

วิชาเหมืองข้อมูล (Data Mining) 4124305

บทที่_6_การจำแนก
ข้อมูล_Data_Classification_Decision_Tree.pptx



Asst. Prof. Dr. Nattapong Songneam

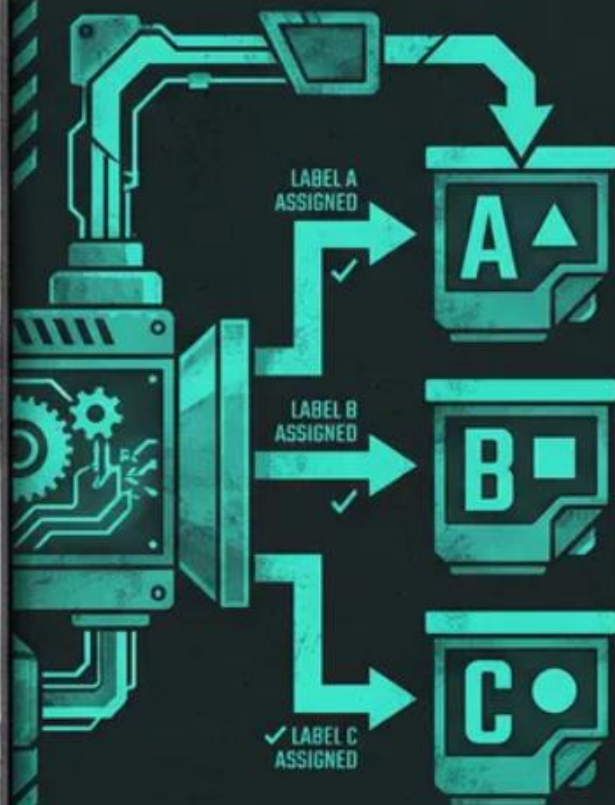
Department of Computer Science,
Faculty of Science and Technology,
Phranakhon Rajabhat University

