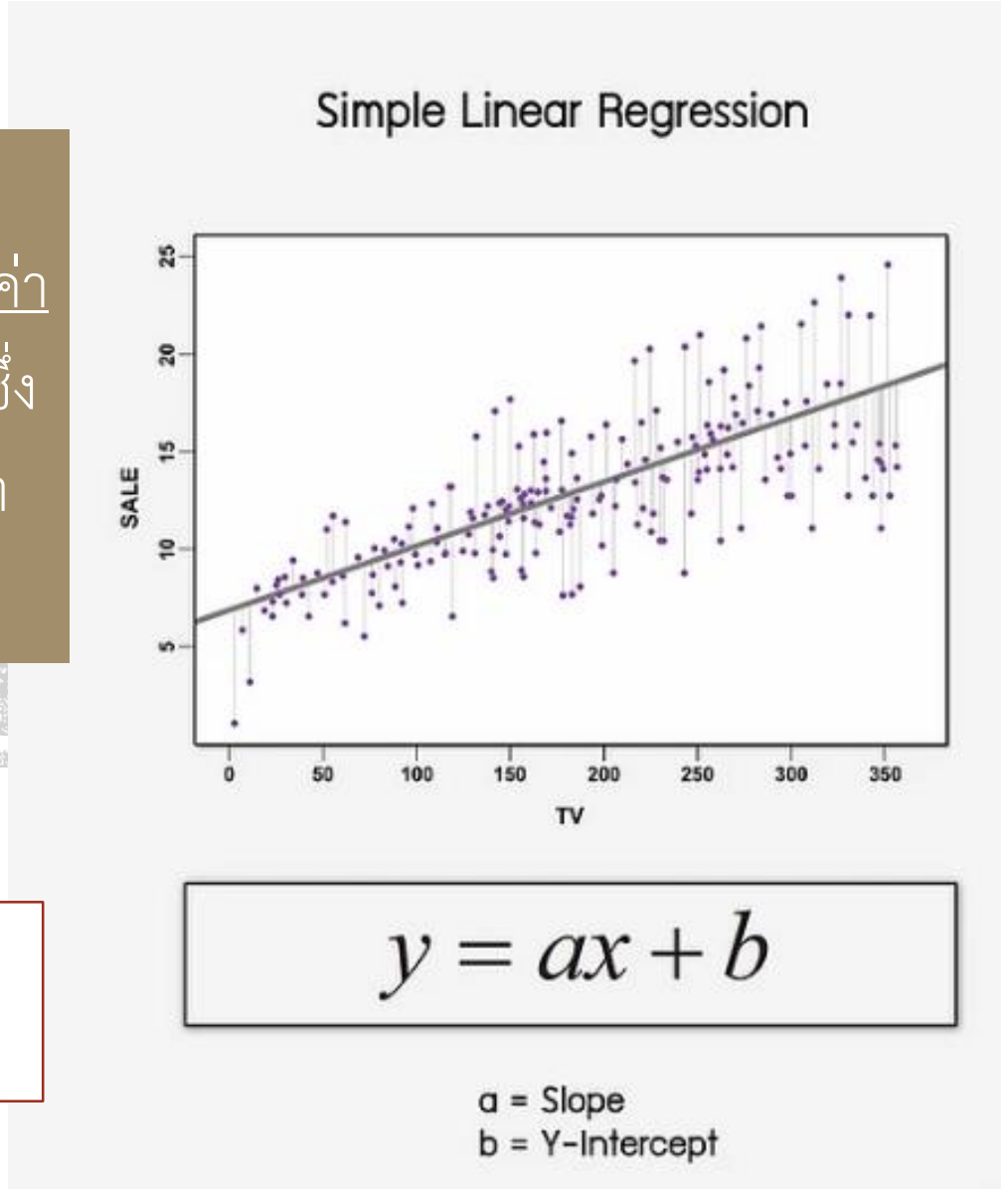
สรุปสูตร **y = x .....**

Linear Regression

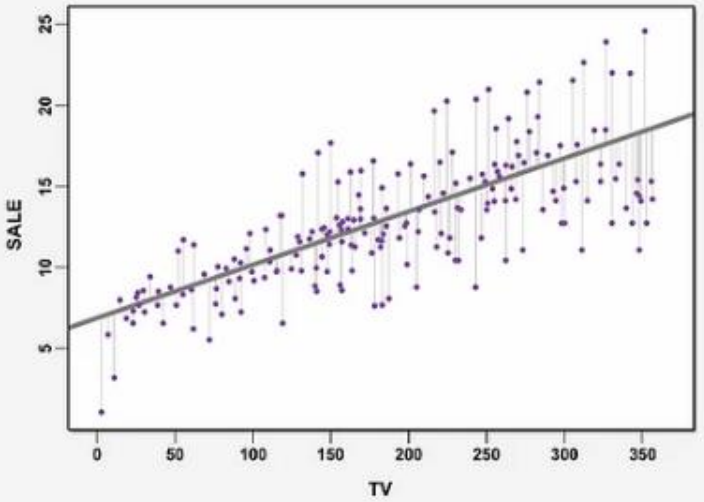
การวิเคราะห์การถดถอยเชิงเส้น เป็นการคำนวณหาความสัมพันธ์ระหว่างตัวแปร 2 ตัวแปร คือ ตัวแปรที่เราทราบค่า (Predictor :x) และตัวแปรที่เราไม่ทราบค่า (Response :y) ซึ่งเป็นความสัมพันธ์แบบเชิงเส้น (Linear) โดยการคำนวณจากค่า x และ y ที่มีความสัมพันธ์กัน

x= ตัวแปรอิสระ ตัวแปรที่ทราบค่า | ตัวประมาณการ (Predictor)

y= ตัวแปรตาม ตัวแปรที่เราไม่ทราบค่า | ตัวตอบสนอง (Response)

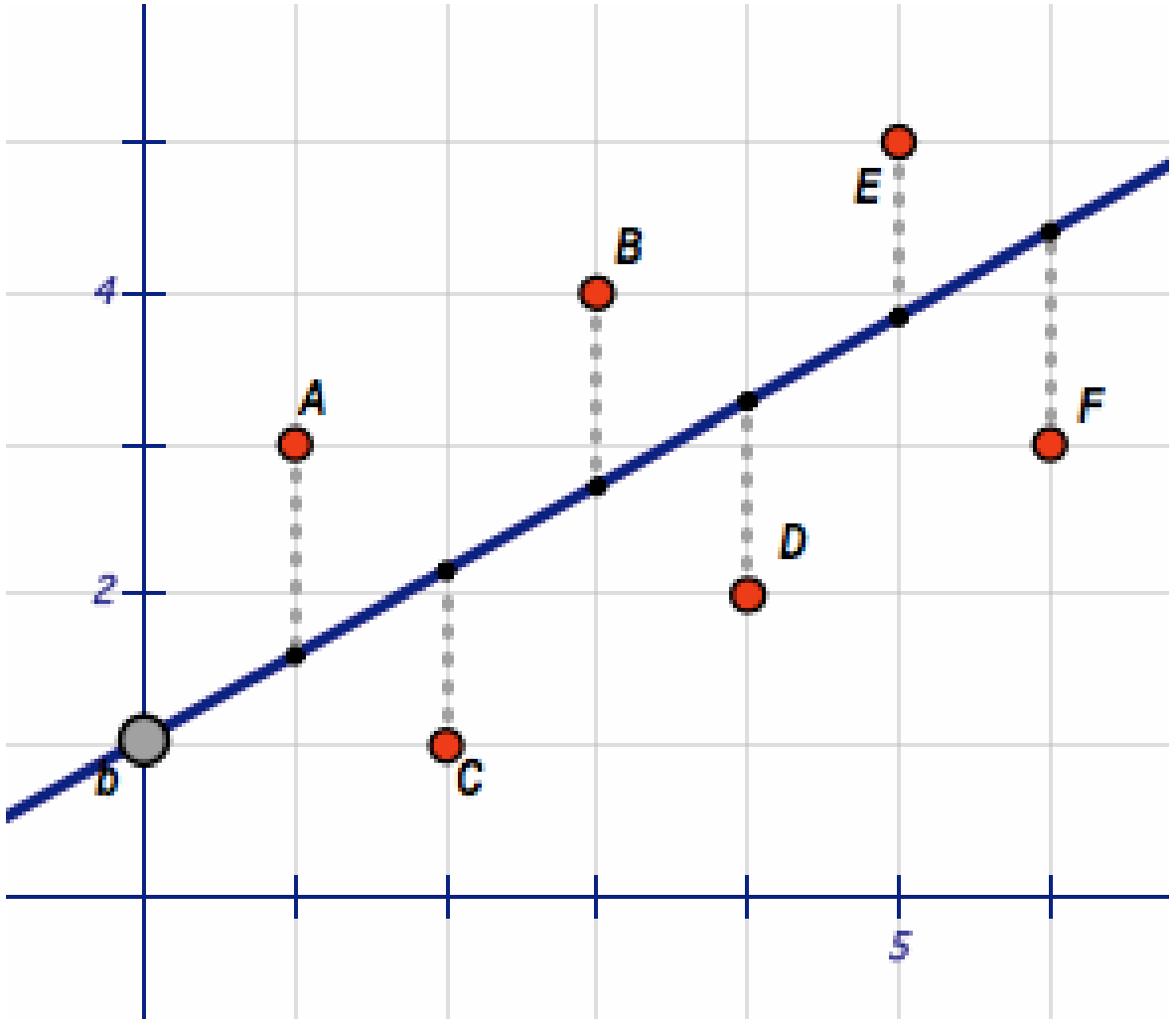


Simple Linear Regression



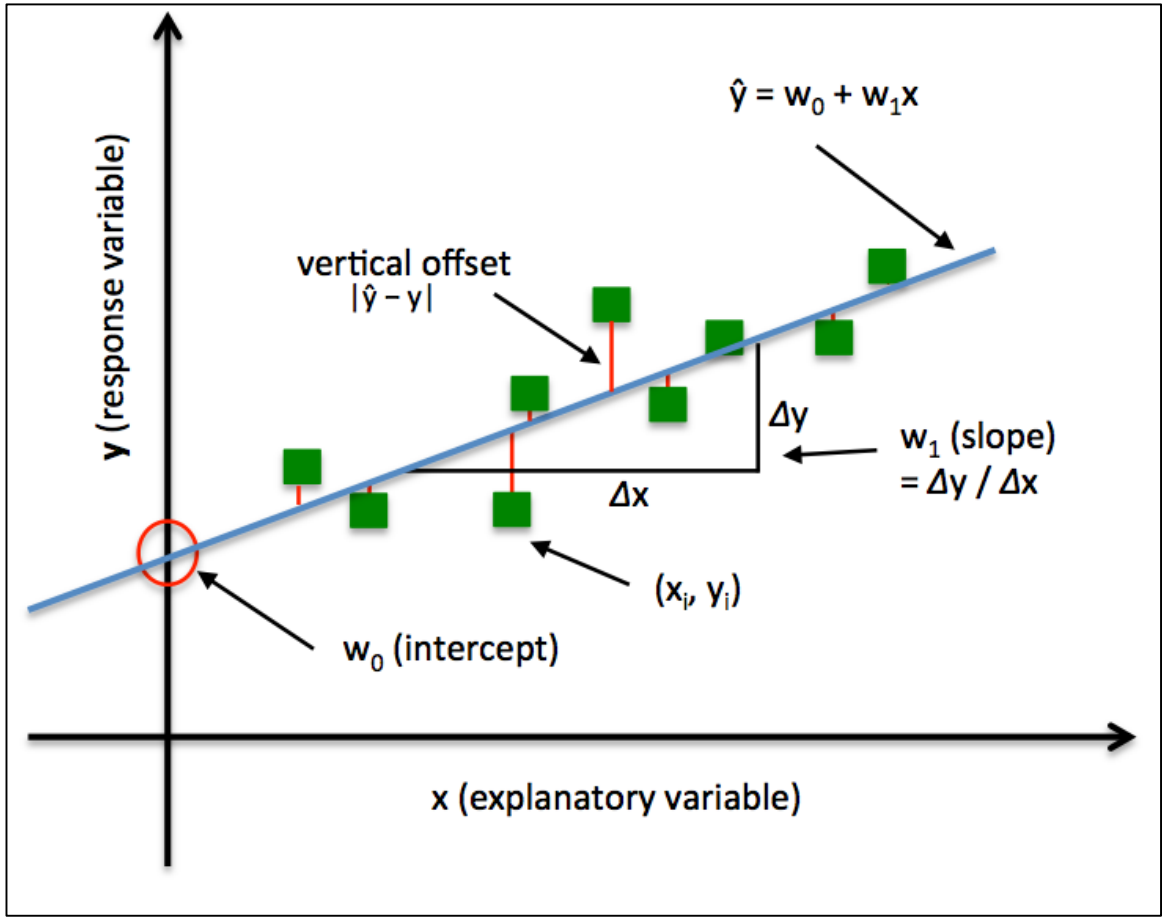
$$y = ax + b$$

a = Slope
b = Y-Intercept



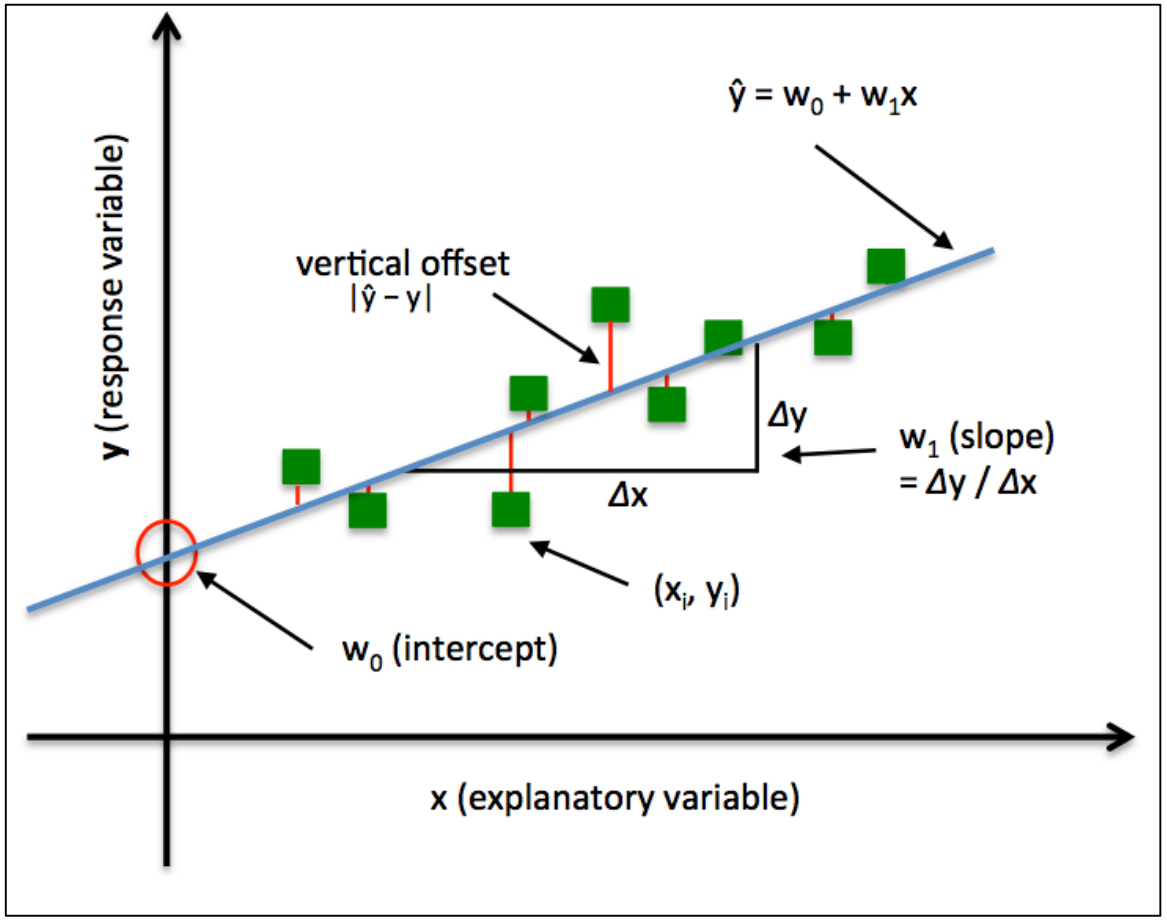
$$Y = \beta_0 + \beta_1 X + \varepsilon$$

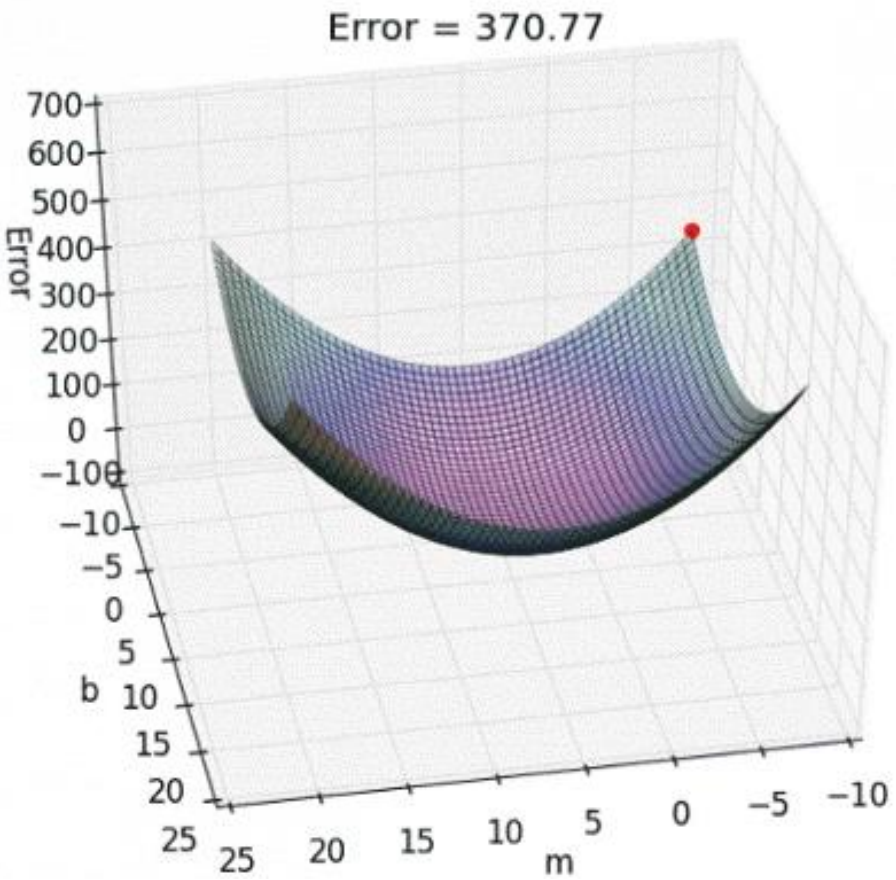
- ค่า Y คือค่าที่เราสนใจเป็นหลัก (dependent variable, response หรือ outcome)
- ค่า X คือตัวแปรที่เราต้องการนำมาอธิบายผลของ Y (independent variable, predictor หรือ explanatory)
- ค่าคงที่ β_0 (w_0) เรียกว่า coefficient (เมื่อ X มีค่าเป็น 0 : Y (y-intercept))
- ค่าน้ำหนัก β_1 (w_1) เป็น parameter ของค่า X
  - บอกถึงน้ำหนักของ X ที่ส่งผลต่อ Y
  - ยิ่ง β_1 มาก แสดงว่าค่า X สามารถอธิบายผลของ Y ได้มาก
  - โดย β_1 คือความชัน หรือ slope
- ค่า error (ε) ที่มีการกระจายตัวแบบ normal distribution
  - มีค่าเฉลี่ยอยู่ที่ 0 (mean-zero random error)

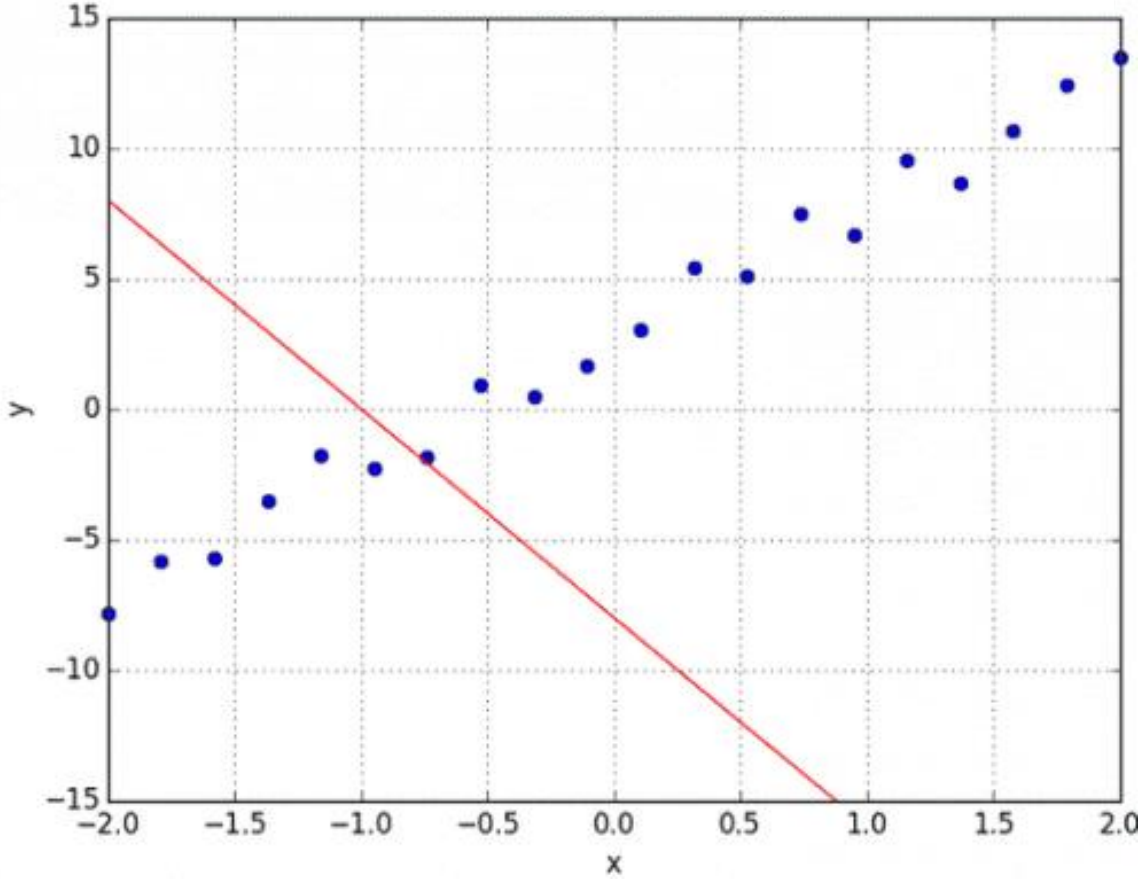


least squares method

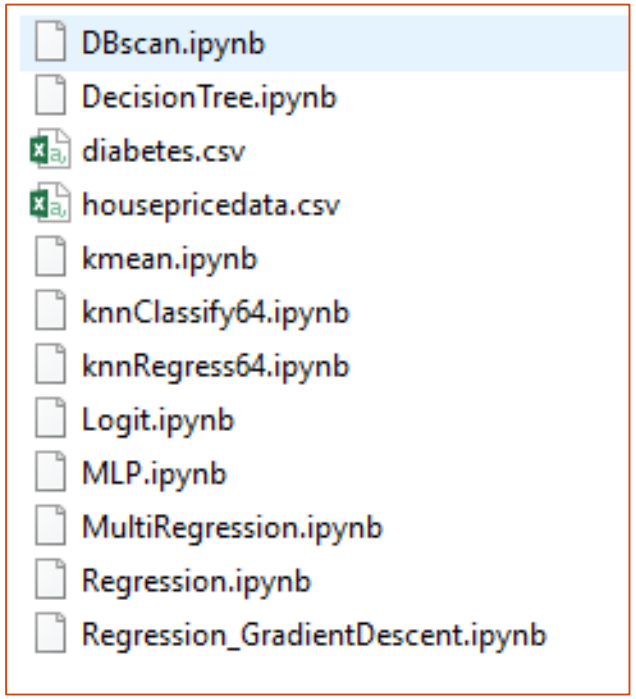
- ค่า **residual** ซึ่งก็คือค่าความคลาดเคลื่อน (error) ระหว่าง response y ของชุดข้อมูล (จุด) กับ response y_hat
- วิธีการคำนวณ error ของทุก ๆ จุด ที่นิยมใช้กันคือ residual sum of squares (RSS)
- จะมีเส้น regression อย่างน้อย 1 เส้นที่ให้ค่า RSS ต่ำที่สุดเสมอ ซึ่งเส้น regression ที่เกิดจาก least squares method จะเรียกว่าเส้น least squares





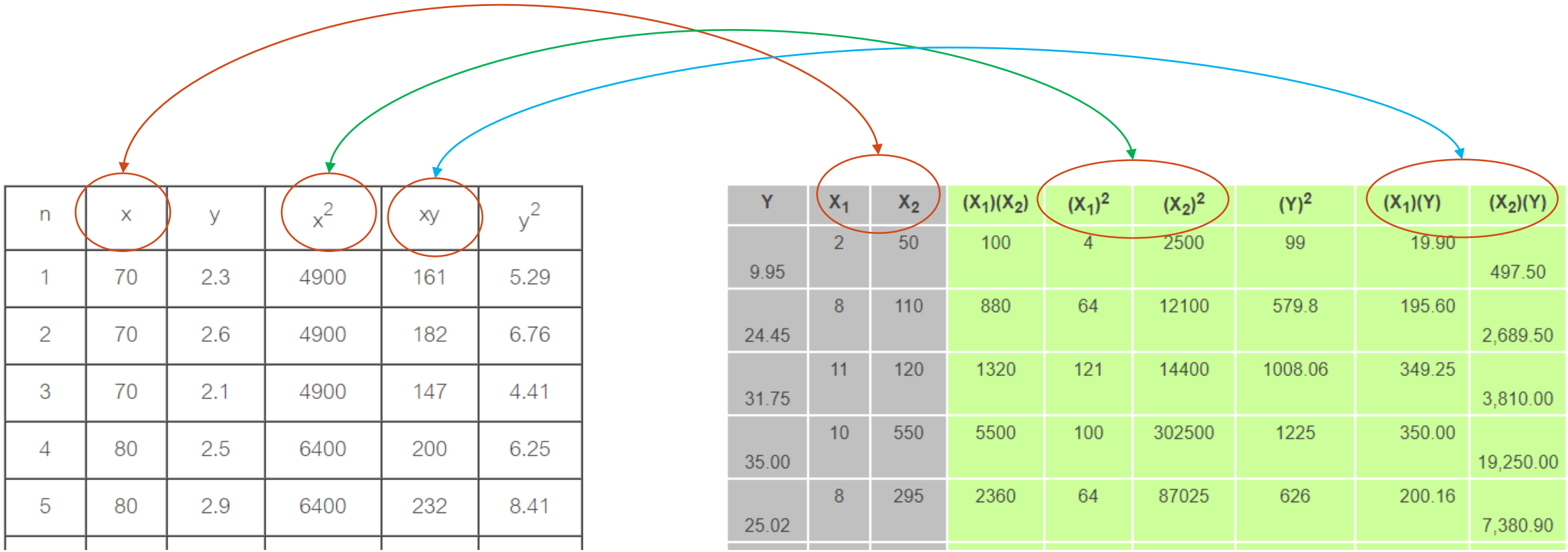


Plot ค่าผลรวม error สำหรับทุก ๆ เส้นตรง



Regression.ipynb

Multiple Linear Regression



n	x	y	x^2	xy	y^2
1	70	2.3	4900	161	5.29
2	70	2.6	4900	182	6.76
3	70	2.1	4900	147	4.41
4	80	2.5	6400	200	6.25
5	80	2.9	6400	232	8.41