<http://www.siam2dev.com>

อัปเดตล่าสุด : 25.01.2569

CLASSIFICATION

DISCRETE CATEGORIES



OUTPUT: CATEGORICAL LABELS

REGRESSION

CONTINUOUS CURVE

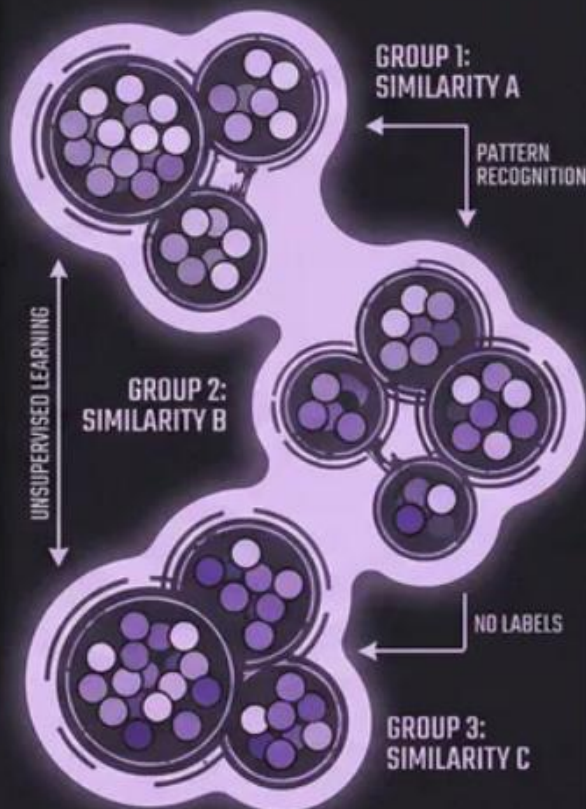


$$Y = f(X) + \epsilon$$

OUTPUT: NUMERICAL VALUE

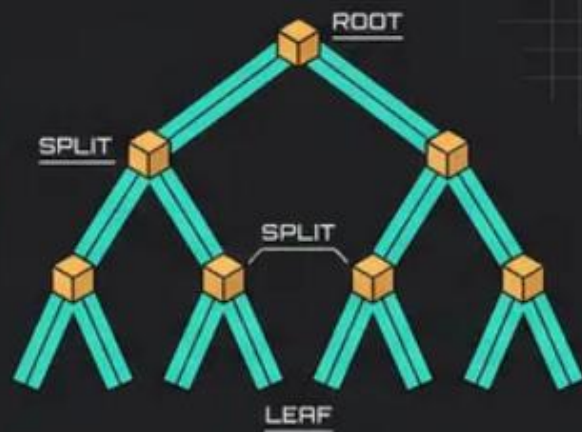
CLUSTERING

GROUPED CIRCULAR CLUSTERS

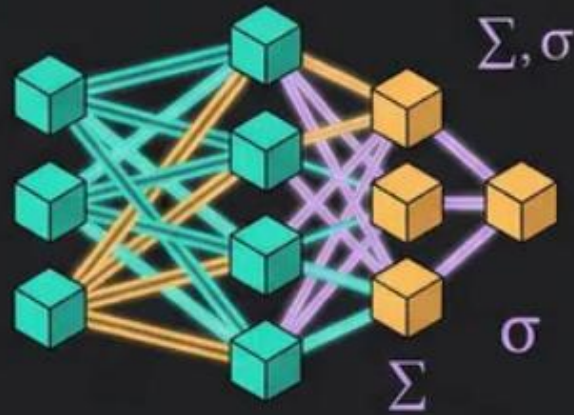


OUTPUT: DATA GROUPS

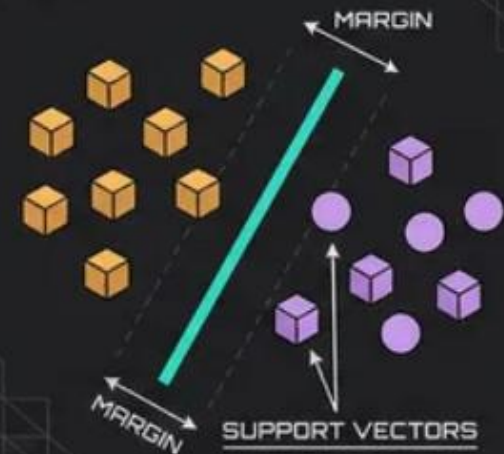
DECISION TREE



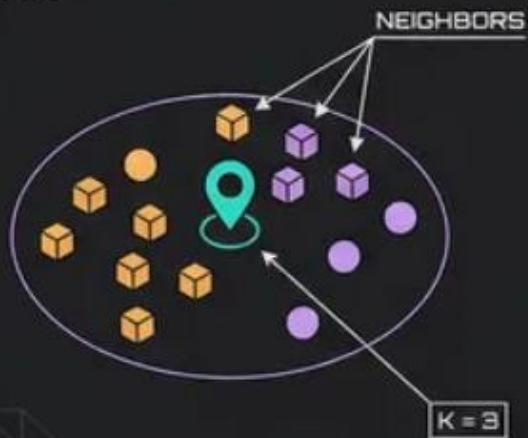
NEURAL NETWORK



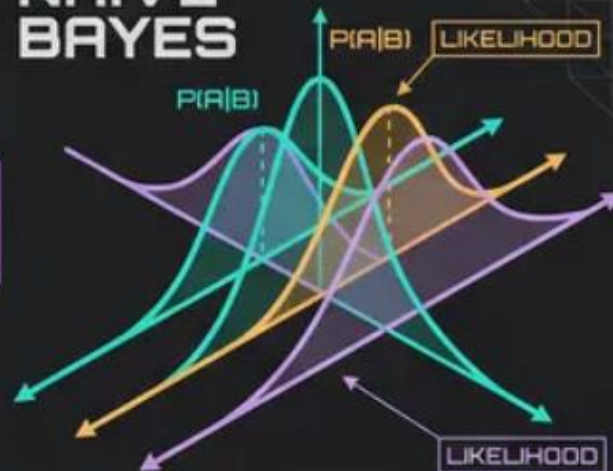
SVM



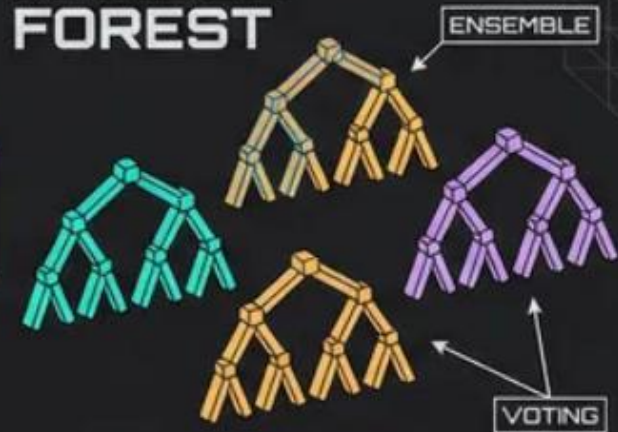
KNN



NAIVE BAYES



RANDOM FOREST





**MEDICAL
DIAGNOSIS**



**FRAUD
DETECTION**



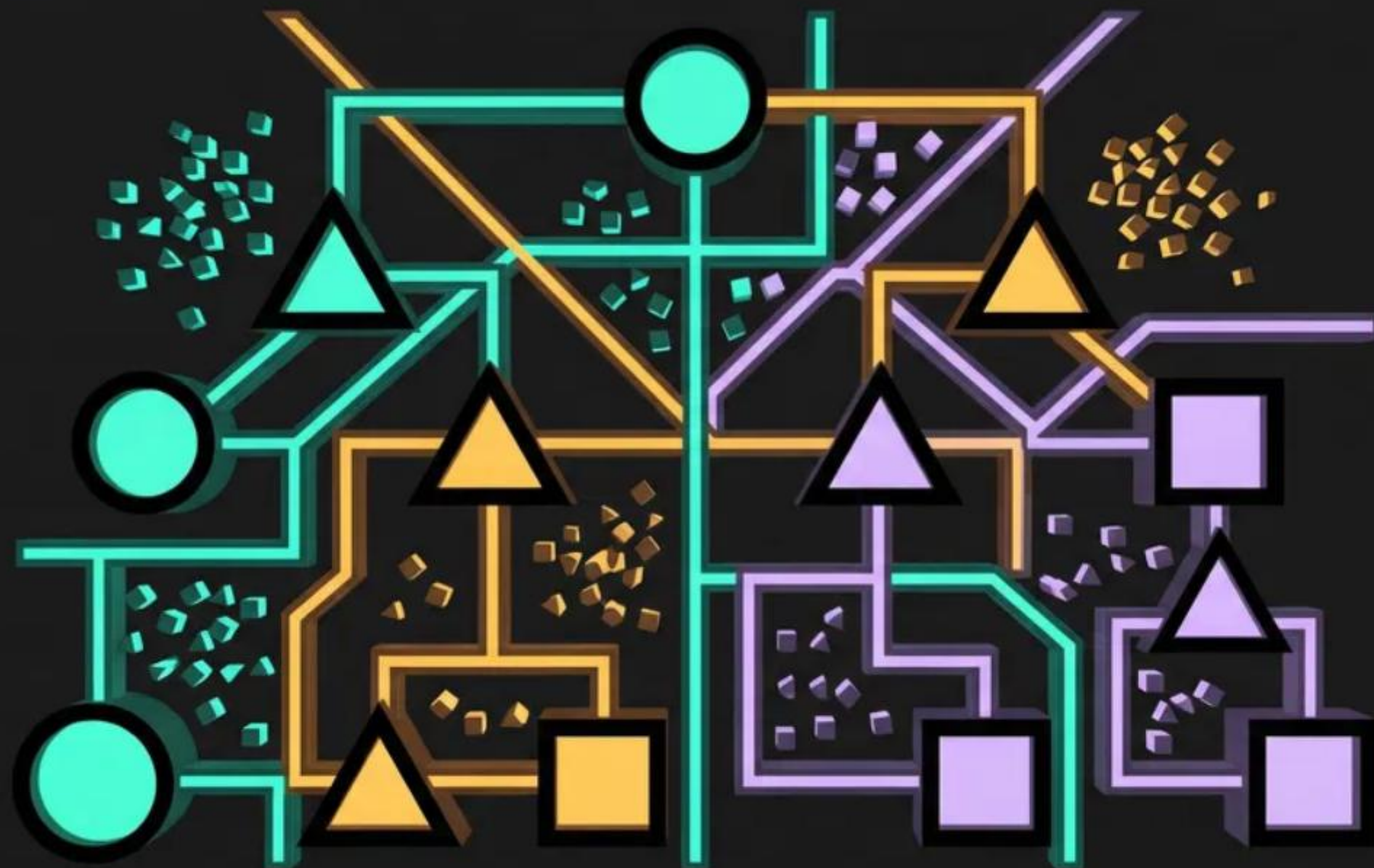
**SPAM
FILTERING**



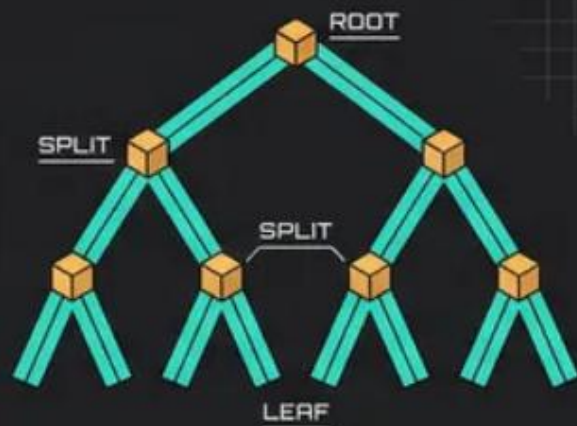
**CUSTOMER
SEGMENTATION**

NEO-BRUTALIST AI CLASSIFICATION FLOWCHART: DATA TO DEPLOYMENT

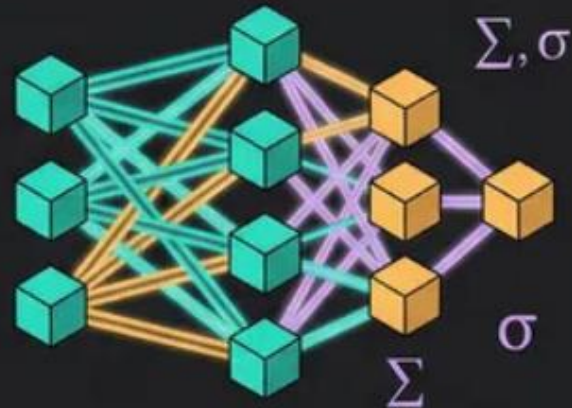




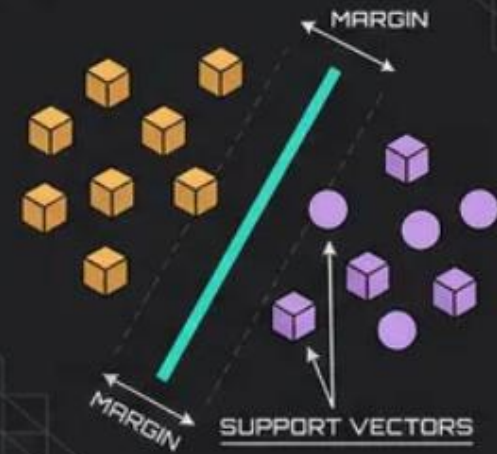
DECISION TREE



NEURAL NETWORK



SVM



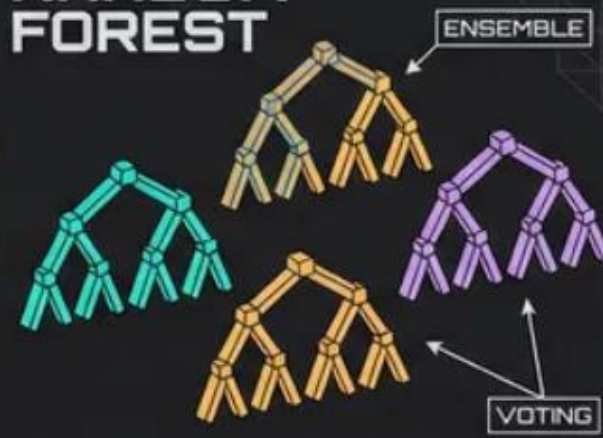
KNN



NAIVE BAYES



RANDOM FOREST



กำหนดรายละเอียดการทำงานกลุ่ม

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของปัญหา

1.2 วัตถุประสงค์

1) ศึกษา

2) พัฒนา ...

3) ประเมิน

1.3 แผนการดำเนินงาน

1.4 ขอบเขต

1.5 ผลที่คาดว่าจะได้รับ

เอกสาร 5 บท

บทที่ 2 การเตรียมข้อมูล

2.1 ทำความเข้าใจเกี่ยวกับธุรกิจ

2.2 การทำความเข้าใจข้อมูล

2.3 การทำความสะอาดข้อมูล

2.4 การเตรียมไฟล์ข้อมูล .csv (.xlsx)

100 รายการ
1000 รายการ

ไฟล์ Word + ไฟล์ข้อมูล CSV

กำหนดส่งงานกลุ่ม

บทที่ 3 การทำเหมืองข้อมูล (การสร้างโมเดล)

3.1 การใช้ต้นไม้ตัดสินใจ

3.2 การหากฎความสัมพันธ์

3.3 การทำ KNN

3.4 การแบ่งข้อมูล สำหรับการฝึกสอน (Training Data set) และการทดสอบ (Test Data set)

บทที่ 4 การวัดประสิทธิภาพโมเดล

4.1 ผลการทดสอบโมเดล ของแต่ละวิธี

4.2 การวัดประสิทธิภาพโมเดล

บทที่ 5 สรุปผล อภิปรายผล และข้อเสนอแนะ

ไฟล์ Word + ไฟล์ข้อมูล CSV
+ ไฟล์ Python

เนื้อหา (Agenda)

6.1 วัตถุประสงค์

6.2 ความหมายของการจำแนกข้อมูล (Classification)

6.3 ความหมายของการทำนาย (Prediction)

6.4 ข้อแตกต่างระหว่าง การทำนายข้อมูล และการจำแนกข้อมูล

6.5 กระบวนการของการจำแนกข้อมูล

6.6 เทคนิคที่ใช้ในการจำแนกข้อมูล (Classification Techniques)

เนื้อหา (Agenda)

6.7 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree)

6.8 ตัววัดประสิทธิภาพของโมเดล (Performance)

6.9 การแบ่งข้อมูลสำหรับการสอนและการทดสอบ (Split data)

6.10 การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล (Validation)

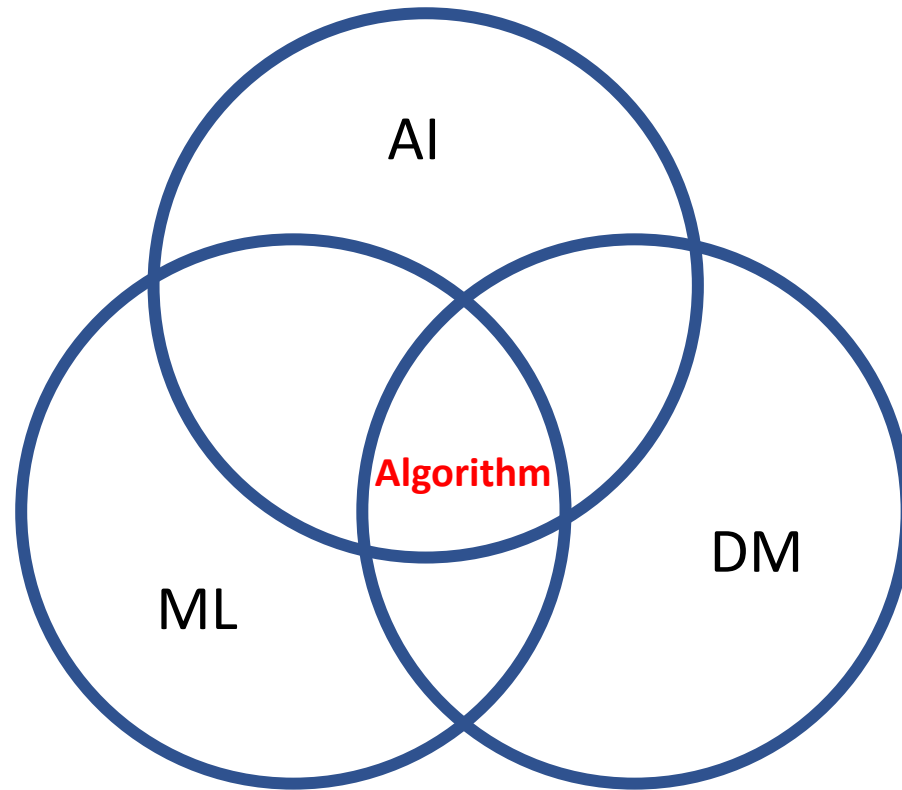
การจำแนกและการทำนายข้อมูล

ในบทนี้จะทำการศึกษาเกี่ยวกับเทคนิคต่าง ๆ ในการจำแนกประเภทของข้อมูล อาทิ เช่น

- **การจำแนกข้อมูลด้วยการสร้างต้นไม้ตัดสินใจ (Decision tree classifier)**
- การจำแนกข้อมูลด้วยเบย์เซียนและเบย์เซียนปี-ลฟเน็คเวิร์ค (Bayesian classifier and Bayesian belief networks)
- การจำแนกข้อมูลด้วยกฎ (Rule-based classifiers)
- การจำแนกข้อมูลด้วยโครงข่ายประสาทเทียมและการส่งค่าย้อนกลับ (Neural network and backpropagation)
- การจำแนกข้อมูลจากกฎความสัมพันธ์ของข้อมูล (Classification based on association rule mining)
- การค้นหาเพื่อนบ้านใกล้สุด k อันดับ (k -nearest-neighbor)

การทำนายข้อมูลและทำการศึกษาเกี่ยวกับการทำนายข้อมูล ที่จะประกอบไปด้วยการถดถอยเชิงเส้นตรง (linear regression) และการถดถอยที่ไม่เป็นเส้นตรง (Nonlinear regression) ท้ายสุด—เราจะทำการศึกษาเกี่ยวกับการวัดความถูกต้องและความผิดพลาดในการจำแนกข้อมูล รวมถึงส่วนต่อเติมหรือวิธีการเพิ่มประสิทธิภาพในแง่มุมต่าง ๆ สำหรับการจำแนกและการทำนายข้อมูล

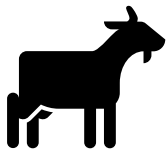
ความสัมพันธ์ระหว่าง AI , ML และ DM



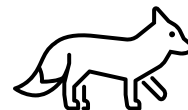
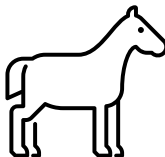
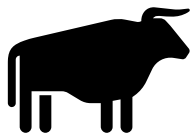
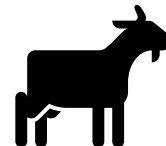
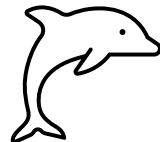
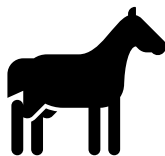
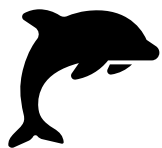
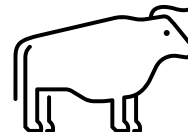
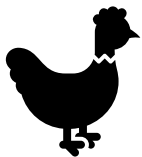
CRISP-DM 6 STEP

FEATURE EXTRACTION

UNKNOWN



สิ่งยังไม่รู้



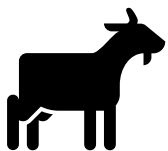
Retrieval

Recognition

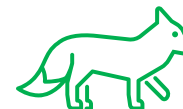
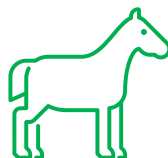
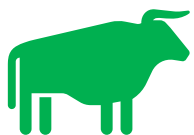
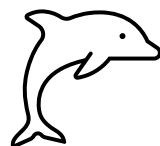
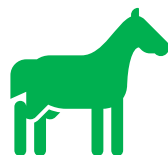
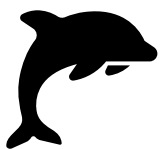
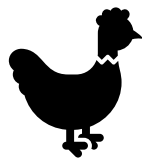
FEATURE EXTRACTION

หรือ FEATURE ของภาพ

UNKNOWN



สิ่งยังโมรู้



Retrieval

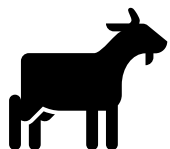
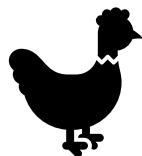
การค้นคืนภาพ

Recognition

FEATURE EXTRACTION

หรือ FEATURE ของภาพ

UNKNOWN



สิ่งยังไม่รู้

Retrieval

การค้นคืนภาพ

255	123	211	211	112	231				

R

ION

หรือ FEATURE ของภาพ

RGB



สั่งยังไม่

F

N

การค้นคืนภาพ

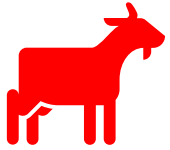
FEATURE EXTRACTION

- Color
 - SHAPE
 - TEXTURE
- Classification (การจำแนกข้อมูล)
 - Clustering (การแบ่งกลุ่มข้อมูล)
- Color
 - RED
 - BLACK

FEATURE EXTRACTION

การแยกแยะคุณลักษณะของข้อมูล

การหาลักษณะเด่น



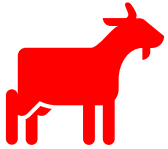
Data	Attribute1 เขา	Attribute2 จำนวนขา	Attribute3 ขน	Attribute4 ปีก	Attribute1 กินพืช
แมว	0	4	1	0	0
สุนัข	0	0	0	0	-1
วัว	1	3	1	0	6
แกะ	1	7	0	0	-1
นก	0	2	1	1	1

Outlier ค่าผิดปกติ

FEATURE EXTRACTION

การแยกแยะคุณลักษณะของข้อมูล

การหาลักษณะเด่น



การเรียนรู้แบบมีผู้สอน (Supervise Learning)

1 แมว

2 วัว

Data	Attribute1 เขา	Attribute2 จำนวนขา	Attribute3 ขน	Attribute4 ปีก	Attribute1 กินพืช	Class
แมว	1	4	1	0	0	1
สุนัข	0	0	0	1	1	1
วัว	1	3	1	1	6	1
แกะ	1	7	0	1	1	2
unkno wn	1	4	1	0	1	???