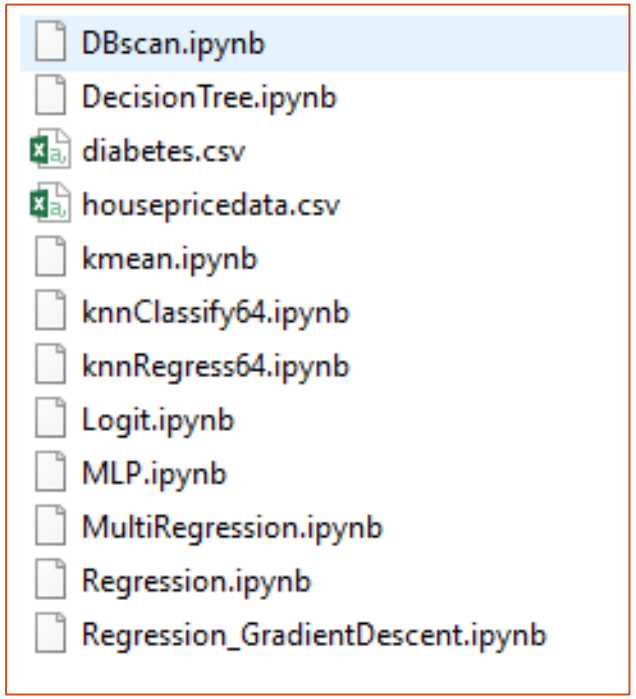
Y	X_1	X_2	$(X_1)(X_2)$	$(X_1)^2$	$(X_2)^2$	$(Y)^2$	$(X_1)(Y)$	$(X_2)(Y)$
9.95	2	50	100	4	2500	99	19.90	497.50
24.45	8	110	880	64	12100	579.8	195.60	2,689.50
31.75	11	120	1320	121	14400	1008.06	349.25	3,810.00
35.00	10	550	5500	100	302500	1225	350.00	19,250.00
25.02	8	295	2360	64	87025	626	200.16	7,380.90

Linear Regression

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

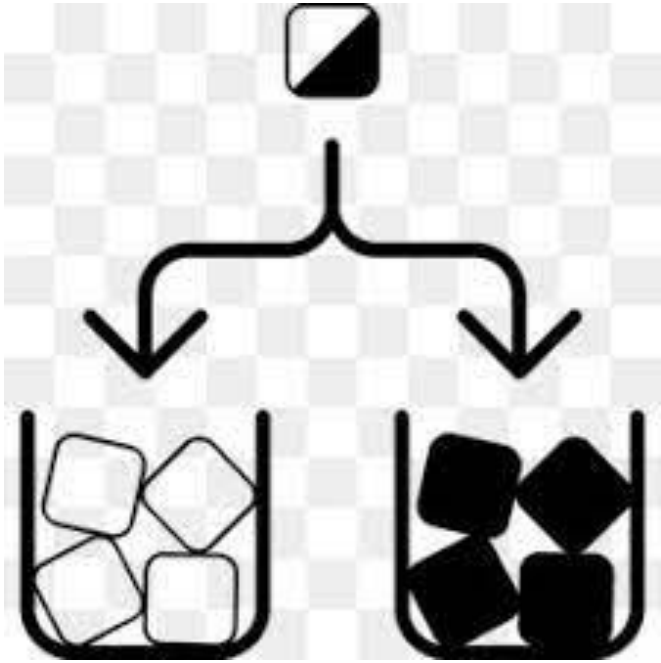
Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \varepsilon$$

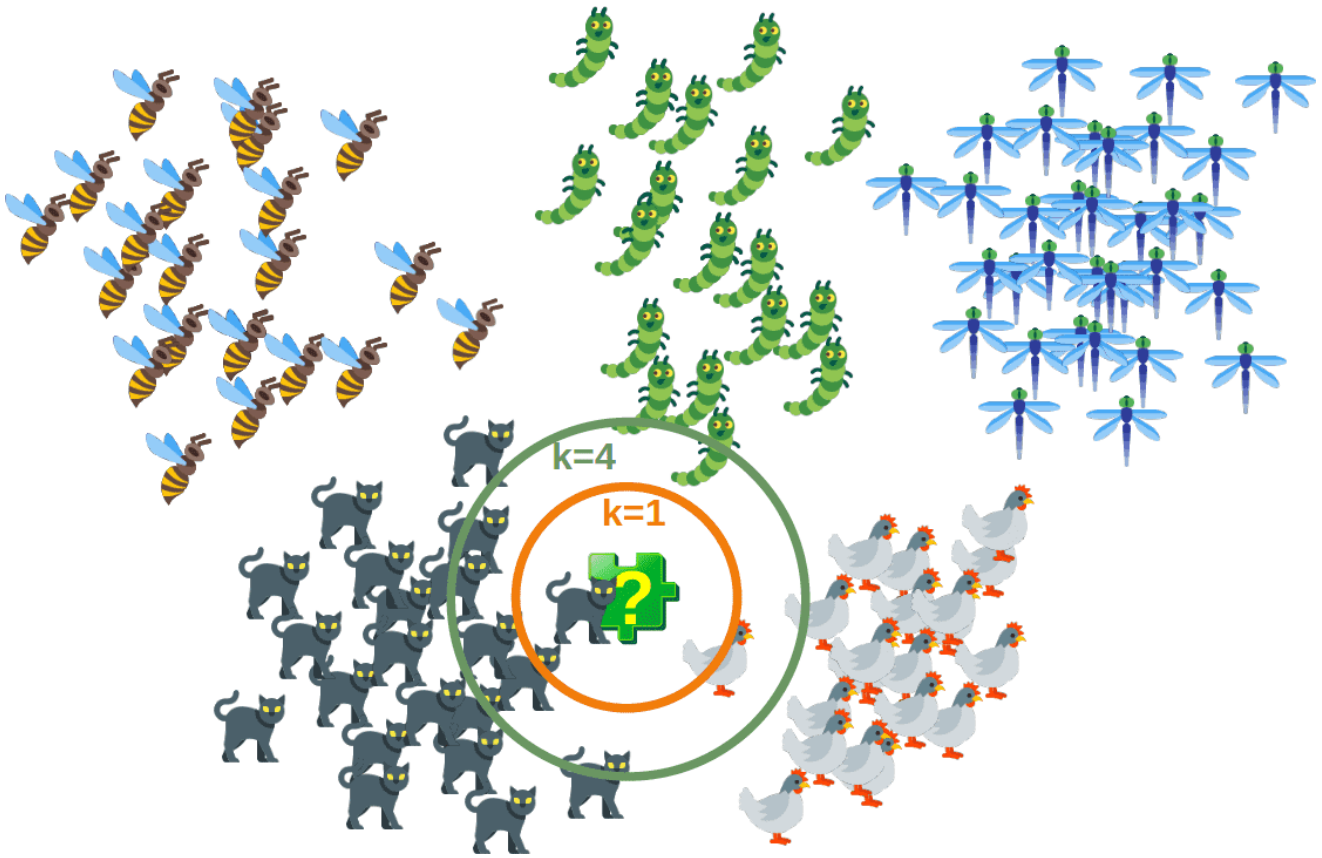


MultiRegression.ipynb

Classification

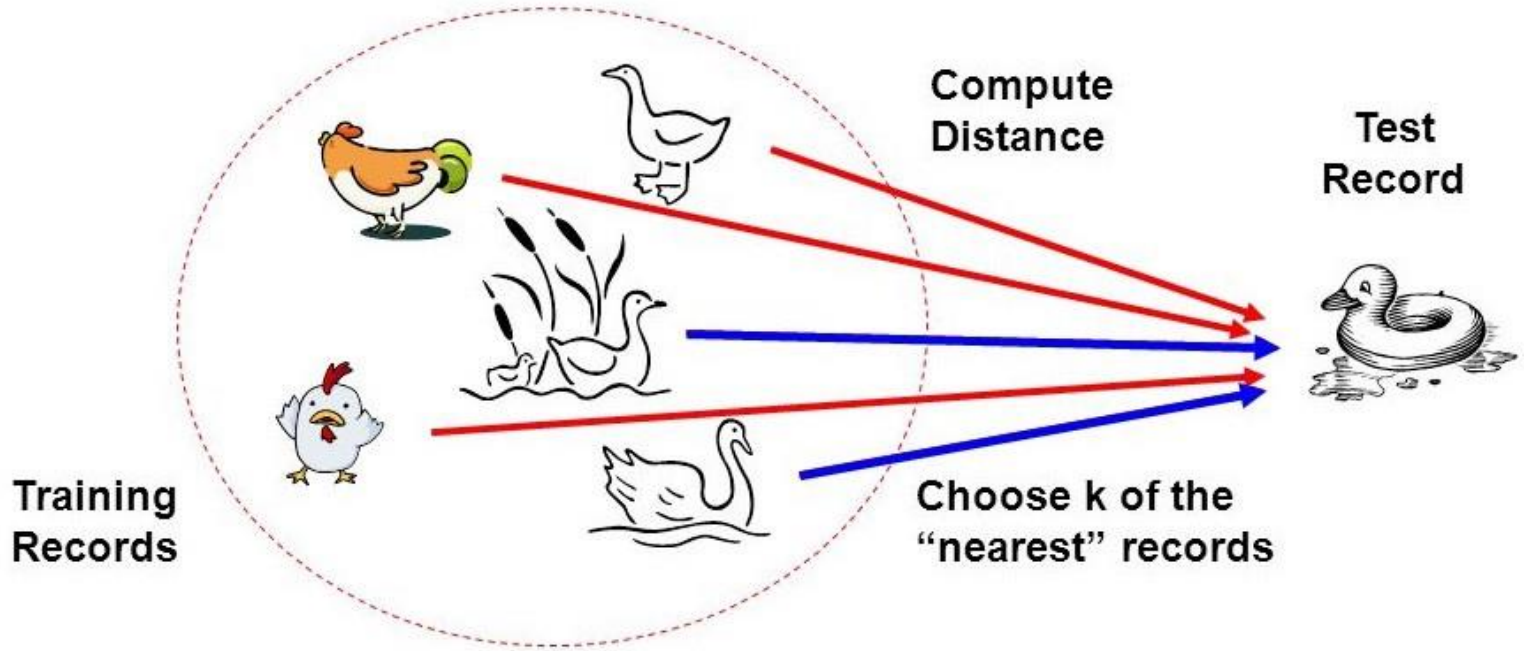


รูปในวงกลมสีสัม (แมว) คืออะไร คนรู้ แต่หุ่นยนต์ไม่รู้



K-Nearest Neighbors (KNN)

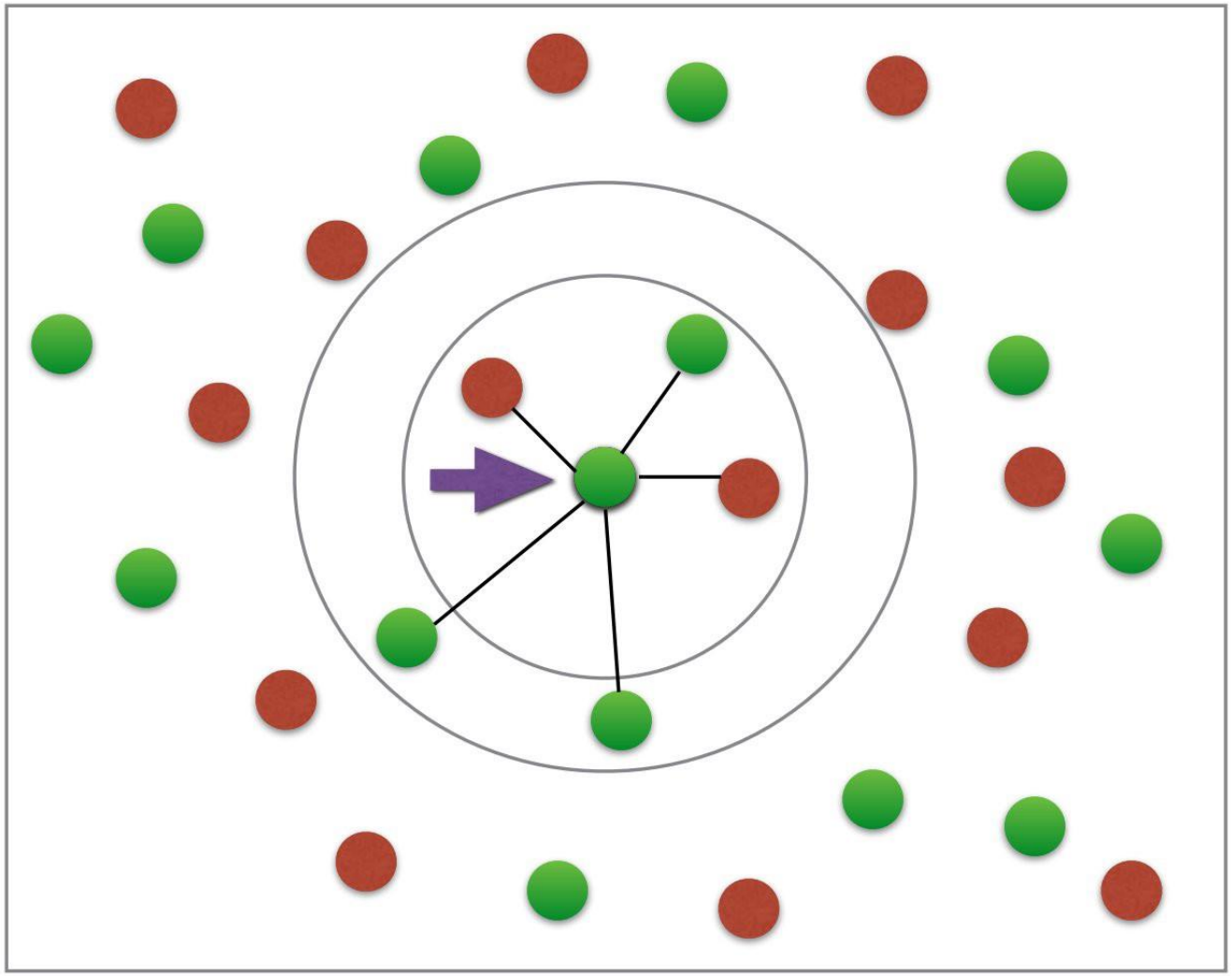
- เป็น ML algorithm อีกหนึ่งตัวที่ใช้กันมาก เพราะว่า**เข้าใจง่าย** แก้ปัญหา (พยากรณ์) ได้ทั้ง regression และ classification
- ใช้หลักการ**เปรียบเทียบ หรือ วัด** ข้อมูลที่สนใจกับข้อมูลอื่นว่ามีความคล้ายคลึงมากน้อยเพียงใด (Distance function) หากข้อมูลที่กำลังสนใจนั้นอยู่ใกล้ข้อมูลใดมากที่สุด อัลกอริทึ่มจะเลือก (Vote) ให้ค่านั้นเป็นคำตอบ



ขั้นตอนโดยสรุป ดังนี้

1. กำหนดขนาดของ **K** (ควรกำหนดให้เป็นเลขคี่)
2. คำนวณระยะห่าง (Distance) ของข้อมูลที่ต้องการพิจารณากับกลุ่มข้อมูลตัวอย่าง
3. จัดเรียงลำดับของระยะห่าง และเลือกพิจารณาชุดข้อมูลที่ใกล้จุดที่ต้องการพิจารณาตามจำนวน K ที่กาหนดไว้
4. พิจารณาข้อมูลจำนวน k ชุด และสังเกตว่ากลุ่ม (class) ไหนที่ใกล้จุดที่พิจารณาเป็นจำนวนมากที่สุด (Vote)
5. กำหนด class ให้กับจุดที่พิจารณา (class) ที่ใกล้จุดพิจารณามากที่สุด

ฟังก์ชันระยะทาง (Distance Function)



Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

ฟังก์ชันระยะทาง (Distance Function) เป็นการคำนวณค่าระยะห่างระหว่างสองเรคคอร์ด เพื่อที่จะมาวัดความคล้ายคลึงกันของข้อมูล

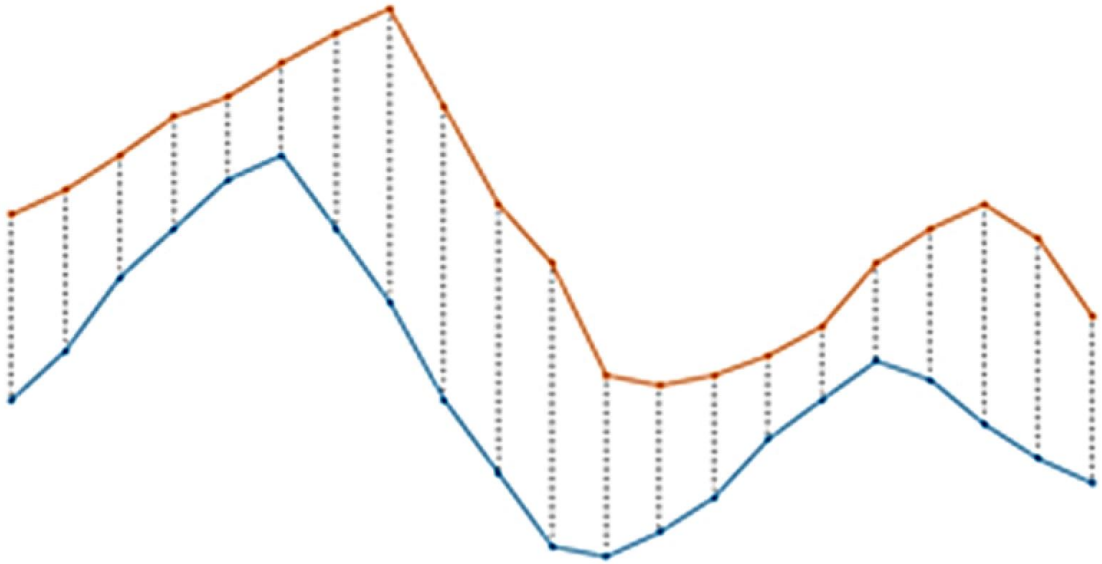
- การวัดระยะแบบแมนฮัตตัน (Manhattan distance) เป็นการนำค่าที่คำนวณได้ในหนึ่งเรคคอร์ด (Record) มารวมกัน
- ระยะทางแบบยุคลิด (Euclidean distance) เป็นการหารากที่สอง (Square Root) ในแต่ละตัวแปร (attribute) แล้วนำมารวมกัน แล้วนำค่าที่คำนวณได้ในหนึ่งเรคคอร์ด (Record) มารวมกัน
- DTW

Distance functions

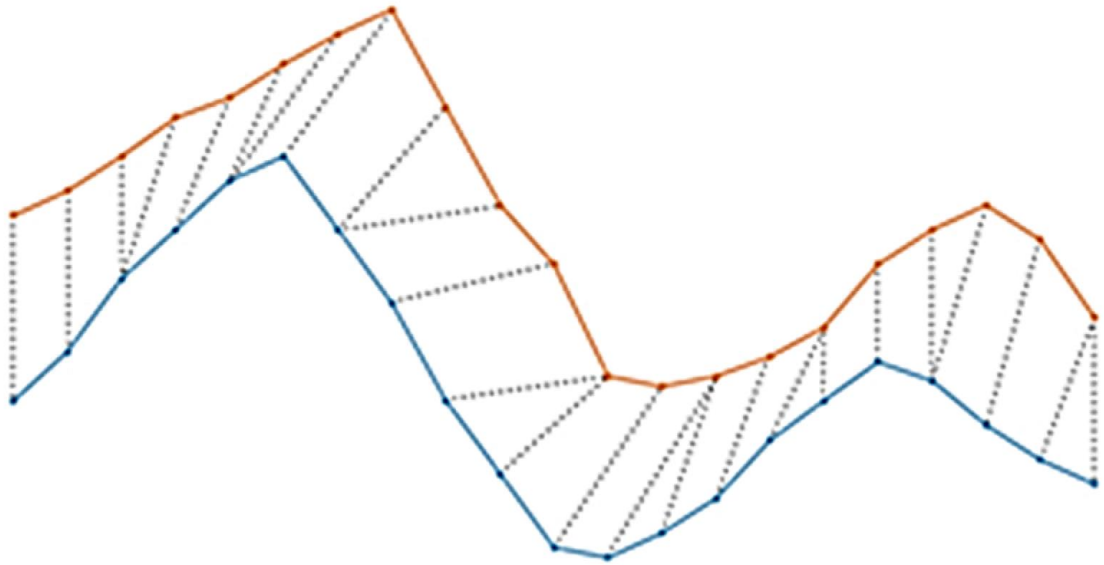
Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

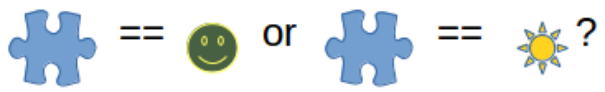


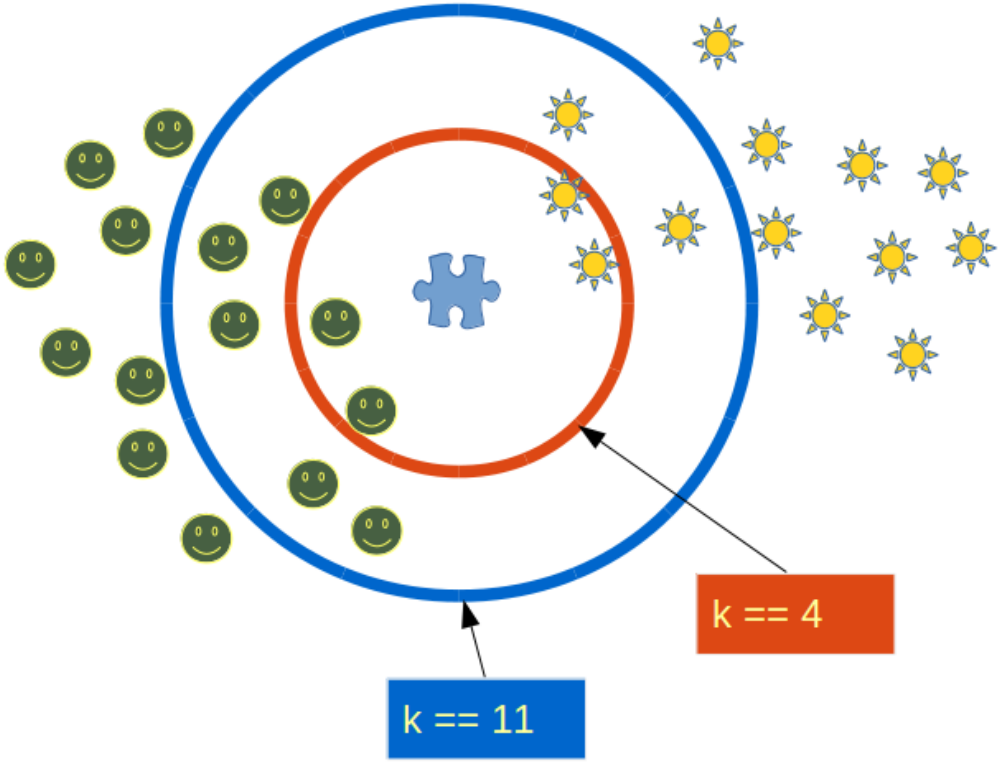
(a) Euclidean distance

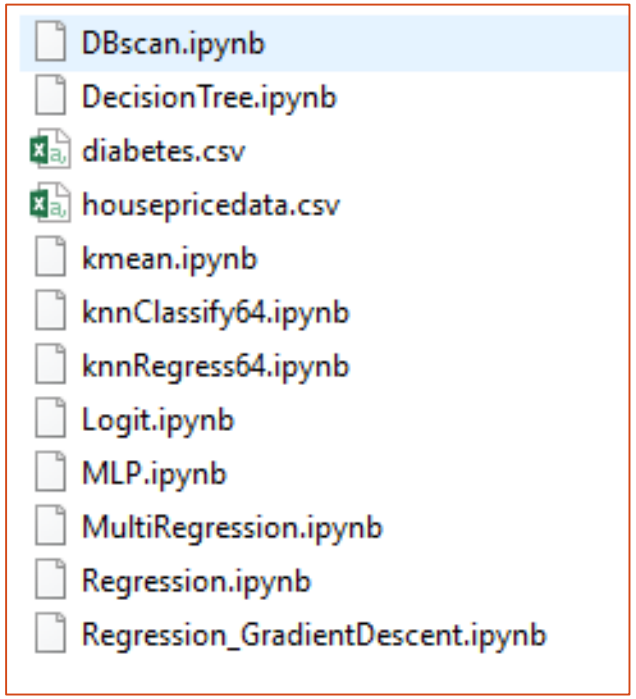


(b) Dynamic time warping

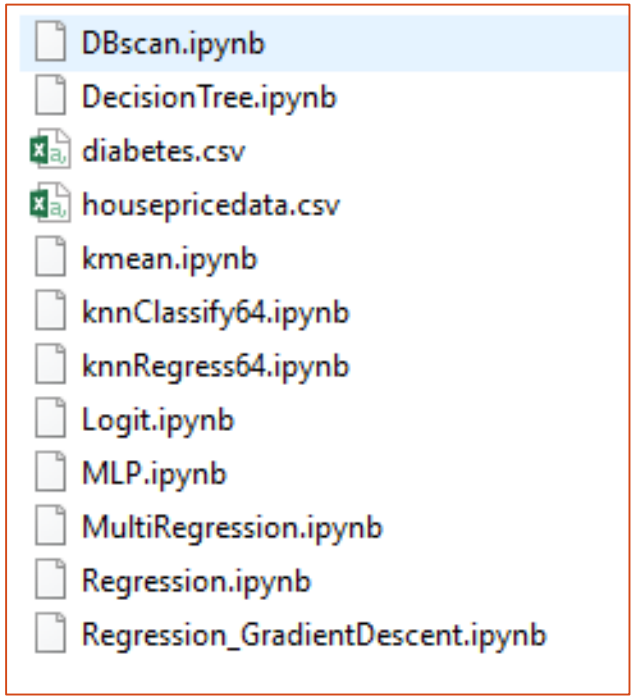
- Nonlinear Learning Algorithm
- Classification / Regression
- Lazy algorithm
- Distance function : Euclidean / DTW distance