FEATURE EXTRACTION

การแยกแยะคุณลักษณะของข้อมูล

การหาคุณลักษณะเด่น

การเรียนรู้แบบมีผู้สอน (Supervise Learning)



1 แมว

2 วัว

ข้อมูลสำหรับการสอน (Training Dataset)

Data	Attribute1 เขา	Attribute2 จำนวนขา	Attribute3 ขน	Attribute4 ปีก	Attribute1 กินพืช	Class
แมว	1	4	1	0	0	1
สุนัข	0	0	0	1	1	1
วัว	1	3	1	1	6	1
แกะ	1	7	0	1	1	2
unkno wn	1	4	1	0	1	???

ข้อมูลสำหรับการทดสอบ (Testing Dataset)

6.1 วัตถุประสงค์

1. เพื่อศึกษาและเข้าใจหลักการ การจำแนกข้อมูล (Classification) และการ Regression เบื้องต้น
2. เพื่อการสร้างโมเดลที่สามารถจำแนกหรือแยกข้อมูลออกเป็นกลุ่มหรือหมวดหมู่ต่าง ๆ
3. เพื่อการสร้างโมเดลการทำนายด้วยวิธี ต้นไม้ตัดสินใจ (decision tree)

6.2 ความหมายของการจำแนกและการทำนายข้อมูล

- ในการที่จะเข้าใจว่าการจำแนกข้อมูลคืออะไร ลองพิจารณาตัวอย่างดังต่อไปนี้
- 1) พนักงานสินเชื่อของธนาคารต้องการที่จะทำการวิเคราะห์ข้อมูลเพื่อที่จะทำการศึกษาว่าการกู้ยืมในครั้งหนึ่ง ๆ มีครั้งไหนบ้างที่ปลอดภัยและครั้งไหนบ้างที่มีความเสี่ยง
- 2) ผู้จัดการฝ่ายการตลาดของบริษัทขายอุปกรณ์ไฟฟ้าต้องการข้อมูลเพื่อช่วยในการคาดเดาว่า “ลูกค้ามีคุณลักษณะอย่างไรที่จะทำการซื้อคอมพิวเตอร์จากบริษัท”
- 3) นักวิจัยทางการแพทย์ต้องการที่จะวิเคราะห์ข้อมูลเกี่ยวกับมะเร็งเต้านมเพื่อที่จะทำการทำนายว่าผู้ป่วยควรจะได้รับการดูแลด้วยวิธีใดภายใต้วิธีการรักษาทั้ง 3 วิธีที่ได้รับความนิยมอย่างแพร่หลาย
- จากตัวอย่างทั้ง 3 ข้างต้นจะต้องการการวิเคราะห์ข้อมูลซึ่งก็คือ การจำแนกข้อมูล (Classification)

6.2 ความหมายของการจำแนกและการทำนายข้อมูล

- ที่ซึ่งจะทำการสร้างโมเดลหรือตัวจำแนกข้อมูล (Classifier) เพื่อทำนายหมวดหมู่ของข้อมูล (Categories/Class) เช่น แบ่งออกเป็น 2 กลุ่ม ปลอดภัย หรือ เสี่ยง ในการวิเคราะห์ข้อมูลสินเชื่อ วิธีการรักษา A หรือ B หรือ C ในการวิเคราะห์ข้อมูลของผู้ป่วยมะเร็งเต้านม เป็นต้น ในส่วนของการทำนายข้อมูล ลองพิจารณาตัวอย่างดังต่อไปนี้—สมมติว่าผู้จัดการฝ่ายการตลาดต้องการที่จะทำนายหรือคาดเดาลูกค้าคนหนึ่ง ๆ จะทำการจ่ายเงินซื้อสินค้าจากบริษัทเป็นจำนวนเท่าไร การวิเคราะห์ข้อมูลลักษณะนี้จะเป็นส่วนของการทำนายข้อมูลเชิงตัวเลข (Numeric prediction)

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

- การจำแนกข้อมูลจะประกอบไปด้วย 2 กระบวนการหลัก ดังแสดงตัวอย่างในภาพที่ 6.1 และ 6.2 ที่เป็นการกู้-ยืมเงิน โดยจากภาพที่ 6.1 จะเป็นกระบวนการ *การสร้างตัวจำแนกข้อมูล* จากชุดของข้อมูลที่เป็นอินพุต ที่ซึ่งแต่ละเรคคอร์ดของข้อมูล que ทำการพิจารณาจะประกอบไปด้วยเซตของแอทริบิวต์ที่บ่งบอกถึงคุณลักษณะของบุคคลที่ทำการกู้-ยืมเงิน และหมวดหมู่ของบุคคลนั้น ๆ ว่ามีความปลอดภัยหรือมีความเสี่ยงในการให้กู้-ยืมเงินหรือไม่ โดยกระบวนการสร้างตัวจำแนกข้อมูลมักถูกเรียกว่า 'learning' หรือ 'training'
- ขั้นตอนของการเรียนรู้ หรือการสอน ที่เกิดจากการนำเอาขั้นตอนวิธีสำหรับการจำแนกข้อมูลมาดำเนินการกับข้อมูล ข้อมูลเรคคอร์ด X หนึ่ง ๆ ในชุดข้อมูลที่ทำการพิจารณาจะประกอบไปด้วยเซตของแอทริบิวต์
- $X=(x_1,x_2,...,x_n)$

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

ถ้าเราจะทำนาย apple

	ลำดับ	รูปร่าง	สี	ก้าน	เมล็ด	ใบ	คลาส
X	1	ทรงกลม	แดง	มี	มี	มี	ใช่
	2	ทรงรี	เขียว	มี	มี	ไม่มี	ไม่ใช่
	3	ทรงกลม	เขียว	มี	มี	มี	ใช่

class label attribute

คลาส / คำตอบ

Y

ลักษณะเด่น(Feature) ของ apple : รูปร่าง , สี , ก้าน , เมล็ด , ใบ

ลำดับ	รูปร่าง	สี	ก้าน	เมล็ด	ใบ	คลาส			
1	ทรงกลม	แดง	0.1	มี	มี	มี	1	ใช้	
2	ทรงรี	เขียว		0.2	มี	มี	ไม่มี	0	ไม่ใช้
3	ทรงกลม	เขียว			มี	มี	มี	0	ใช้
4	กลมรี	เหลือง	0.3	มี	มี	มี		ใช้	
5	ยาวรี	ม่วง		มี	มี	ไม่มี		ไม่ใช้	
6	กลม	ส้ม		มี	มี	มี		ใช้	
7	ทรงรี	ดำ		มี	มี	ไม่มี		ไม่ใช้	
8	กลม	ขาว		มี	มี	มี		ใช้	
9	ยาว	น้ำตาล		มี	มี	ไม่มี		ไม่ใช้	
10	กลม	ชมพู		มี	มี	มี		ใช้	
11	รูปไข่	แดง		มี	มี	ไม่มี		ไม่ใช้	
12	กลมแบน	เขียว		มี	มี	มี		ใช้	
13	สามเหลี่ยม	เหลือง		มี	มี	ไม่มี		ไม่ใช้	
14	สี่เหลี่ยม	ส้ม		มี	มี	มี		ใช้	
15	ห้าเหลี่ยม	ม่วง		มี	มี	ไม่มี		ไม่ใช้	
16	หกเหลี่ยม	ดำ		มี	มี	มี		ใช้	
17	เจ็ดเหลี่ยม	ขาว		มี	มี	ไม่มี		ไม่ใช้	
18	แปดเหลี่ยม	น้ำตาล		มี	มี	มี		ใช้	
19	เก้าเหลี่ยม	ชมพู		มี	มี	ไม่มี		ไม่ใช้	
20	สิบเหลี่ยม	แดง		มี	มี	มี		ใช้	

ลำดับ	รูปร่าง	สี	ก้าน	เมล็ด	ใบ	คลาส
1	ทรงกลม	แดง	มี	มี	มี	ใช่
2	ทรงรี	เขียว	มี	มี	ไม่มี	ไม่ใช่
3	ทรงกลม	เขียว	มี	มี	มี	ใช่
4	กลมรี	เหลือง	มี	มี	มี	ใช่
5	ยาวรี	ม่วง	มี	มี	ไม่มี	ไม่ใช่
6	กลม	ส้ม	มี	มี	มี	ใช่
7	ทรงรี	ดำ	มี	มี	ไม่มี	ไม่ใช่
8	กลม	ขาว	มี	มี	มี	ใช่
9	ยาว	น้ำตาล	มี	มี	ไม่มี	ไม่ใช่
10	กลม	ชมพู	มี	มี	มี	ใช่
11	รูปไข่	แดง	มี	มี	ไม่มี	ไม่ใช่
12	กลมแบน	เขียว	มี	มี	มี	ใช่
13	สามเหลี่ยม	เหลือง	มี	มี	ไม่มี	ไม่ใช่
14	สี่เหลี่ยม	ส้ม	มี	มี	มี	ใช่
15	ห้าเหลี่ยม	ม่วง	มี	มี	ไม่มี	ไม่ใช่
16	หกเหลี่ยม	ดำ	มี	มี	มี	ใช่
17	เจ็ดเหลี่ยม	ขาว	มี	มี	ไม่มี	ไม่ใช่
18	แปดเหลี่ยม	น้ำตาล	มี	มี	มี	ใช่
19	เก้าเหลี่ยม	ชมพู	มี	มี	ไม่มี	ไม่ใช่
20	สิบเหลี่ยม	แดง	มี	มี	มี	ใช่
21	ทรงกลม	แดง	ไม่มี	มี	มี	ใช่
22	ทรงรี	เขียว	ไม่มี	มี	ไม่มี	ไม่ใช่
23	ทรงกลม	เขียว	ไม่มี	มี	มี	ใช่
24	กลมรี	เหลือง	ไม่มี	มี	มี	ใช่
25	ยาวรี	ม่วง	ไม่มี	มี	ไม่มี	ไม่ใช่
26	กลม	ส้ม	ไม่มี	มี	มี	ใช่
27	ทรงรี	ดำ	ไม่มี	มี	ไม่มี	ไม่ใช่
28	กลม	ขาว	ไม่มี	มี	มี	ใช่
29	ยาว	น้ำตาล	ไม่มี	มี	ไม่มี	ไม่ใช่
30	กลม	ชมพู	ไม่มี	มี	มี	ใช่
31	รูปไข่	แดง	ไม่มี	มี	ไม่มี	ไม่ใช่
32	กลมแบน	เขียว	ไม่มี	มี	มี	ใช่
33	สามเหลี่ยม	เหลือง	ไม่มี	มี	ไม่มี	ไม่ใช่
34	สี่เหลี่ยม	ส้ม	ไม่มี	มี	มี	ใช่
35	ห้าเหลี่ยม	ม่วง	ไม่มี	มี	ไม่มี	ไม่ใช่
36	หกเหลี่ยม	ดำ	ไม่มี	มี	มี	ใช่
37	เจ็ดเหลี่ยม	ขาว	ไม่มี	มี	ไม่มี	ไม่ใช่
38	แปดเหลี่ยม	น้ำตาล	ไม่มี	มี	มี	ใช่
39	เก้าเหลี่ยม	ชมพู	ไม่มี	มี	ไม่มี	ไม่ใช่
40	สิบเหลี่ยม	แดง	ไม่มี	มี	มี	ใช่
41	กลม	แดง	มี	ไม่มี	มี	ใช่
42	ทรงรี	เขียว	มี	ไม่มี	ไม่มี	ไม่ใช่
43	ทรงกลม	เขียว	มี	ไม่มี	มี	ใช่
44	กลมรี	เหลือง	มี	ไม่มี	มี	ใช่