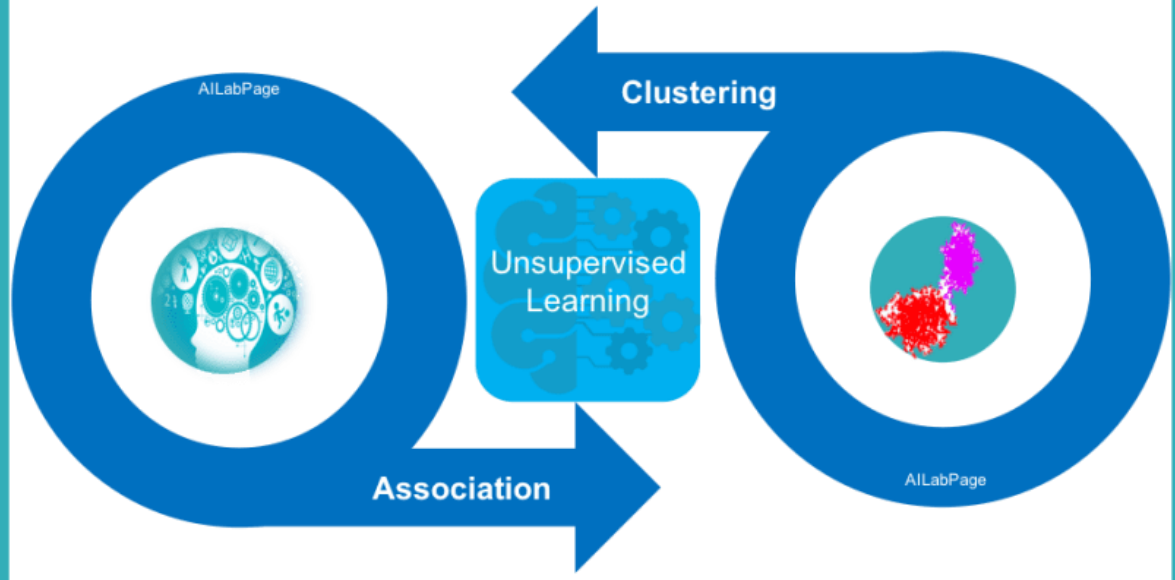
knnClassify64.ipynb



DecisionTree.ipynb

Unsupervised Learning

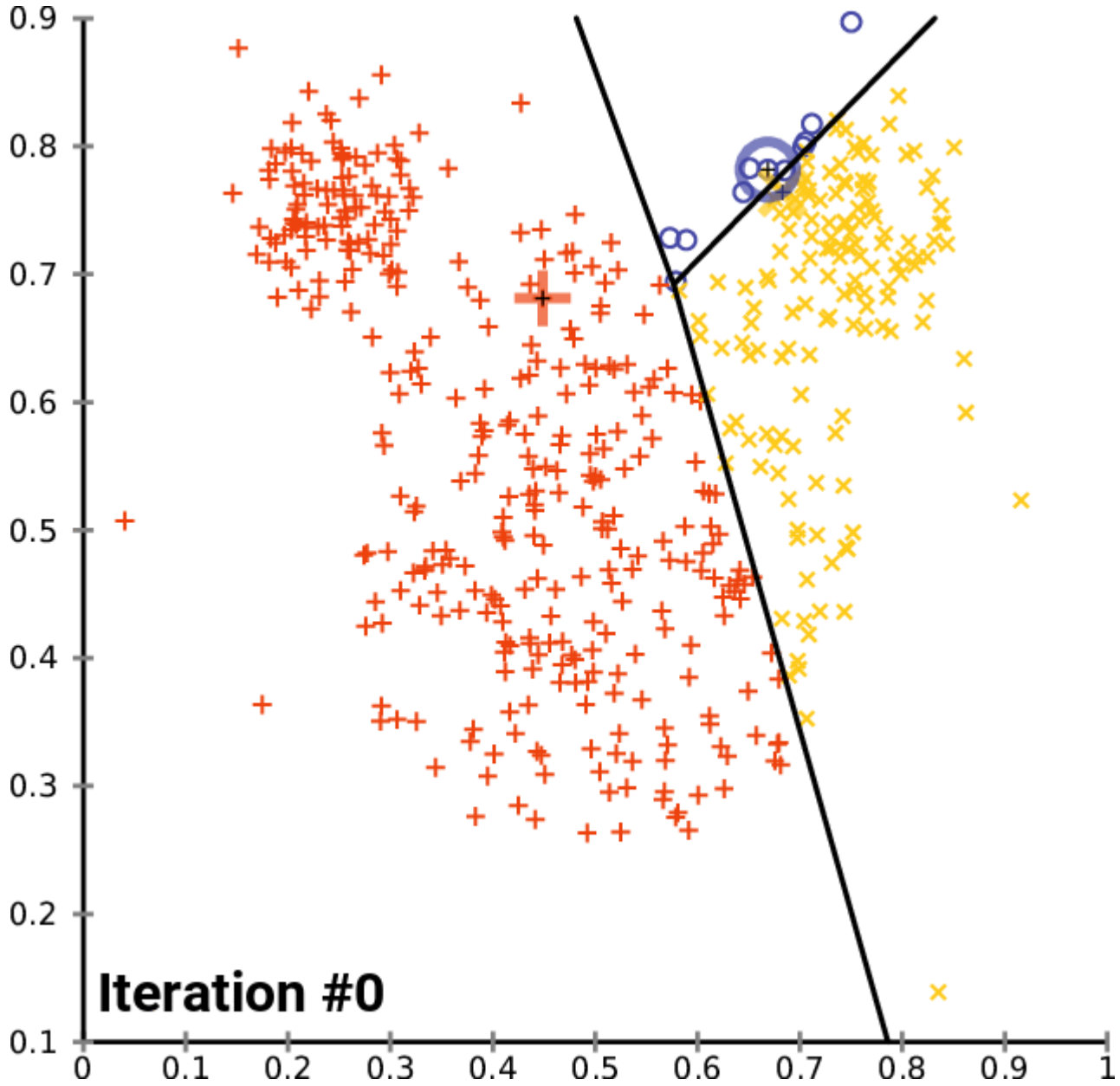
The association rule mining is used in unsupervised scenarios to discover interesting patterns.



Grouping of data automatically split the dataset into groups according to similarity. AILabPage

K-means คือ วิธีการหนึ่งใน Data mining อยู่ในกลุ่มของ Unsupervised Learning หรือแปลตรงๆคือการเรียนรู้แบบไม่ต้องสอน

K-means เป็นอัลกอริทึมเทคนิคการเรียนรู้โดยไม่มีผู้สอน**ที่ง่ายที่สุด** เพราะเป็น การแก้ปัญหาการจัดกลุ่มที่รู้จักกันทั่วไป โดยอัลกอริทึม K-Means จะตัดแบ่ง (Partition) วัตถุออกเป็น K กลุ่ม และแทนค่าแต่ละกลุ่มด้วยค่าเฉลี่ยของกลุ่ม ซึ่งใช้เป็นจุดศูนย์กลาง (centroid) ของกลุ่มในการวัดระยะห่างของข้อมูลในกลุ่มเดียวกัน

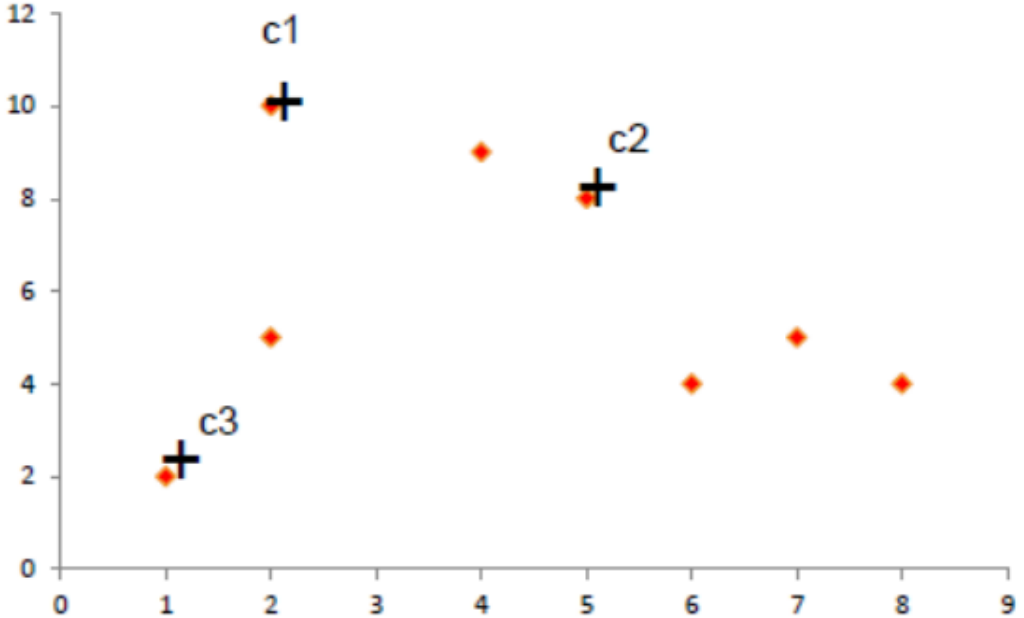

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Iteration #0

วิธีการของ K-means มี 4 ขั้นตอนดังนี้

1. กำหนดจำนวนกลุ่มขึ้นมาก่อน เช่น 2 กลุ่มหรือหมายความว่าค่า K=2 (กำหนดเป็น C1 และ C2) และสุ่มตำแหน่งแกน x,y ให้กับ C1 และ C2 จะได้ C1(x1,y1) และ C2(x2,y2)
2. ดูตำแหน่งของสมาชิกแต่ละสมาชิกว่าอยู่ใกล้ใครมากกว่ากันก็ให้คนนั้นเป็นสมาชิกของ C นั้น จากตรงนี้จะรู้แล้วว่าสมาชิกแต่ละคนอยู่ในกลุ่มใดระหว่าง C1 และ C2
3. ปรับ x,y ของ C1 และ C2 ใหม่ให้อยู่ตรงกลางของกลุ่ม
4. ทำ ตามข้อ 2 และข้อ 3 อีกครั้งจนกว่า C1 และ C2 ตำแหน่งไม่เปลียน

ตัวอย่างการทำ K-MEAN CLUSTERING

- สุ่มค่าเริ่มต้น จำนวน k ค่า เรียกว่า cluster centers (centroid)
- สมมติ k =3 แสดงว่า c1,c2และc3 เป็น centroidที่เราสุ่มขึ้นมา c1(2, 10),c2(5,8) และ c3(1,2)



Clustering : K-mean โดย ผศ.วิภาวรรณ บัวทอง

ตัวอย่างการทำ K-MEAN CLUSTERING

ขั้นตอนที่ 1

- หาความห่างกันระหว่างข้อมูล 2 ข้อมูล คือ หาความห่างจากข้อมูล A=(x1,y1)และ centroid =(x2,y2) โดยใช้สูตร **Euclidean** ดังนี้

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

- และนำไปใส่ในตาราง

		c1(2, 10)	c2 (5, 8)	c3(1, 2)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(2, 10)				
A2	(2, 5)				
A3	(8, 4)				
A4	(5, 8)				
A5	(7, 5)				
A6	(6, 4)				
A7	(1, 2)				
A8	(4, 9)				

ตัวอย่างการทำ K-MEAN CLUSTERING

ขั้นตอนที่ 2

หาระยะห่างระหว่างข้อมูล กับจุดศูนย์กลาง (ตัวอย่างบางชุดข้อมูล)

point mean1
x1, y1 *x2, y2*
(2, 10) (2, 10)

$$distance(point, mean1) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
$$= \sqrt{(2-2)^2 + (10-10)^2}$$
$$= 0$$

point mean2
x1, y1 *x2, y2*
(2, 10) (5, 8)

$$distance(point, mean2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
$$= \sqrt{(2-5)^2 + (10-8)^2}$$
$$= 3.61$$

point mean3
x1, y1 *x2, y2*
(2, 10) (1, 2)

$$distance(point, mean3) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
$$= \sqrt{(2-1)^2 + (10-2)^2}$$
$$= 8.06$$

ตัวอย่างการทำ K-MEAN CLUSTERING

เมื่อใส่ข้อมูลในตารางจะได้การจัดกลุ่มข้อมูลดังต่อไปนี้ และนำไปสร้างกลุ่มใหม่

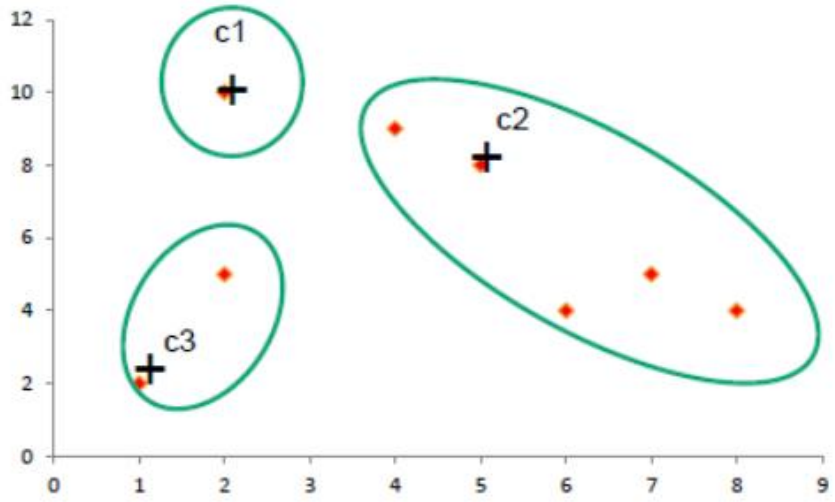
	Point	c1(2,10)	c2(5,8)	c3(1,2)	Cluster
A1	(2,10)	0.00	3.61	8.06	1
A2	(2,5)	5.00	4.24	3.16	3
A3	(8,4)	8.49	5.00	7.29	2
A4	(5,8)	3.61	0.00	7.21	2
A5	(7,5)	7.07	3.60	6.71	2
A6	(6,4)	7.21	4.12	5.39	2
A7	(1,2)	8.06	7.21	0.00	3
A8	(4,9)	2.24	1.41	7.62	2

ตัวอย่างการทำ K-MEAN CLUSTERING

- นำมาสร้างกลุ่มใหม่

Custer 1	Custer 2	Custer 3
A1 (2,10)	A3(8,4)	A2(2,5)
	A4(5,8)	A8(1,2)
	A5(7,5)	
	A6(6,4)	

- จะได้การจัดกลุ่มใหม่ดังภาพ



Clustering : K-mean โดย ผศ.วิภาวรรณ บัวทอง

ตัวอย่างการทำ K-MEAN CLUSTERING

ขั้นตอนที่ 3

หาค่าเฉลี่ยแต่ละกลุ่ม ให้เป็น ค่าจุดศูนย์กลางใหม่

Custer 1	Custer 2	Custer 3
A1 (2,10)	A3(8,4)	A2(2,5)
	A4(5,8)	A8(1,2)
	A5(7,5)	
	A6(6,4)	

สำหรับ Cluster 1 มีจุดเดียวคือ A1(2, 10) แสดงว่า C1(2,10) ยังคงเดิม

สำหรับ Cluster 2 มี 5 จุดอยู่กลุ่มเดียวกัน เพราะฉะนั้นหา C2 ใหม่ ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = C2(6,6)

สำหรับ Cluster 3 มี 2 จุดอยู่กลุ่มเดียวกัน ((2+1)/2, (5+2)/2) = C3(1.5,3.5)

ตัวอย่างการทำ K-MEAN CLUSTERING

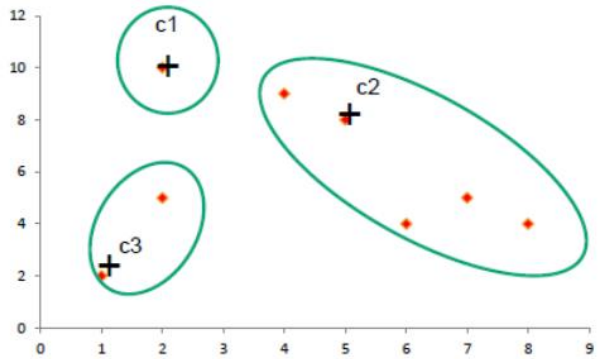
รอบที่ 2 ทำตามวิธีที่ 1-4 จะได้ผลลัพธ์ดังนี้

	Point	c1(2,10)	c2(6,6)	c3(1.5,3.5)	Cluster
A1	(2,10)	0.00	5.66	6.52	1
A2	(2,5)	5.00	4.12	1.58	3
A3	(8,4)	8.49	2.83	6.52	2
A4	(5,8)	3.60	2.24	5.70	2
A5	(7,5)	7.07	1.41	5.70	2

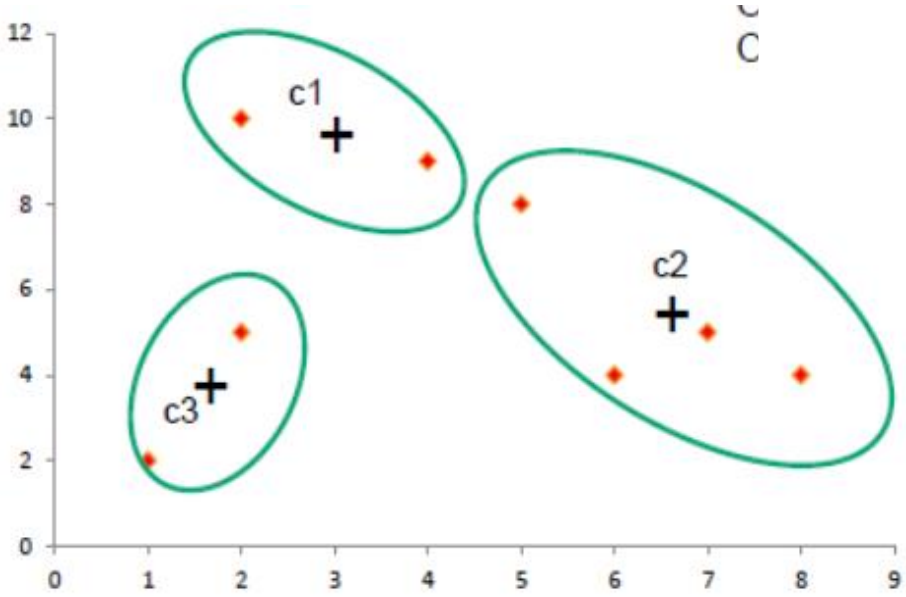
ตัวอย่างการทำ K-MEAN CLUSTERING

รอบที่ 2

Custer 1	Custer 2	Custer 3
A1(2,10)	A3(8,4)	A2(2,5)
A8(4,9)	A4(5,8)	A7(1,2)
	A5(7,5)	
	A6(6,4)	



Clustering : K-mean โดย ผศ.วิภาวรรณ บัวทอง



ตัวอย่างการทำ K-MEAN CLUSTERING

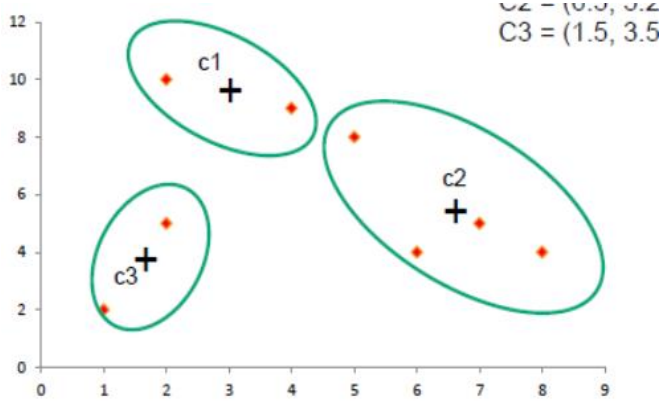
รอบที่ 3

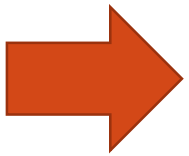
	Point	c1(3,9.5)	c2(6.5,5.25)	c3(1.5,3.5)	Cluster
A1	(2,10)	1.11	6.54	6.52	1
A2	(2,5)	4.61	4.50	1.58	3
A3	(8,4)	7.43	1.96	6.52	2
A4	(5,8)	2.50	3.13	5.70	1
A5	(7,5)	6.02	0.56	5.70	2
A6	(6,4)	6.26	1.35	4.53	2
A7	(1,2)	7.76	6.39	1.58	3

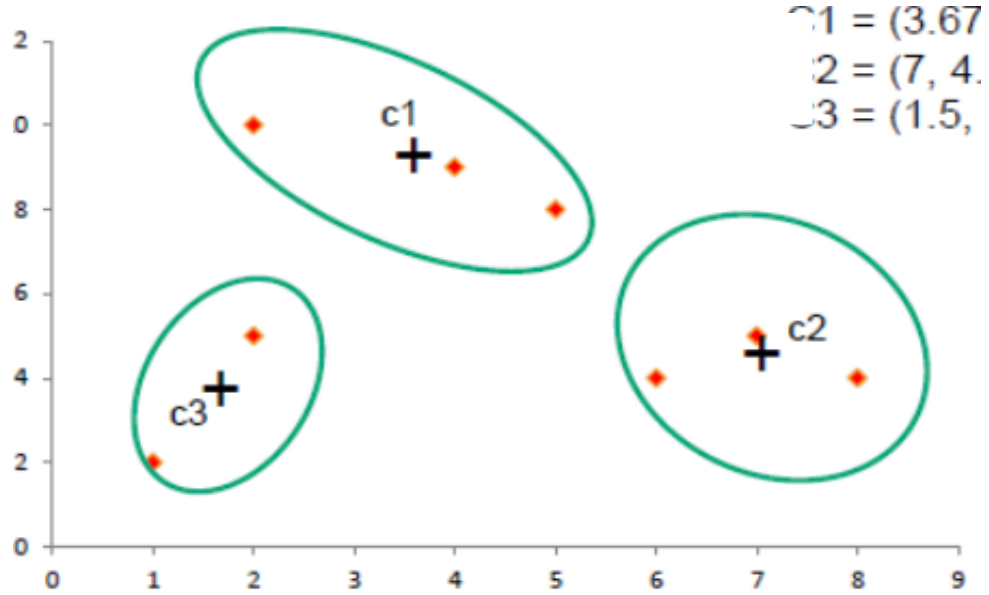
ตัวอย่างการทำ K-MEAN CLUSTERING

รอบที่ 3

Custer 1	Custer 2	Custer 3
A1(2,10)	A3(8,4)	A2(2,5)
A8(4,9)	A4(5,8)	A7(1,2)
A4(5,8)	A6(6,4)	







ตัวอย่างการทำ K-MEAN CLUSTERING

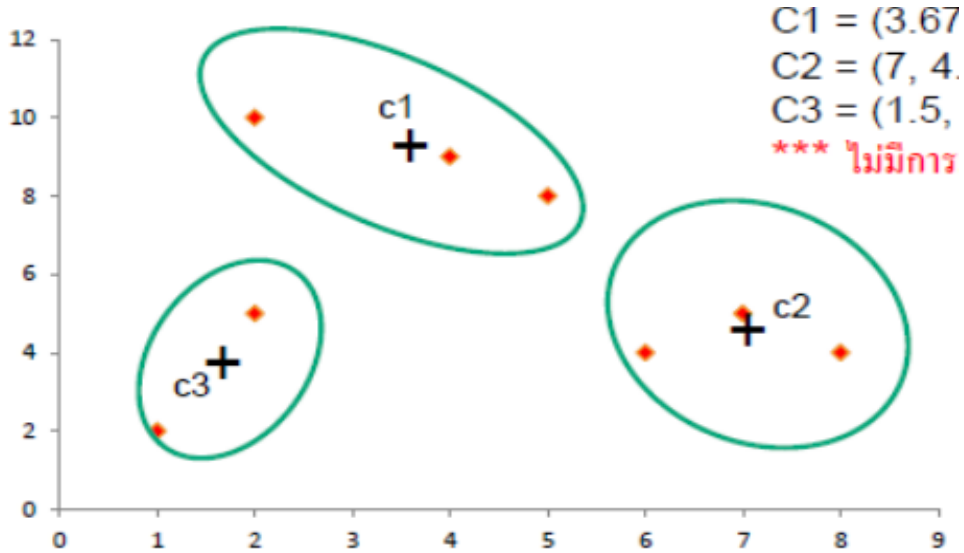
รอบที่ 4

	Point	c1(3.67,9)	c2(7,4.33)	c3(1.5,3.5)	Cluster
A1	(2,10)	1.94	7.56	6.52	1
A2	(2,5)	4.33	5.04	1.58	3
A3	(8,4)	6.62	1.05	6.52	2
A4	(5,8)	1.67	4.18	5.70	1
A5	(7,5)	5.21	0.67	5.70	2
A6	(6,4)	5.52	1.05	4.53	2

ตัวอย่างการทำ K-MEAN CLUSTERING

รอบที่ 4

Custer 1	Custer 2	Custer 3
A1(2,10)	A3(8,4)	A2(2,5)
A8(4,9)	A4(5,8)	A7(1,2)
A4(5,8)	A6(6,4)	



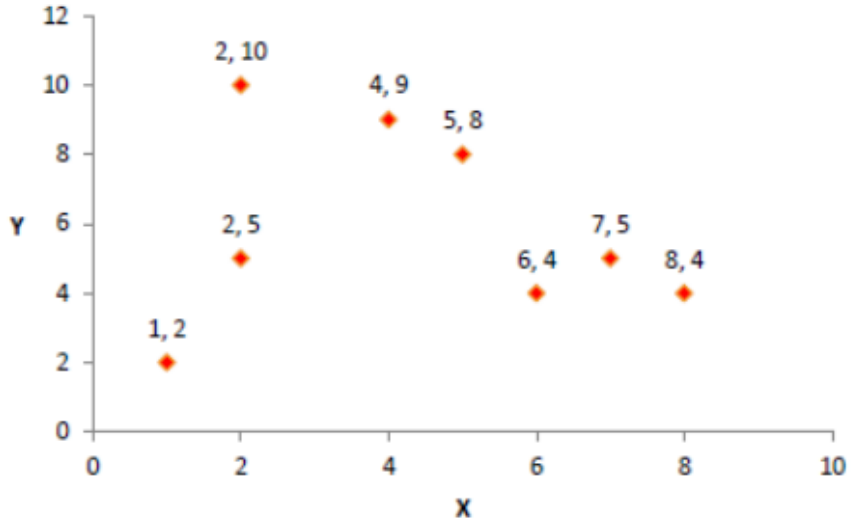
Stop

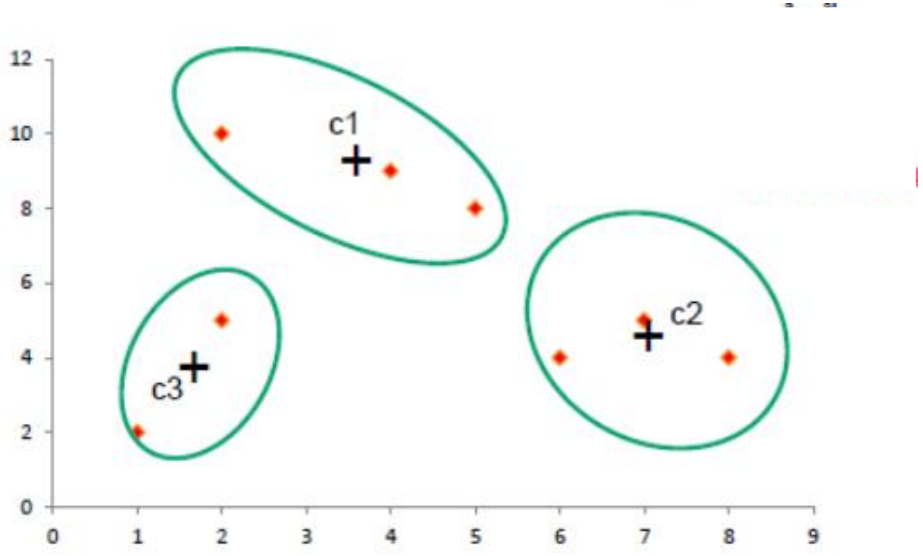
****เมื่อไม่มีการเปลี่ยนแปลงให้หยุดการทำงาน****