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

ถ้าเราจะทำนาย apple

×

ลำดับ	รูปร่าง	สี	ก้าน	เมล็ด	ใบ	คลาส
1	ทรงกลม	แดง	มี	มี	มี	ใช่
2	ทรงรี	เขียว	มี	มี	ไม่มี	ไม่ใช่
3	ทรงกลม	เขียว	มี	มี	มี	ใช่

คลาส / คำตอบ

class label attribute

ลักษณะเด่นของ apple : รูปร่าง , สี , ก้าน , เมล็ด , ใบ

ถ้าเราจะทำนาย apple

class label attribute

คลาส / คำตอบ

ลักษณะเด่นของ apple : รูปร่าง , สี , ก้าน , เมล็ด , ใบ

X

ลำดับ	รูปร่าง	สี	ก้าน	เมล็ด	ใบ	คลาส
1	ทรงกลม	แดง	มี	มี	มี	ใช่
2	ทรงรี	เขียว	มี	มี	ไม่มี	ไม่ใช่
3	ทรงกลม	เขียว	มี	มี	มี	ใช่

ลักษณะเด่นของ apple : รูปร่าง , สี , ก้าน , เมล็ด , ใบ

ถ้าเราจะทำนาย **สินเชื่อ**

class label attribute **คลาส / คำตอบ**

ลักษณะเด่นของ ลูกค้า : เงินเดือน รายได้ รายจ่าย ประวัติการชำระ

X

ลำดับ	อายุ	อาชีพ	รายได้	รายจ่าย	เงินเดือน	ประวัติ	คลาส
1	36	อาจารย์	50000	30000	40000	ล่าช้า	เสี่ยง
2	40	หมอ	70000	40000	50000	ไม่มี	ปลอดภัย
3	50	ภารโรง	30000	10000	20000	ไม่มี	ปลอดภัย
4	24	พนักงานออฟฟิศ	40000	20000	30000	ค้างชำระ	เสี่ยง

ข้อมูลสำหรับสอน (training data)

*** จำนวนของคุณลักษณะ (Attribute) ไม่มากไป ไม่น้อยไป
ขึ้นอยู่กับข้อมูล
ขึ้นอยู่กับวิธีการ (Algorithm)

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

ถ้าเราจะทำนาย apple

×

ลำดับ	รูปร่าง	สี	ก้าน	เมล็ด	ใบ	คลาส
1	ทรงกลม	แดง	มี	มี	มี	ใช่
2	ทรงรี	เขียว	มี	มี	ไม่มี	ไม่ใช่
3	ทรงกลม	เขียว	มี	มี	มี	ใช่

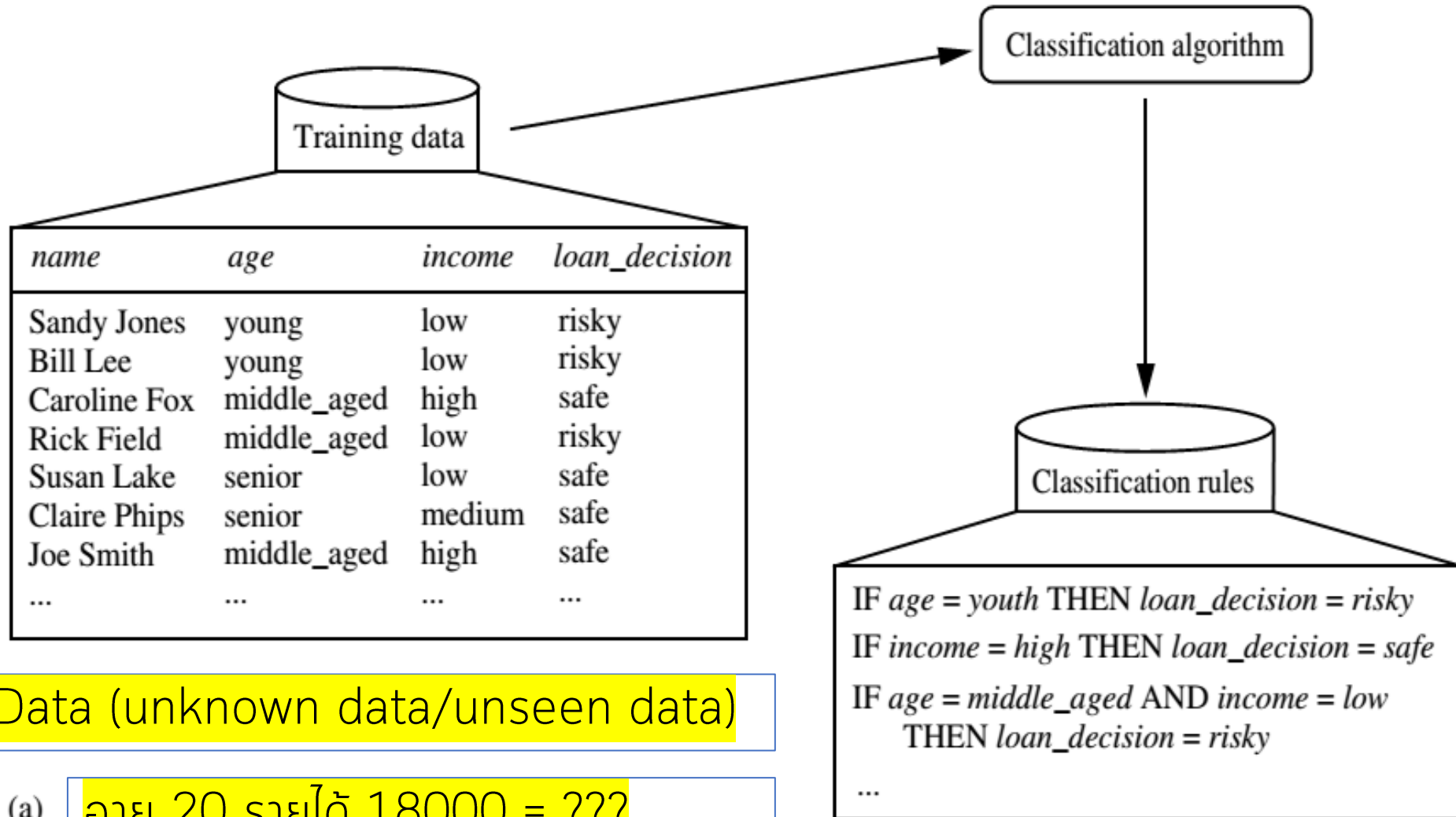
คลาส / คำตอบ

class label attribute

ลักษณะเด่นของ apple : รูปร่าง , สี , ก้าน , เมล็ด , ใบ

ความหมายของการจำแนกและการทำนายข้อมูล(ต่อ)

กระบวนการเรียนรู้ (Training)



Testing Data (unknown data/unseen data)

(a) อายุ 20 รายได้ 18000 = ???

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

n แอตริบิวต์ที่บ่งบอกถึงคุณลักษณะต่าง ๆ ของข้อมูลเรคคอร์ด X นอกจากนั้นเรคคอร์ด X ยังมีข้อมูลอีกหนึ่งแอตริบิวต์ที่บ่งบอกถึงหมวดหมู่ของข้อมูล (class label attribute) โดยแอตริบิวต์หมวดหมู่ข้อมูลจะเป็นข้อมูลแบบไม่ต่อเนื่อง (discrete-valued) โดยชุดข้อมูลที่เป็นอินพุตสำหรับการสร้างตัวจำแนกข้อมูลจะถูกเรียกว่า “ชุดข้อมูลสำหรับสอน (training data) ตัวอย่าง (samples/instances) ชุดข้อมูล (data points) or สิ่งของ (objects)” เป็นต้น หมายเหตุ—เนื่องจากแต่ละเรคคอร์ดในชุดข้อมูลที่เป็นอินพุตจะมีแอตริบิวต์หมวดหมู่ข้อมูลแนบอยู่ด้วย

แอตริบิวต์ที่บ่งบอกถึงหมวดหมู่ของข้อมูล คำตอบที่สอนให้คอมพิวเตอร์เรียนรู้

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

n แอททริบิวต์ที่บ่งบอกถึงคุณลักษณะต่าง ๆ ของข้อมูลเรคคอร์ด X นอกจากนั้นเรคคอร์ด X ยังมี ข้อมูล อีก หนึ่ง แอททริบิวต์ ที่ บ่ง บอ ก ถึง ห ม ว ด ห มู่ ข อ ง ขั อ มู ล (class label attribute) โดยแอททริบิวต์หมวดหมู่ข้อมูลจะเป็นข้อมูลแบบไม่ต่อเนื่อง (discrete-valued) โดยชุดข้อมูลที่เป็นอินพุตสำหรับการสร้างตัวจำแนกข้อมูลจะถูกเรียกว่า “ชุดข้อมูลสำหรับสอน (training data) ตัวอย่าง (samples/instances) ชุดข้อมูล (data points) or สิ่งของ (objects)” เป็นต้น หมายเหตุ—เนื่องจากแต่ละเรคคอร์ดในชุดข้อมูลที่เป็นอินพุตจะมีแอททริบิวต์หมวดหมู่ข้อมูลแนบอยู่ด้วย

แอททริบิวต์ที่บ่งบอกถึงหมวดหมู่ของข้อมูล คำตอบที่สอนให้คอมพิวเตอร์เรียนรู้

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

ดังนั้น การจำแนกข้อมูลด้วยข้อมูลลักษณะนี้จะเรียกว่า **การเรียนรู้แบบมีผู้สอน (Supervised learning)** คือ การสร้างตัวจำแนกข้อมูลจะถูกสอนโดยแอตทริบิวต์หมวดหมู่ข้อมูลต่าง ๆ ที่ถูกแนบอยู่ในแต่ละเรคคอร์ดของชุดข้อมูล โดยการเรียนรู้แบบมีผู้สอนจะแตกต่างกับการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning หรือ clustering) ที่จะไม่ทราบถึงหมวดหมู่ของข้อมูล ตัวอย่างเช่น ในการวิเคราะห์ข้อมูลการกู้-ยืมเงินที่ไม่มีหมวดหมู่ข้อมูลที่บ่งบอกว่าการกู้ยืมครั้งหนึ่ง ๆ มีความเสี่ยงหรือไม่ เราจะสามารถวิเคราะห์ข้อมูลได้จากการเชื่อมโยงเรคคอร์ดของการกู้-ยืมเงินที่มีลักษณะใกล้เคียงกันหรือเหมือนกันให้อยู่ในกลุ่มเดียวกัน เป็นต้น สรุป การเรียนแบบไม่มีผู้สอน คือ ไม่ได้ระบุคำตอบไว้ให้

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

ขั้นตอนที่สองของการจำแนกข้อมูล (ดังแสดงในรูปที่ 5.2) จะเป็นการเรียกใช้ตัวจำแนกข้อมูลที่สร้างขึ้นจากขั้นตอนที่หนึ่งเพื่อทำการจำแนกข้อมูล โดยในตอนเริ่มต้น ตัวจำแนกข้อมูลจะถูกทดสอบและประเมินค่าความถูกต้อง (ถ้าเราใช้ชุดข้อมูลสำหรับสอนในการทดสอบตัวจำแนกข้อมูล จะทำให้ความถูกต้องจะมีค่าค่อนข้างสูง เนื่องจากตัวจำแนกข้อมูลที่สร้างขึ้นจะเหมาะกับข้อมูลชุดนั้นเป็นอย่างมาก (**overfit**) แต่ถ้าเราใช้ชุดข้อมูลที่แตกต่างออกไปในการทดสอบ (test set) โดยชุดข้อมูลที่ใช้จะต้องมีแอตทริบิวต์หมวดหมู่ข้อมูลแนบอยู่ด้วย จะทำให้เราทราบค่าความถูกต้องของตัวจำแนกข้อมูลได้) โดยค่าความถูกต้องของตัวจำแนกข้อมูลที่ถูกสร้างขึ้นจะเป็นเปอร์เซ็นต์ของตัวจำแนกข้อมูลที่สามารถจำแนกข้อมูลได้อย่างถูกต้อง (ตัวจำแนกข้อมูลบ่งบอกถึงหมวดหมู่ข้อมูลได้เหมือนกับหมวดหมู่ข้อมูลที่ถูกแนบมากับข้อมูลเรคคอร์ดหนึ่ง ๆ)

ความหมายของการจำแนกและการทำนายข้อมูล(ต่อ)

Classification rules

กระบวนการทดสอบ (Testing)

Test data

<i>name</i>	<i>age</i>	<i>income</i>	<i>loan_decision</i>
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

(b)

New data

(John Henry, middle_aged, low)

Loan decision?

risky

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

เมื่อค่าความถูกต้องของตัวจำแนกข้อมูลมีค่าที่น่าพึงพอใจหรือยอมรับได้ เราจะใช้ตัวจำแนกข้อมูลในการจำแนกหรือบ่งบอกถึงหมวดหมู่ข้อมูลที่เข้ามาใหม่ที่ซึ่งเราไม่ทราบหมวดหมู่ข้อมูลมาก่อน (ข้อมูลที่เข้ามาใหม่จะถูกเรียกว่า **'unknown'** หรือ **'previously unseen'** data) ตัวอย่างเช่น ตัวจำแนกข้อมูลที่ถูกสร้างขึ้นในภาพที่ 5.1 จะถูกใช้เพื่อตัดสินใจการให้กู้-ยืมเงินของเอกสารที่ยื่นเข้ามาใหม่ว่าจะให้กู้-ยืมหรือไม่ เป็นต้น

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

1. Feature , Feature Extraction
2. Attributes
3. Label , Target
4. Prediction
5. Training Dataset , Testing Dataset
6. Split Data , Cross-validation
7. Overfit
8. Unknown

6.2 ความหมายของการจำแนกและการทำนายข้อมูล (ต่อ)

1. Feature , Feature Extraction

6.2 ความหมายของการจำแนกข้อมูล (ต่อ...)

■ การจำแนกข้อมูลมีการนำไปใช้ในหลายสาขา ดังนี้

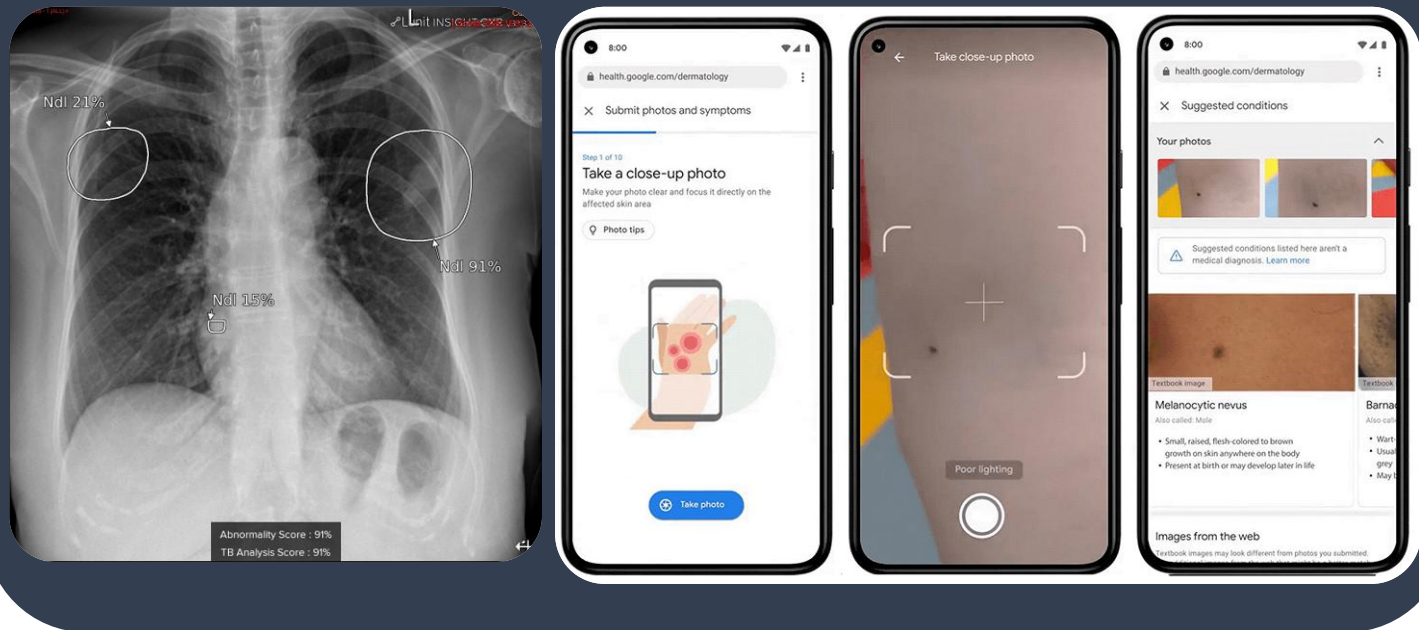
- 1) **การจำแนกภาพ (Image Classification):** การแยกแยะวัตถุหรือสิ่งของในภาพ
- 2) **การจำแนกข้อความ (Text Classification):** การจำแนกเอกสารหรือข้อความออนไลน์ได้แก่การกระทำทางภาษาธรรมชาติ (NLP)
- 3) **การจำแนกออกซัลลาร (Sentiment Analysis):** การระบุความรู้สึกในข้อความ
- 4) **การจำแนกโรค (Disease Classification):** การตรวจวินิจฉัยโรคจากรูปภาพทางการแพทย์
- 5) **การจำแนกการโจมตี (Intrusion Detection):** การตรวจจับการทำผิดของระบบคอมพิวเตอร์

การจำแนกข้อมูลเป็นเครื่องมือที่มีประโยชน์ในการทำนายและทำความเข้าใจข้อมูลในรูปแบบที่ถูกต้องและมีประสิทธิภาพ.

6.2 ความหมายของการจำแนกข้อมูล (ต่อ...)

■ การจำแนกข้อมูลมีการนำไปใช้ในหลายสาขา ดังนี้

1) **การจำแนกภาพ (Image Classification):** การแยกแยะวัตถุหรือสิ่งของในภาพ



6.2.1 การจำแนกภาพ (Image Classification)

■ การจำแนกภาพ (Image Classification): การแยกแยะวัตถุหรือสิ่งของในภาพ

การจำแนกภาพ (Image Classification) เป็นกระบวนการที่ให้โมเดลเรียนรู้และจำแนกภาพในกลุ่มหรือประเภทต่าง ๆ ตามลักษณะหรือลักษณะทางคุณสมบัติของภาพนั้น ๆ ทำให้สามารถแยกแยะวัตถุหรือสิ่งของในภาพได้ตามที่ถูกกำหนดไว้. กระบวนการนี้มีการใช้โมเดลเครื่องประมวลผล (machine learning) หรือปัญญาประดิษฐ์ (artificial intelligence) เพื่อสร้างระบบที่สามารถรู้จำและจำแนกภาพได้ด้วยตนเอง

■ ตัวอย่างการแยกแยะวัตถุหรือสิ่งของในภาพ เช่น

- ภาพที่มีวัตถุหลายประเภทอยู่รวมกัน เช่น ภาพป่าไม้ ภาพท้องถนน เป็นต้น เราสามารถแยกแยะวัตถุในภาพตามรูปร่าง สี ขนาด ลักษณะพื้นผิว หรือลักษณะการเคลื่อนไหว เป็นต้น
- ภาพที่มีวัตถุประเภทเดียวกันอยู่รวมกัน เช่น ภาพฝูงนก ภาพฝูงปลา เป็นต้น เราสามารถแยกแยะวัตถุในภาพตามลักษณะของวัตถุนั้นๆ เช่น รูปร่าง สี ขนาด เป็นต้น
- ภาพที่มีวัตถุซ้อนทับกัน เช่น ภาพถนนที่มีรถยนต์วิ่งอยู่ เป็นต้น เราสามารถแยกแยะวัตถุในภาพได้โดยพิจารณาจากรูปร่าง สี ขนาด หรือลักษณะพื้นผิวของวัตถุนั้น ๆ

6.2.2 การจำแนกข้อความ (Text Classification)

- **การจำแนกข้อความ (Text Classification):** การจำแนกเอกสารหรือข้อความออนไลน์ได้แก่การกระทำทางภาษาศาสตร์ (NLP)

การจำแนกข้อความ (Text Classification) เป็นกระบวนการที่ให้โมเดลเรียนรู้และจำแนกข้อความลงในกลุ่มหรือประเภทต่าง ๆ ตามลักษณะหรือลักษณะทางคุณสมบัติที่กำหนดไว้. กระบวนการนี้มีการใช้เทคนิคทางภาษาศาสตร์ (Natural Language Processing - NLP) เพื่อทำให้โมเดลสามารถเข้าใจและวิเคราะห์ข้อความได้อย่างถูกต้อง

- **การจำแนกข้อความมีการนำไปใช้ในหลายสาขา เช่น:**

- การจำแนกข้อความเป็นกลุ่ม (Text Categorization): การจำแนกข้อความลงในหมวดหมู่หรือกลุ่มต่าง ๆ เช่น ข่าว, บทวิจารณ์, หรือ เรื่องเศรษฐกิจ
- การวิเคราะห์ความรู้สึก (Sentiment Analysis): การระบุและจำแนกความรู้สึกในข้อความว่าเป็นบวก, ลบ, หรือ นิวตริล
- การตรวจจับการทำธุรกรรม (Business Intelligence): การจำแนกข้อความที่เกี่ยวข้องกับธุรกิจ เช่น ความต้องการของลูกค้าหรือ การเรียกร้อง
- การตอบคำถาม (Question Answering): การจำแนกข้อความเพื่อตอบคำถามที่เกี่ยวข้อง

6.2.3 การจำแนกออกซัลลาร (Sentiment Analysis)

■ การจำแนกออกซัลลาร (Sentiment Analysis): การระบุความรู้สึกในข้อความ

การจำแนกออกซัลลาร (Sentiment Analysis) เป็นกระบวนการที่ให้โมเดลเรียนรู้และจำแนกความรู้สึกหรือทัศนคติที่ปรากฏในข้อความ เช่น ว่าข้อความนั้นมีเนื้อหาที่เชิงบวก, เชิงลบ, หรือเป็นเนื้อหาที่เป็นกลาง ๆ อยู่ในแนวไหน. การจำแนกออกซัลลารมักถูกนำไปใช้ในการวิเคราะห์ความคิดเห็นของลูกค้า, การตรวจสอบทัศนคติในโซเชียลมีเดีย, การวิเคราะห์รีวิวของผู้ใช้, และในตลาดอื่น ๆ ที่เกี่ยวข้องกับการทำนายความรู้สึก.

■ การจำแนกออกซัลลารมักถูกนำไปใช้ในหลายแวดวง เช่น:

- การวิเคราะห์ความคิดเห็นลูกค้า (Customer Reviews Analysis): การตรวจสอบความรู้สึกของลูกค้าต่อผลิตภัณฑ์หรือบริการ.
- การวิเคราะห์โซเชียลมีเดีย (Social Media Analysis): การตรวจสอบทัศนคติที่ปรากฏในโพสต์หรือความคิดเห็นบนโซเชียลมีเดีย.

6.2.3 การจำแนกออกซัลลาตรา (Sentiment Analysis) (ต่อ)

■ การจำแนกออกซัลลาตรามักถูกนำไปใช้ในหลายแวดวง เช่น:

- การคัดกรองข่าว (News Sentiment Analysis): การวิเคราะห์ความรู้สึกที่ปรากฏในข่าว.
- การจัดการความเครียด (Stress Management): การตรวจสอบความรู้สึกของบุคคลในสถานการณ์เฉพาะ เช่น การทำงานหรือการศึกษา

การจำแนกออกซัลลาตรามีความสำคัญในการทำให้ชีอาร์ที่เกี่ยวข้องกับความรู้สึกและทัศนคติเพื่อเข้าใจพฤติกรรมและความพึงพอใจของผู้ใช้หรือลูกค้า

6.3 ความหมายของการทำนายข้อมูล

■ ความหมายของการทำนายข้อมูล

การทำนายข้อมูล (Prediction) เป็นกระบวนการทางวิทยาการข้อมูล (Data Science) หรือการเรียนรู้เชิงลึกน้อย (Machine Learning) ที่มุ่งเน้นการสร้างแบบจำลอง (Model) ที่สามารถทำนายค่าหรือผลลัพธ์ที่คาดหวังจากข้อมูลที่มีอยู่ การทำนายนั้นมีลักษณะการทำงานในลำดับดังนี้:

ตัวอย่างงานที่ใช้การทำนายรวมถึงการทำนายราคาทอง, การทำนายยอดขายสินค้า, หรือการทำนายผลกระทบทางสุขภาพจากตัวแปรต่าง ๆ การทำนายมีการประยุกต์ใช้ในหลายด้าน เช่น การธุรกิจ, การแพทย์, และอื่น ๆ ที่ต้องการการวิเคราะห์และการทำนายในข้อมูลที่มีความซับซ้อน.