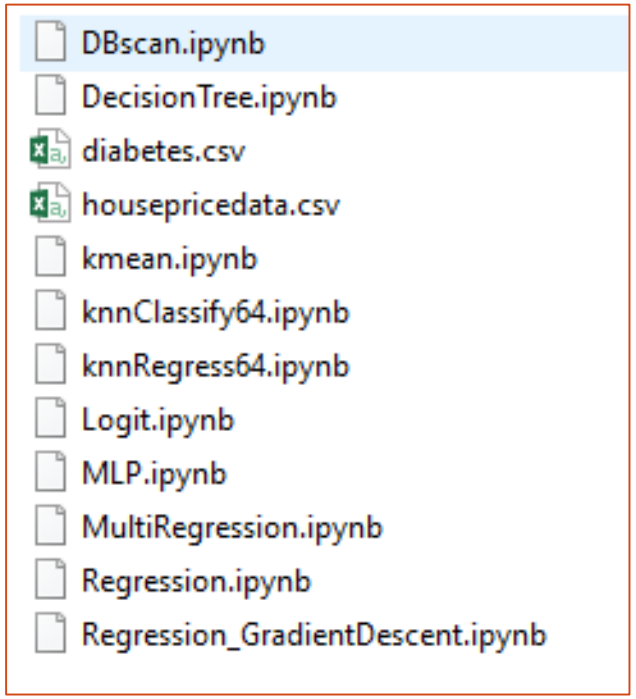
ตัวอย่างการทำ K-MEAN CLUSTERING

ผลลัพธ์

จะได้ผลลัพธ์ดังภาพ






DBscan.ipynb
DecisionTree.ipynb
diabetes.csv
housepricedata.csv
kmean.ipynb
knnClassify64.ipynb
knnRegress64.ipynb
Logit.ipynb
MLP.ipynb
MultiRegression.ipynb
Regression.ipynb
Regression_GradientDescent.ipynb

kmean.ipynb

ข้อดีของการทำ K-Mean Clustering

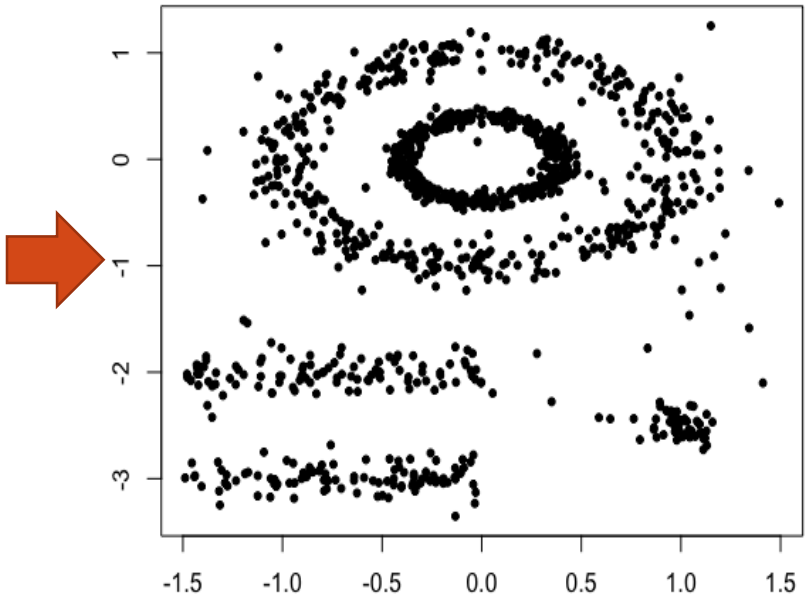
1.เมื่อจำนวนข้อมูลมีจำนวนมาก และมีจำนวนกลุ่มน้อย การหา ค่าเฉลี่ยแบบ K-means อาจจะคำนวณได้เร็วกว่า การจัดกลุ่ม แบบอื่น ๆ (Hierarchical)

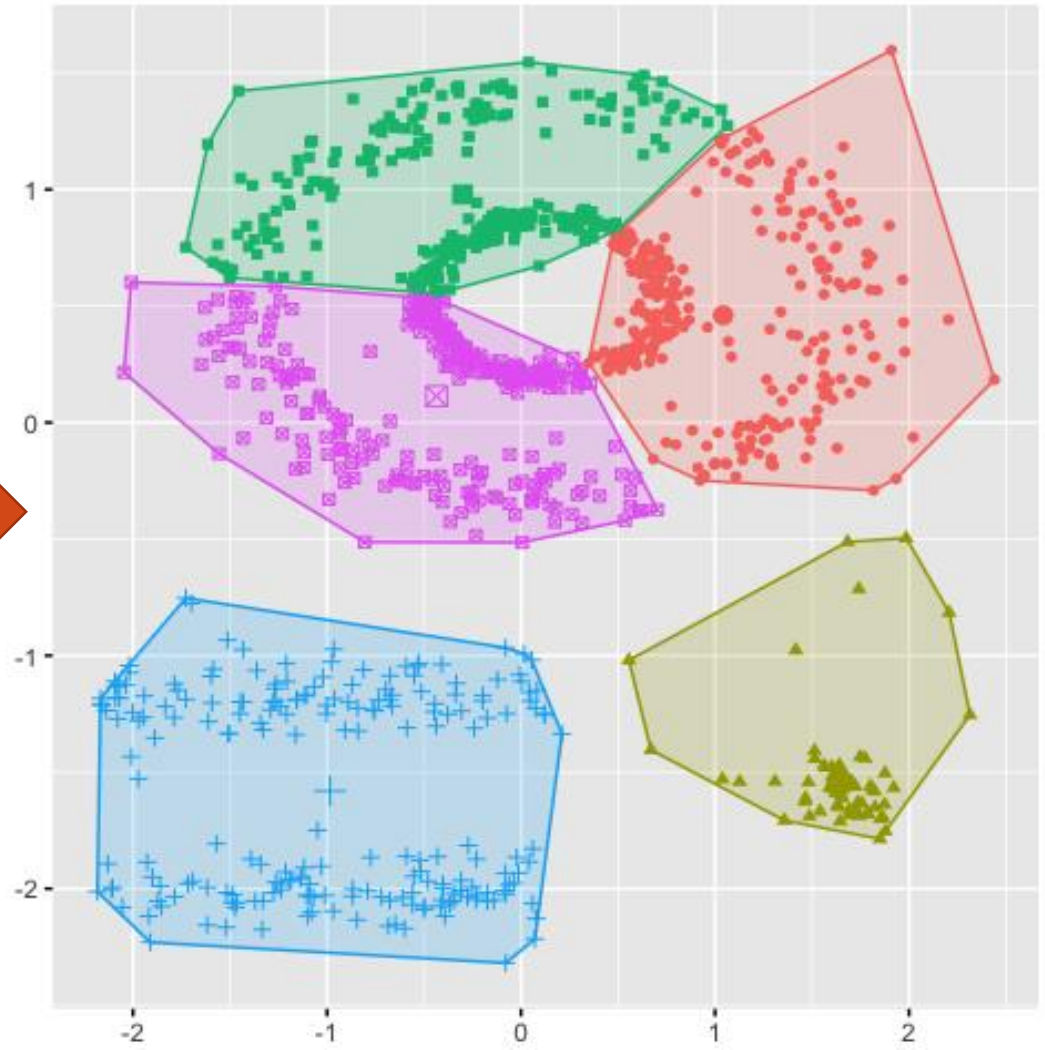
2.ขั้นตอนการหาค่าเฉลี่ยแบบ K-means อาจจะได้สมาชิก ภายในกลุ่ม หนาแน่นกว่าการจัดกลุ่มแบบ Hierarchical โดยเฉพาะถ้ากลุ่มเป็นวงกลม

ข้อด้อยของการทำ K-Mean Clustering

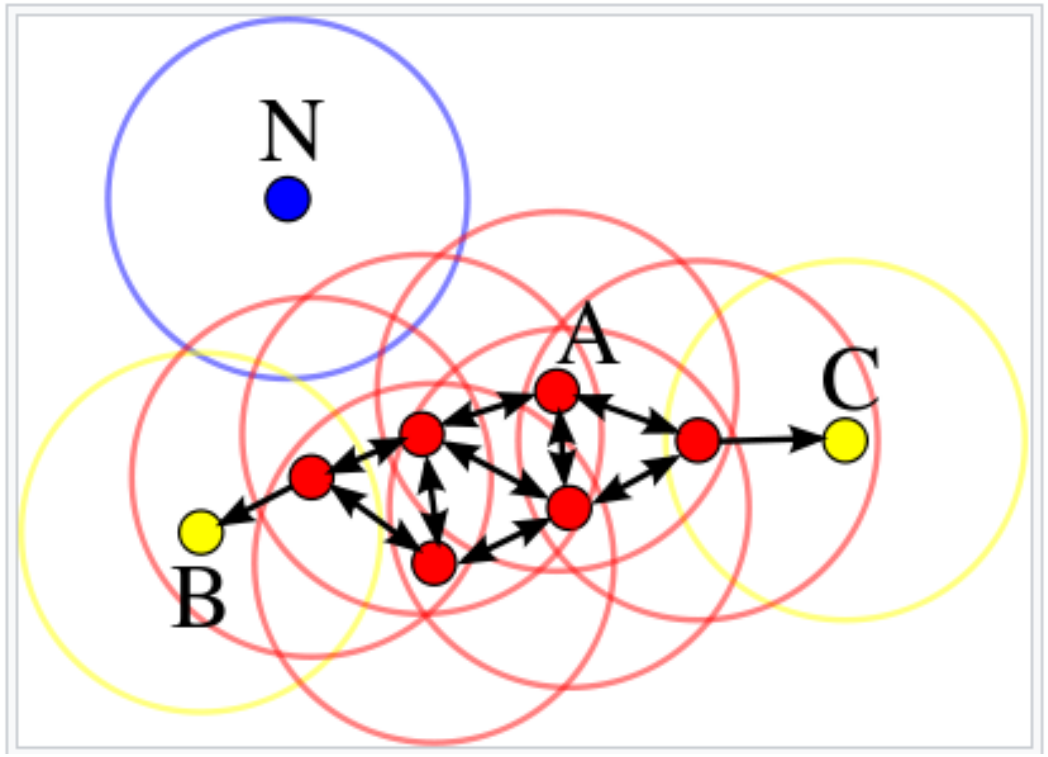
1. การหาค่า K ที่เหมาะสมคาดเดาได้ยาก
2. ทำงานได้ไม่ดีถ้ากลุ่มข้อมูลไม่เป็นรูปวงกลม
3. มีข้อจากัดในเรื่องของขนาด ความหนาแน่น และรูปร่าง

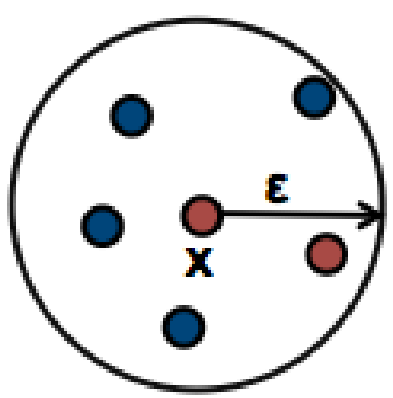
บางครั้งการแบ่งกลุ่มแบบที่ผ่านมา ใช้กับข้อมูลบางลักษณะ ไม่ได้ ต้องใช้การวัดความหนาแน่นข้อมูล





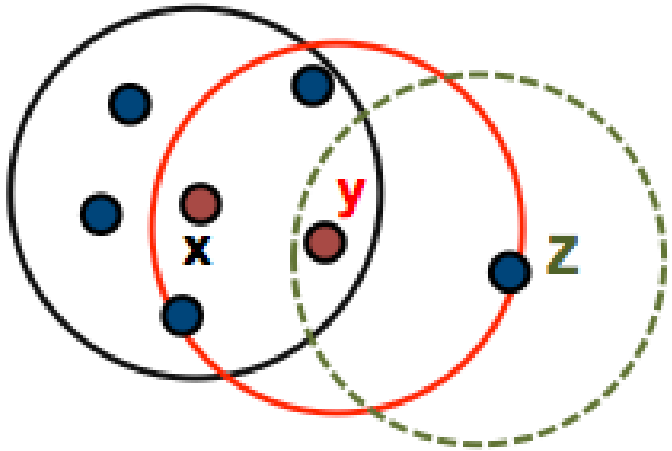
DBScan





MinPts = 6

(a)



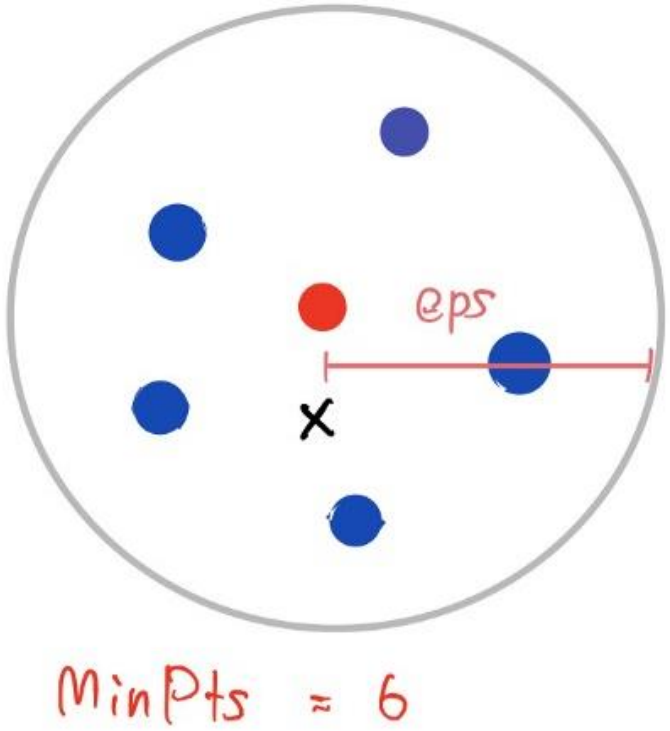
(b)

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

เป็นการหาบริเวณที่ข้อมูลเกาะกลุ่มกัน ซึ่งสามารถคำนวณได้จาก data point ที่อยู่รอบๆ ในรัศมีที่กำหนด