6.3 การทำนาย (Prediction)

กระบวนการทางคอมพิวเตอร์ (algorithm) ที่ใช้สำหรับการพยากรณ์หรือทำนายผลลัพธ์จากข้อมูลทั่วไปสามารถแบ่งออกเป็น 2 ประเภทใหญ่ได้ดังนี้

1. **Supervised Learning (การเรียนรู้แบบมีผู้สอน):** ในกระบวนการ Supervised Learning, โมเดลถูกฝึกด้วยข้อมูลที่มีคำตอบ (label) และต้องการทำนายผลลัพธ์หรือคำตอบเมื่อได้รับข้อมูลใหม่

ตัวอย่าง Algorithms

- Linear Regression: ใช้สำหรับการพยากรณ์ค่าต่อเนื่อง (continuous values) เช่น ราคา, อุณหภูมิ, หรือผลผลิต
- Logistic Regression: ใช้สำหรับการพยากรณ์ค่าแบบไบนารี (binary values) เช่น การตกลงหรือไม่ตกลง, การชนะหรือแพ้
- **Decision Trees:** สร้างโมเดลที่เป็นต้นไม้ที่ช่วยในการตัดสินใจบนข้อมูลแบบมีลำดับ
- Support Vector Machines (SVM): ใช้สำหรับการจัดกลุ่มข้อมูล (classification) หรือการพยากรณ์ค่าต่อเนื่อง
- Neural Networks: โมเดลทางคอมพิวเตอร์ที่จำลองการทำงานของสมองมนุษย์

6.3 การทำนาย (Prediction)

2. Unsupervised Learning (การเรียนรู้แบบไม่มีผู้สอน):

- คำอธิบาย: ในกระบวนการ Unsupervised Learning, โมเดลถูกฝึกด้วยข้อมูลที่ไม่มีคำตอบ (label) และเน้นการค้นหาโครงสร้างหรือลำดับที่ซ่อนอยู่ในข้อมูล.

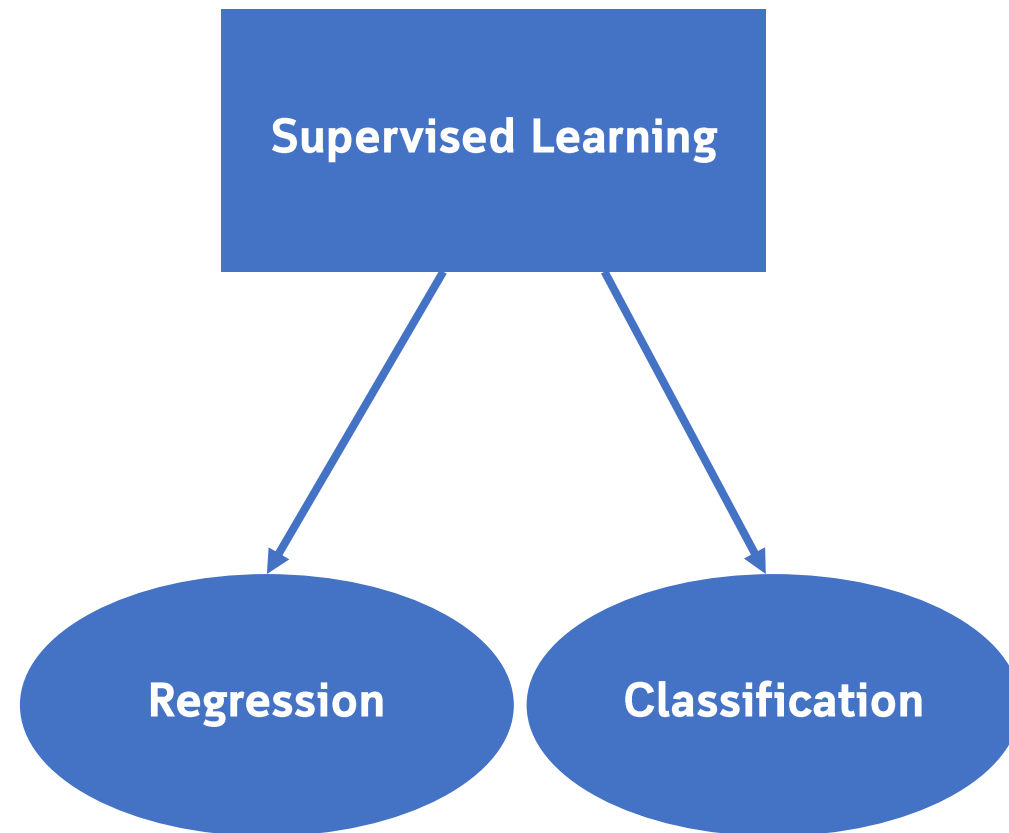
ตัวอย่าง Algorithms:

- K-Means Clustering: ใช้สำหรับการจัดกลุ่มข้อมูลออกเป็นกลุ่มหลาย ๆ กลุ่ม.
- Hierarchical Clustering: ใช้สำหรับการจัดกลุ่มข้อมูลออกเป็นลำดับแบบต้นไม้.
- Principal Component Analysis (PCA):** ใช้สำหรับลดขนาดของข้อมูลและค้นหาความสัมพันธ์ที่ซ่อนอยู่.
- Autoencoders: โมเดลทางคอมพิวเตอร์ที่ใช้ในการลดมิติของข้อมูล (dimensionality reduction) หรือสร้างรูปภาพที่เป็นความหมาย (semantic images) จากข้อมูล.

ทั้งสองประเภทนี้มีการใช้งานอย่างกว้างขวางในการแก้ปัญหาทางคอมพิวเตอร์ที่เกี่ยวข้องกับการทำนาย, จัดกลุ่ม, และค้นหาความสัมพันธ์ในข้อมูล.

6.3.1 การเรียนรู้โดยมีผู้สอน (Supervised Learning)

การเรียนรู้แบบมีผู้สอน เป็นศาสตร์แขนงหนึ่งใน AI หรือปัญญาประดิษฐ์ ภายใต้หัวข้อ Machine Learning ที่กำลังเป็นที่นิยมในการศึกษาและวิจัยกันในปัจจุบัน เนื่องจากทำได้ง่าย(Tools) ต้นทุนต่ำ เพียงใช้คอมพิวเตอร์เครื่องเดียวก็สามารถศึกษาและทำงานจนเห็นผลได้เลย โดยศาสตร์แขนงนี้มีมานานมากแล้ว ตั้งแต่ปี 1959 ถูกเสนอโดย Arthur Samuel เป็นนักวิทยาศาสตร์คอมพิวเตอร์ชาวอเมริกันผู้เชี่ยวชาญด้านเกมคอมพิวเตอร์ ปัญญาประดิษฐ์ และการเรียนรู้ของเครื่อง โปรแกรม แต่ด้วยเทคโนโลยีหรือระบบประมวลผลในตอนนั้นยังล้าสมัยอยู่ ทำให้ยังไม่เป็นที่นิยม ผิดกับในปัจจุบัน



ภาพที่ 6.3 การเรียนรู้แบบมีผู้สอน

6.3.1 การเรียนรู้โดยมีผู้สอน (Supervised Learning)

การเรียนรู้โดยมีผู้สอน (Supervised Learning)

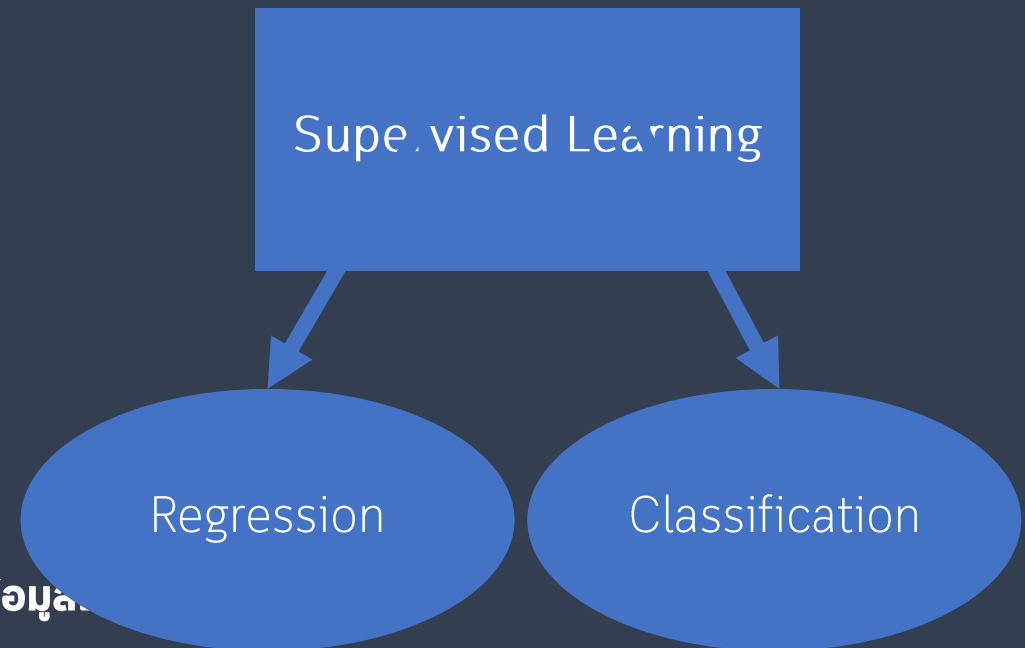
การเรียนรู้โดยมีผู้สอน เป็นวิธีการเรียนรู้แบบหนึ่งใน Machine Learning ที่ใช้ชุดข้อมูลที่มีทั้ง ข้อมูลเข้า (Input) และ ข้อมูลออก (Output) ที่ถูกต้อง เพื่อฝึกให้โมเดลสามารถเรียนรู้ความสัมพันธ์ระหว่างข้อมูลเหล่านี้ และสามารถทำนายผลลัพธ์สำหรับ ข้อมูลใหม่ได้

ตัวอย่าง

- การจัดหมวดหมู่อีเมล (Spam/Not Spam)
- การคาดการณ์ราคาบ้าน
- การวิเคราะห์ภาพ (แมวหรือสุนัข)
- การแปลภาษา

ประเภทของการเรียนรู้โดยมีผู้สอน

- **การจัดประเภท (Classification)** : โมเดลจะเรียนรู้ที่จะจัดหมวดหมู่ข้อมูล
- **การถดถอย (Regression)** : โมเดลจะเรียนรู้ที่จะคาดการณ์ค่าตัวเลขต่อเนื่อง



6.3.2 ขั้นตอนหลักของการเรียนรู้โดยมีผู้สอน

ขั้นตอนหลักของการเรียนรู้โดยมีผู้สอนมีดังนี้:

1. ข้อมูลการฝึก (Training Data): จะต้องมีข้อมูลที่มีคำตอบหรือป้ายกำกับแล้วในการฝึกระบบ ซึ่งประกอบไปด้วยคุณลักษณะ (features) และคำตอบที่ถูกต้อง (label) เช่น ภาพของแมวที่มีป้ายกำกับว่าเป็นแมวหรือไม่
2. โมเดลการเรียนรู้ (Learning Model): ต้องกำหนดโมเดลที่จะใช้ในการเรียนรู้ ซึ่งเป็นโครงสร้างทางคณิตศาสตร์ที่สามารถเรียนรู้จากข้อมูลได้ เช่น Neural Networks, Support Vector Machines, Decision Trees, หรืออื่นๆ
3. การฝึก (Training): ในขั้นตอนนี้, โมเดลจะถูกฝึกโดยใช้ข้อมูลฝึก โดยระบบจะปรับพารามิเตอร์ของโมเดลเพื่อให้มีความสามารถในการทำนายคำตอบที่ถูกต้อง
4. การทดสอบ (Testing): หลังจาก that โมเดลถูกฝึกเสร็จ, ต้องทดสอบประสิทธิภาพของโมเดลด้วยข้อมูลทดสอบที่ไม่ได้ถูกใช้ในขั้นตอนการฝึก นี้เพื่อประเมินว่าโมเดลมีความสามารถทำนายได้ดีแค่ไหน
5. ประเมินและปรับปรุง (Evaluation and Fine-Tuning): หากโมเดลมีประสิทธิภาพไม่ดีพอ, จะต้องปรับปรุงโมเดลโดยใช้ข้อมูลการทดสอบ เพื่อให้มีประสิทธิภาพที่ดีในการทำนายคำตอบ

การเรียนรู้โดยมีผู้สอนมักถูกนำมาใช้ในงานต่างๆ เช่น การจำแนกประเภทของภาพ, การจำแนกข้อความ, หรือการทำนายราคาของหุ้น เป็นต้น

6.3.3 ตัวอย่างข้อมูลการฝึก (Training Data)

เพื่อให้คุณเข้าใจมากขึ้น, นี่คือตัวอย่างข้อมูลการฝึก (Training Data) สำหรับงานการจำแนกภาพของสัตว์ว่าเป็นแมวหรือหมา:

คุณลักษณะ (Features): ในที่นี้, คุณลักษณะของภาพเป็นเส้นตรงของพิกเซล (pixels) ในภาพที่ถูกเข้ารหัสเป็นตัวเลข. เช่น, ในกรณีของภาพขนาด 64x64 pixels, คุณลักษณะจะเป็นอาเรย์ขนาด 64x64

คำตอบ (Labels): คำตอบหรือป้ายกำกับแสดงว่าภาพนั้นเป็นแมว (cat) หรือหมา (dog). เราให้ "1" แทนแมวและ "0" แทนหมา.