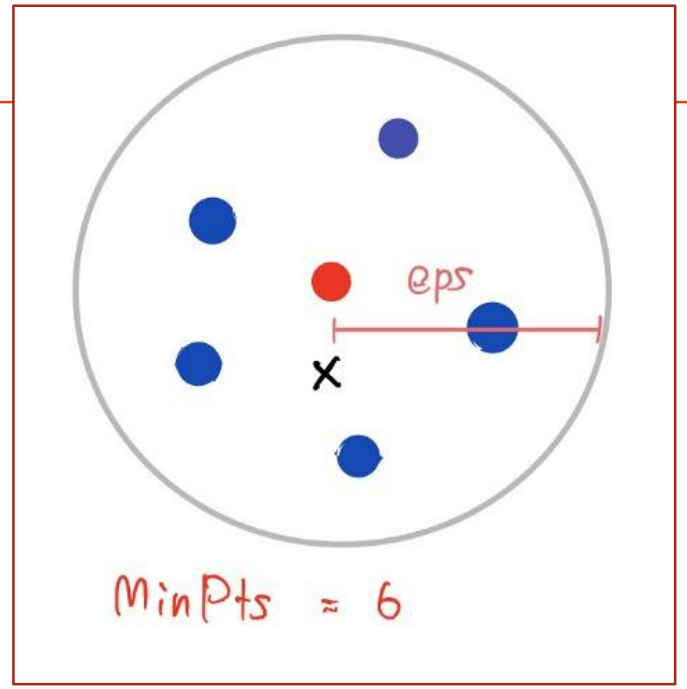
การใช้ DBSCAN ได้มี 2 parameter คือ

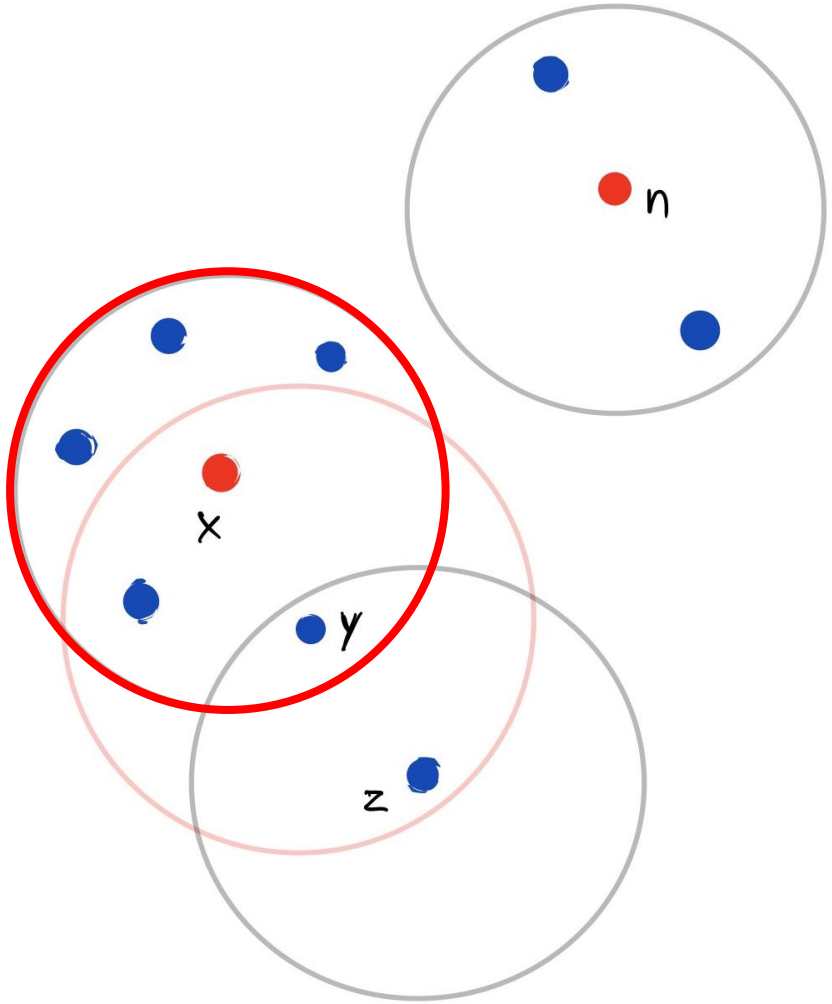
1. eps รัศมีจากจุดศูนย์กลาง
2. MinPts จำนวน data point ขั้นต่ำสำหรับการกำหนด center



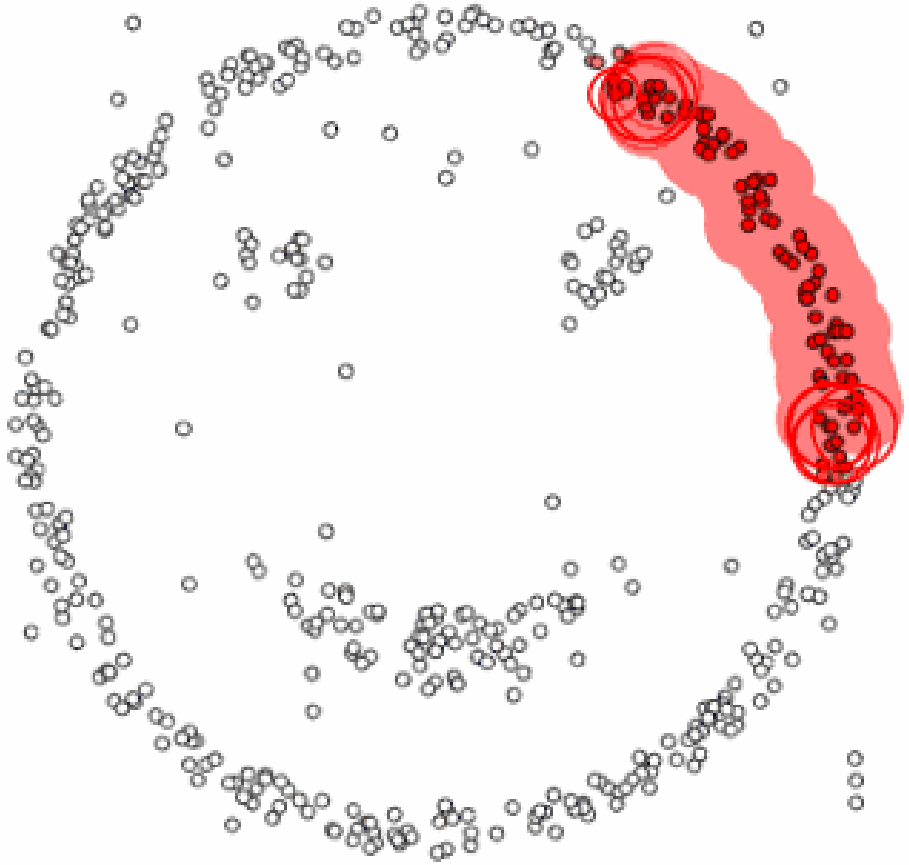
Algorithm ของ DBSCAN

1. ในแต่ละ data point จะคำนวณหา neighbor point ทั้งหมดในรัศมี eps ถ้า data point ไหนมี neighbor point มากกว่าหรือเท่ากับ MinPts ให้ data point นั้นเป็น core point และสร้างเป็น cluster ใหม่
2. ในแต่ละ core point ถ้ามี neighbor ที่เชื่อมต่อกับอีก core point ได้ ให้รวมเป็น cluster เดียวกัน
3. ถ้า data point ไหนไม่เชื่อมต่อกับ core point ก็จะให้ data point นั้นเป็น Noise ซึ่งจะไม่อยู่ใน cluster ใดๆเลย





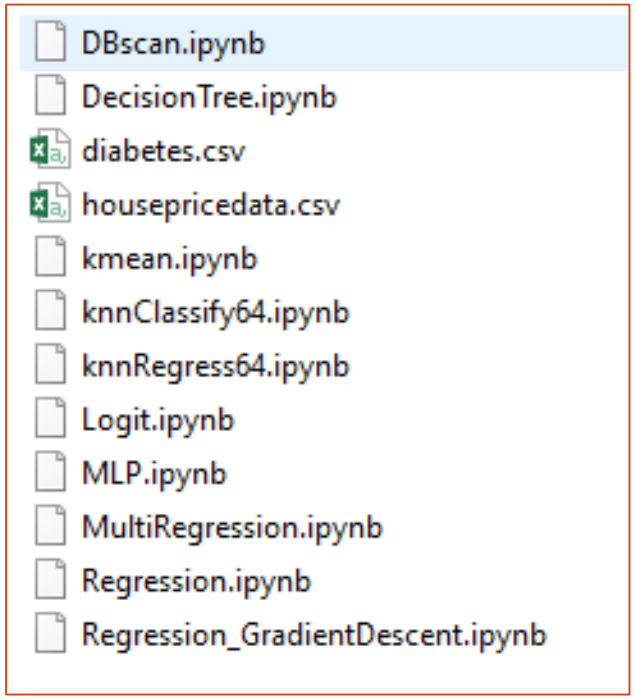
- x เรียกว่า Core point เพราะมี Neighbor point อย่างน้อย 6
- y เรียกว่า Border เพราะมี Neighbor point ไม่ถึง 6 แต่อยู่ในรัศมีของ x
- z เรียกว่า Border เพราะมี Neighbor point ไม่ถึง 6 แต่อยู่ในรัศมีของ y ซึ่ง y ก็อยู่ในรัศมีของ Core point x ทำให้ z ถือว่าอยู่ใน cluster เดียวกันกับ x และ y
- n เรียกว่า Noise หรือ Outlier เพราะจุดนั้นไม่ได้อยู่ในรัศมีของ Core point ใดๆ เลย ซึ่ง Noise นั้นจะเป็นข้อมูลที่เราต้องการตัดออกไป และไม่รวมอยู่ใน Cluster



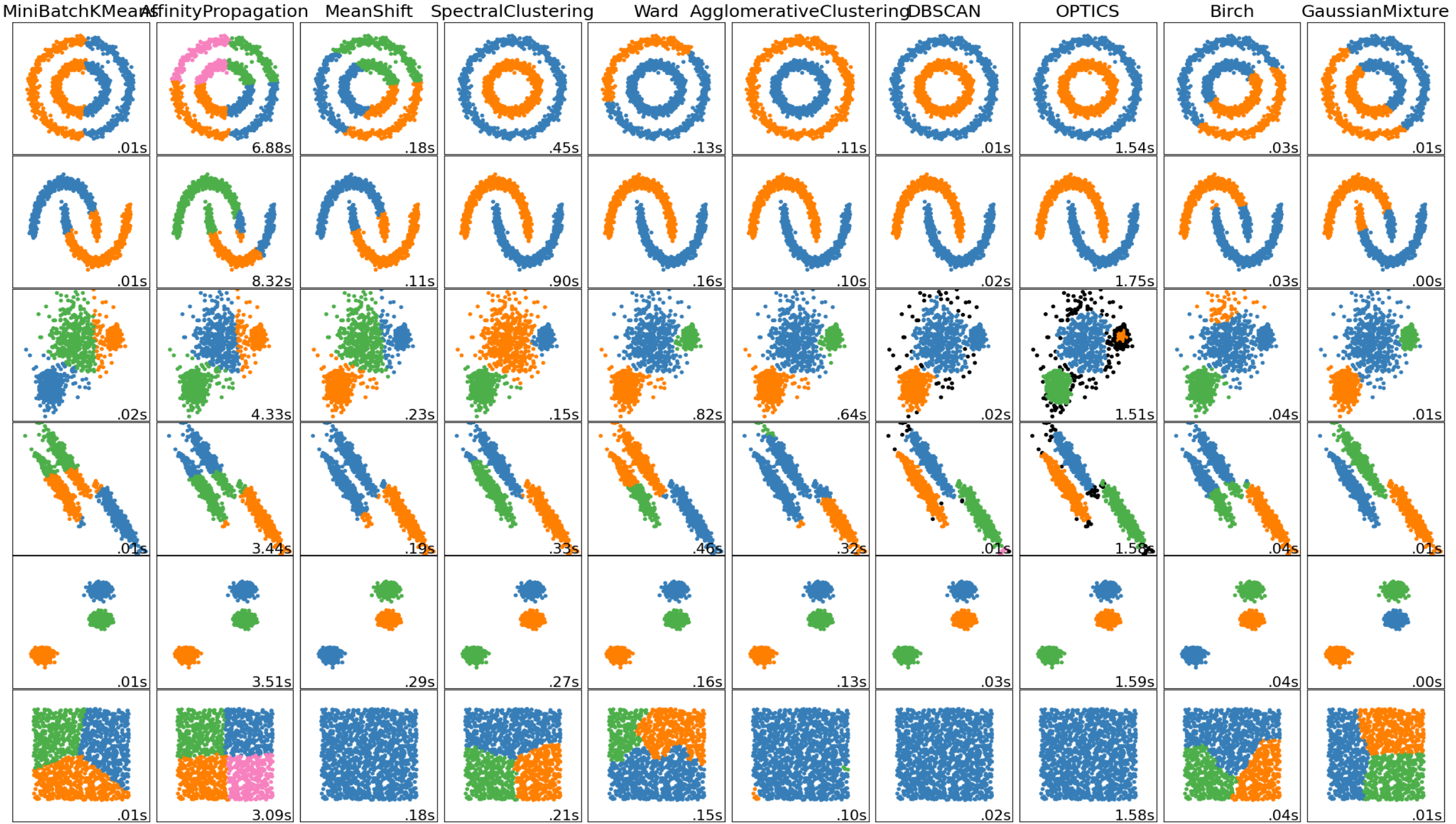
epsilon = 1.00
minPoints = 4

Restart

Pause



DBscan.ipynb


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
OPTICS
Birch
GaussianMixture
.01s
6.88s
.18s
.45s
.13s
.11s
.01s
1.54s
.03s
.01s
.01s
8.32s
.11s
.90s
.16s
.10s
.02s
1.75s
.03s
.00s
.02s
4.33s
.23s
.15s
.82s
.64s
.02s
1.51s
.04s
.01s
.01s
3.44s
.19s
.33s
.46s
.32s
.01s
1.58s
.04s
.01s
.01s
3.51s
.29s
.27s
.16s
.13s
.03s
1.59s
.04s
.00s
.01s
3.09s
.18s
.21s
.15s
.10s
.02s
1.58s
.04s
.01s


MACHINE
LEARNING