ตัวอย่างข้อมูลฝึก

คุณลักษณะ (Features)	คำตอบ (Label)
[0.2, 0.4, 0.1, ..., 0.8] (ของแมว)	1
[0.5, 0.7, 0.3, ..., 0.2] (ของหมา)	0
[0.1, 0.8, 0.6, ..., 0.5] (ของแมว)	1
[0.6, 0.2, 0.7, ..., 0.9] (ของหมา)	0
[0.9, 0.5, 0.3, ..., 0.6] (ของแมว)	1

ในตารางนี้, คุณลักษณะ (Features) คือข้อมูลที่ใช้ในการฝึก, และคำตอบ (Label) คือผลลัพธ์ที่เราต้องการให้โมเดลทำนาย. แต่ละแถวแสดงถึงภาพและคำตอบที่เกี่ยวข้อง. ข้อมูลเหล่านี้จะถูกนำมาใช้ในกระบวนการฝึกโมเดลในขั้นตอนถัดไป

6.3 ตัวอย่างการทำนายข้อมูล

■ ตัวอย่างการทำนายข้อมูล:

- **การทำนายการขาย:** ในธุรกิจ, การทำนายการขายสินค้าหรือบริการในอนาคตจากข้อมูลการขายในอดีตนี้สามารถช่วยให้ธุรกิจทำการวางแผนการผลิตหรือการโปรโมชั่นสินค้าได้อย่างมีประสิทธิภาพ
- **การทำนายความรู้สึกจากข้อความ:** ในการวิเคราะห์ข้อมูลทางภาษาธรรมชาติ การทำนายความรู้สึกจากข้อความที่เขียนโดยผู้ใช้บนโซเชียลมีเดียหรือเว็บไซต์ นี้สามารถช่วยธุรกิจหรือบุคคลทั่วไปเข้าใจความคิดเห็นของผู้ใช้
- **การทำนายโรคจากรูปภาพทางการแพทย์:** ในการวิเคราะห์รูปภาพทางการแพทย์, การทำนายว่าภาพนั้นๆ มีลักษณะของโรคหรือไม่ นี้สามารถช่วยให้การตรวจวินิจฉัยเป็นไปอย่างรวดเร็วและมีประสิทธิภาพ

6.4 ข้อแตกต่างระหว่าง การทำนายข้อมูล และการจำแนกข้อมูล

การทำนายข้อมูล (Prediction) และการจำแนกข้อมูล (Classification) เป็นกระบวนการทางวิทยาการข้อมูล (Data Science) ที่ใช้เทคนิคต่าง ๆ เพื่อประมวลผลข้อมูล โดยมีความแตกต่างระหว่างทั้งสอง ดังนี้

■ ความเป้าหมาย:

- การทำนายข้อมูล (Prediction): มีจุดมุ่งหมายเพื่อทำนายค่าหรือผลลัพธ์ที่ยังไม่ทราบของข้อมูลใหม่, ซึ่งอาจเป็นตัวเลขหรือค่าที่ไม่สามารถจำแนกได้เป็นกลุ่ม.
- การจำแนกข้อมูล (Classification): มีจุดมุ่งหมายเพื่อจำแนกข้อมูลลงในกลุ่มหรือประเภทต่าง ๆ ตามลักษณะหรือลักษณะทางคุณสมบัติที่กำหนดไว้.

■ ผลลัพธ์:

- การทำนายข้อมูล (Prediction): ผลลัพธ์มักเป็นค่าที่ทำนายมาจากโมเดล, ซึ่งอาจเป็นตัวเลขหรือค่าที่ไม่ได้จำแนกเป็นกลุ่ม.
- การจำแนกข้อมูล (Classification): ผลลัพธ์คือการจำแนกข้อมูลลงในกลุ่มหรือประเภทที่กำหนดไว้, เช่น หมวดหมู่ A, B, C.

6.4 ข้อแตกต่างระหว่าง การทำนายข้อมูล และการจำแนกข้อมูล (ต่อ...)

■ ประเภทของปัญหา:

- การทำนายข้อมูล (Prediction): สามารถเป็นปัญหาที่เชิงตัวเลข (regression) หรือที่เชิงการจำแนก (classification) ขึ้นอยู่กับลักษณะของข้อมูล.
- การจำแนกข้อมูล (Classification): เป็นการแยกข้อมูลออกเป็นกลุ่มหรือประเภทที่แตกต่างกัน.

■ ตัวอย่างการใช้:

- การทำนายข้อมูล (Prediction): การทำนายยอดขายของสินค้าในอนาคต, การทำนายราคาหุ้น, หรือการทำนายอุณหภูมิในวันพรุ่งนี้.
- การจำแนกข้อมูล (Classification): การจำแนกออกซาลาตรา (sentiment analysis) ของข้อความ, การจำแนกอยู่ในกลุ่มกลาง, สูง, หรือต่ำ, หรือการจำแนกภาพว่าเป็นแมวหรือหมา.

6.5 กระบวนการของการจำแนกข้อมูลมีขั้นตอนหลักดังนี้

■ กระบวนการของการจำแนกข้อมูลมีขั้นตอนหลักดังนี้

1. **การเตรียมข้อมูล (Data preparation)** : เตรียมข้อมูลให้พร้อมสำหรับการวิเคราะห์ เช่น ลบข้อมูลที่ไม่จำเป็น ตรวจสอบความถูกต้องของข้อมูล เป็นต้น
2. **การเลือกโมเดล (Model selection)** : เลือกโมเดลการเรียนรู้ของเครื่องที่เหมาะสมสำหรับงานที่ต้องการทำ
3. **การฝึกอบรมโมเดล (Model training)** : ฝึกอบรมโมเดลโดยใช้ข้อมูลการฝึกอบรมที่มีป้ายกำกับ
4. **การประเมินโมเดล (Model evaluation)** : ประเมินประสิทธิภาพของโมเดลโดยใช้ข้อมูลทดสอบ
5. **การนำไปใช้งาน (Model deployment)** : นำโมเดลไปใช้ทำนายข้อมูลใหม่

6.6 เทคนิคที่ใช้ในการจำแนกข้อมูล (Classification Techniques)

อัลกอริทึมการเรียนรู้ของเครื่องที่นิยมใช้สำหรับการจำแนกข้อมูล ได้แก่

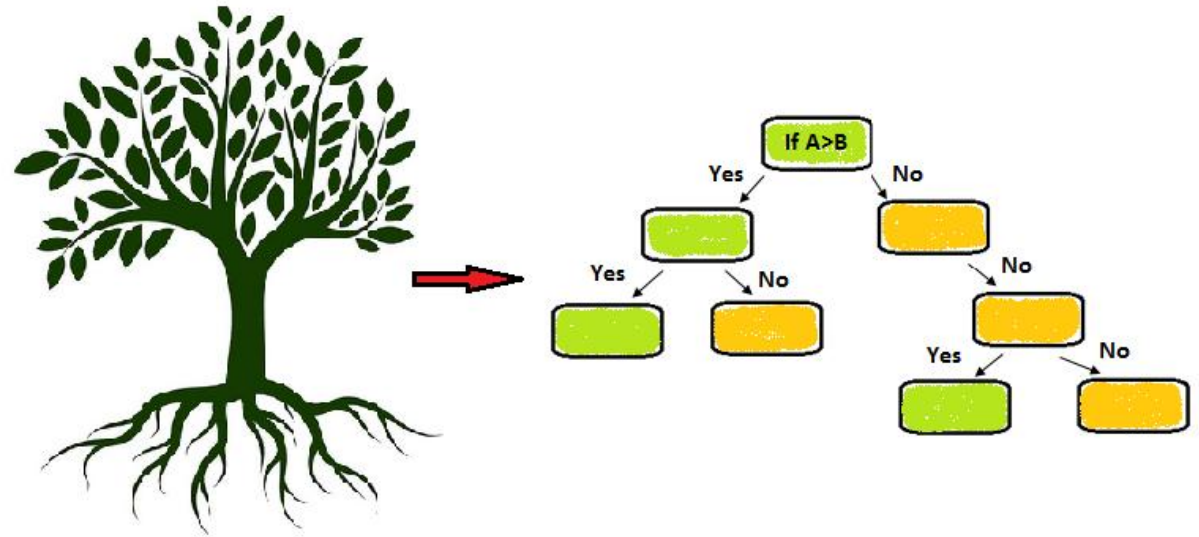
1. **Decision tree** : สร้างโมเดลจำแนกประเภทในรูปแบบของต้นไม้การตัดสินใจ
2. **Support vector machine** : สร้างโมเดลจำแนกประเภทโดยใช้เส้นแบ่งที่แบ่งข้อมูลออกเป็นสองกลุ่มอย่างชัดเจน
3. **Logistic regression** : สร้างโมเดลจำแนกประเภทโดยใช้สมการโลจิสติก
4. **Naive Bayes** : สร้างโมเดลจำแนกประเภทโดยใช้สมการเบย์
5. **K-nearest neighbors** : สร้างโมเดลจำแนกประเภทโดยพิจารณาจากข้อมูลตัวอย่างที่ใกล้เคียงที่สุด
6. **Neural Networks (Deep Learning)**: โครงข่ายประสาทเทียมที่มีหลายชั้น, ทำให้สามารถเรียนรู้ความซับซ้อนได้

■ ฯลฯ

6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

Decision Tree (ต้นไม้ตัดสินใจ) เป็นวิธีการทำนายหรือจำแนกข้อมูลที่มีโครงสร้างเป็นต้นไม้ทำนายซึ่งประกอบด้วย การตัดสินใจตามลำดับของเงื่อนไข (conditions) ในบริบทของการใช้ Decision Tree ในเหมืองข้อมูล (Data Mining), มีการนำ Decision Tree มาใช้เพื่อทำนายหรือจำแนกข้อมูลเพื่อค้นหาความรู้หรือโครงสร้างที่ซ่อนอยู่ในข้อมูล

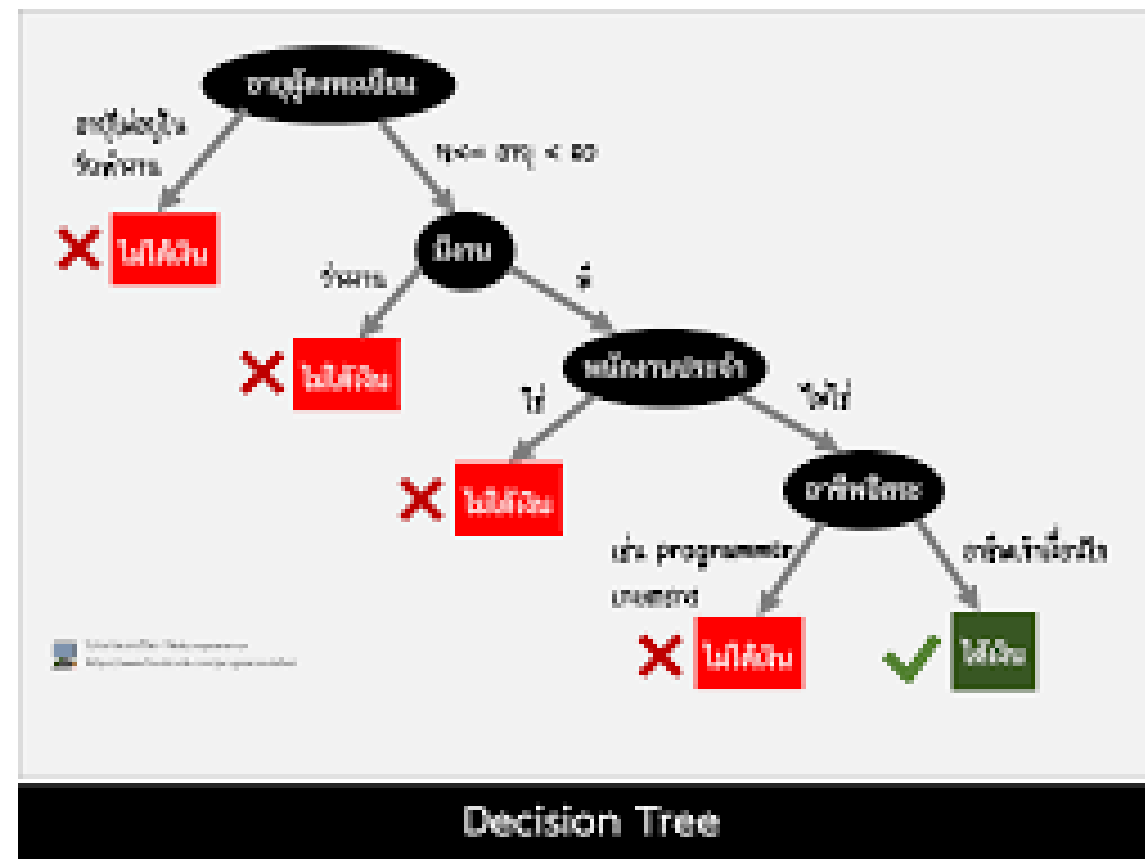


6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

■ ส่วนประกอบของต้นไม้ตัดสินใจประกอบไปด้วย:

1. โหนด (Node): โหนดคือจุดที่ใช้ในการตัดสินใจในต้นไม้. โหนดแบ่งข้อมูลออกเป็นกลุ่มย่อย ๆ โดยใช้ลักษณะหนึ่ง ๆ
2. เส้นกิ่ง (Branch): เส้นกิ่งเชื่อมระหว่างโหนดและแสดงถึงการตัดสินใจที่ทำขึ้น.
3. ลักษณะ (Feature): ลักษณะคือคุณสมบัติหรือตัวแปรที่ใช้ในการแบ่งข้อมูล เมื่อต้นไม้ตัดสินใจต้องการตัดสินใจ, จะเลือกลักษณะที่สามารถแบ่งข้อมูลได้ดีที่สุด
4. น้ำหนัก (Weight): น้ำหนักบนเส้นกิ่งแสดงถึงความสำคัญของแต่ละสาขาหรือการตัดสินใจ



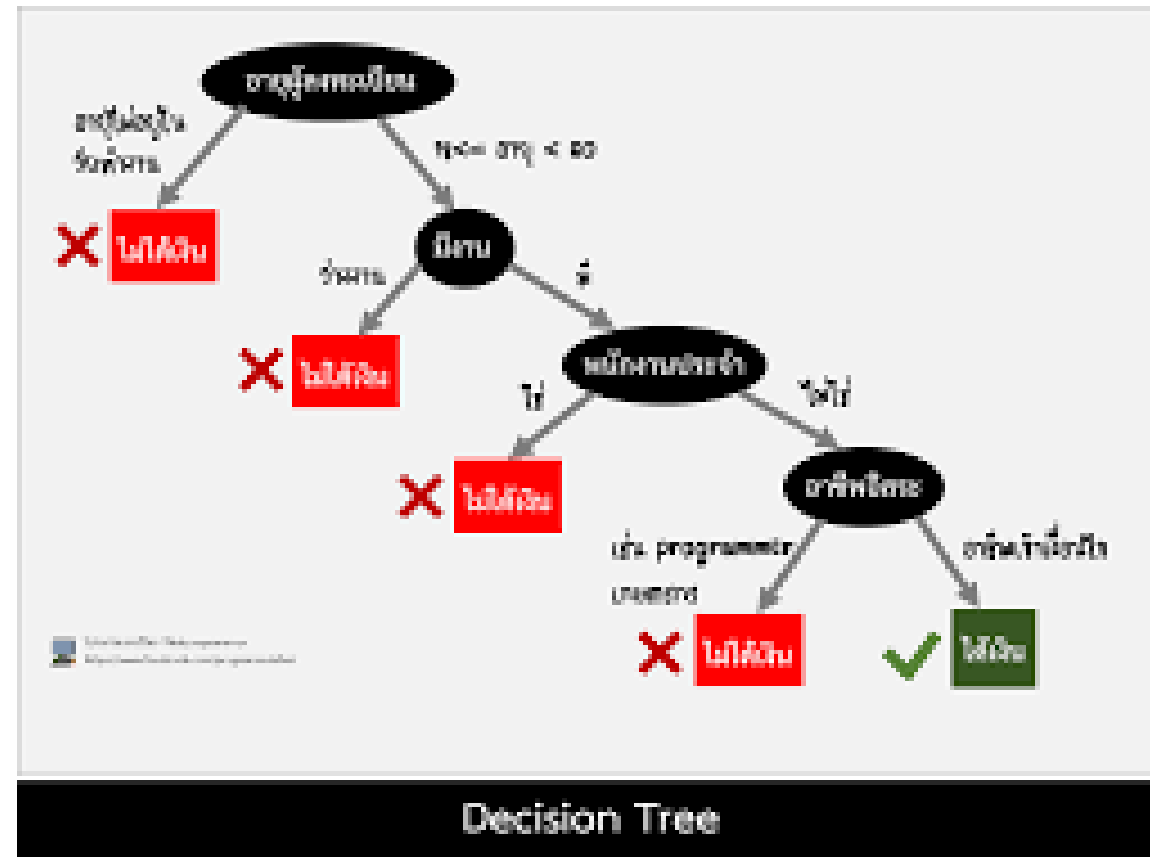
<https://algoaddict.wordpress.com/2022/06/27/จากต้นไม้แห่งการตัดสินใจ/>

6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

■ ส่วนประกอบของต้นไม้ตัดสินใจประกอบไปด้วย:

- ใบ (Leaf): ใบของต้นไม้แสดงถึงผลลัพธ์หรือการตัดสินใจที่สุดท้ายหลังจากที่ได้แบ่งข้อมูลตามลักษณะต่าง ๆ ในต้นไม้
- Root Node (โหนดราก): โหนดที่อยู่ทางด้านบนสุดของต้นไม้ แทนด้วยลักษณะที่สามารถแบ่งข้อมูลได้ดีที่สุด
- Entropy (เ็นโทรปี): การวัดความไม่แน่นอนของข้อมูลในโหนด โดยที่ค่าเ็นโทรปีต่ำแปลว่าข้อมูลมีการแบ่งแยกได้ดี.
- Information Gain (กำไรข้อมูล): การวัดการลดเ็นโทรปีหรือความไม่แน่นอนที่ลดลงหลังจากที่แบ่งข้อมูลตามลักษณะ.



6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

6.6.1 ต้นไม้ตัดสินใจ (Decision tree)

Decision Tree (ต้นไม้ตัดสินใจ) เป็นวิธีการทำนายหรือจำแนกข้อมูลที่มีโครงสร้างเป็นต้นไม้ทำนายซึ่งประกอบด้วย การตัดสินใจตามลำดับของเงื่อนไข (conditions) ในบริบทของการใช้ Decision Tree ในเหมืองข้อมูล (Data Mining), มีการนำ Decision Tree มาใช้เพื่อทำนายหรือจำแนกข้อมูลเพื่อค้นหาความรู้หรือโครงสร้างที่ซ่อนอยู่ในข้อมูล

Decision Tree ในเหมืองข้อมูลมักถูกนำมาใช้ในการค้นหาความรู้ที่อาจซ่อนอยู่ในข้อมูลขนาดใหญ่ โดยทำนายหรือจำแนกปรากฏการณ์ต่าง ๆ ภายในข้อมูล เช่น:

- **การทำนายการซื้อสินค้า:** โดยใช้ข้อมูลการซื้อและพฤติกรรมของลูกค้า
- **การจำแนกโรคทางการแพทย์:** โดยใช้ข้อมูลทางการแพทย์เพื่อทำนายว่าผู้ป่วยมีโรคใด
- **การทำนายความเสี่ยงในการกู้ยืมเงิน:** โดยใช้ข้อมูลการใช้บัตรเครดิตและประวัติการชำระเงิน

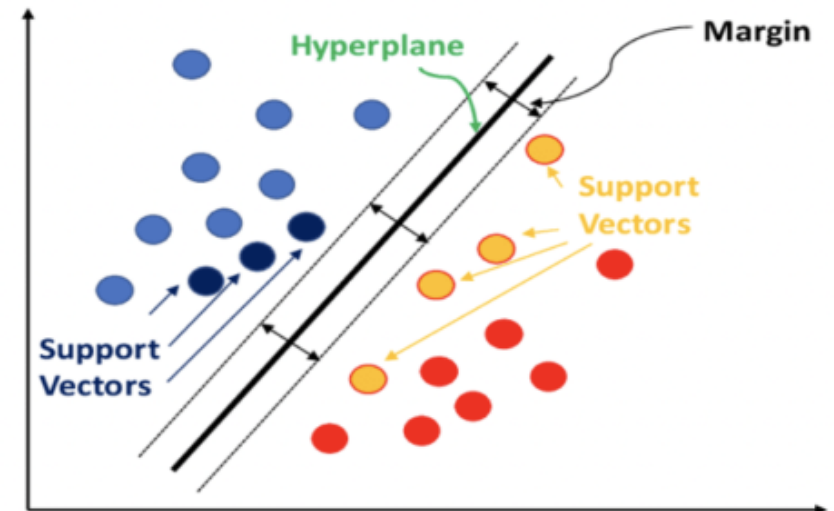
Decision Tree มีข้อได้เปรียบในการอธิบายและตีความผลลัพธ์ของการทำนายหรือจำแนกข้อมูลได้ง่าย และให้ความโปร่งใสในกระบวนการตัดสินใจ อย่างไรก็ตาม, การใช้ Decision Tree อาจต้องพิจารณาปัญหาของการซับซ้อนสูงชัน หรือปัญหาที่มีจำนวนตัวอย่างน้อยได้ทำให้ได้ผลลัพธ์ที่ไม่แม่นยำ

6.6.2 Support Vector Machine (SVM)

6.6.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) เป็นอัลกอริทึมในเศรษฐศาสตร์ข้อมูลที่ใช้ในการจำแนกข้อมูลและการทำนาย SVM ถูกใช้แก้ปัญหาที่ต้องการจำแนกข้อมูลออกเป็นกลุ่มหรือประเภทต่างๆ โดยการสร้างเส้นตามข้อมูลที่แบ่งกลุ่มออกจากกันในทางเชิงเวกเตอร์ในพื้นที่หลาย ๆ มิติ

WHAT IS A
**SUPPORT
VECTOR
MACHINE?**



ที่มา : <https://datatron.com/what-is-a-support-vector-machine/>

6.6.2 Support Vector Machine (SVM)

6.6.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) เป็นอัลกอริทึมในเศรษฐศาสตร์ข้อมูลที่ใช้ในการจำแนกข้อมูลและการทำนาย SVM ถูกใช้แก้ปัญหาที่ต้องการจำแนกข้อมูลออกเป็นกลุ่มหรือประเภทต่าง ๆ โดยการสร้างเส้นตามข้อมูลที่แบ่งกลุ่มออกจากกันในทางเชิงเวกเตอร์ในพื้นที่หลาย ๆ มิติ.

Support vector machine (SVM) เป็นอัลกอริทึมการเรียนรู้ของเครื่องประเภท supervised learning ที่ใช้สำหรับปัญหาการจำแนกประเภท (classification) และปัญหาการหาค่าความถดถอย (regression) SVM ทำงานโดยการสร้างเส้นแบ่งระหว่างคลาสต่าง ๆ ในข้อมูล เส้นแบ่งนี้เรียกว่า hyperplane และจุดที่อยู่ใกล้ hyperplane มากที่สุดเรียกว่า support vectors

ในยุคของเมืองข้อมูล (data city) ข้อมูลมีปริมาณมหาศาลและมีความซับซ้อนสูง SVM จึงเป็นหนึ่งในอัลกอริทึมการเรียนรู้ของเครื่องที่ได้รับความนิยมมากที่สุดสำหรับปัญหาการประมวลผลข้อมูลขนาดใหญ่ เนื่องจาก SVM มีความสามารถในการเรียนรู้คุณสมบัติที่ซับซ้อนของข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ

6.6.3 การถดถอยโลจิสติก (Logistic regression) (ต่อ...)

6.6.3 การถดถอยโลจิสติก (Logistic regression)

การถดถอยโลจิสติก (Logistic regression) เป็นเทคนิคการวิเคราะห์ข้อมูลเชิงพหุ (Multiple regression) ที่ใช้เพื่อประมาณค่าความน่าจะเป็นของการเกิดเหตุการณ์หนึ่งขึ้น โดยเหตุการณ์นั้นจะมีเพียงสองสถานะเท่านั้น เช่น เกิดหรือไม่เกิด เสี่ยงหรือไม่เสี่ยง เป็นต้น

การถดถอยโลจิสติกเป็นเทคนิคที่ได้รับความนิยมอย่างมากในงานเหมืองข้อมูล เนื่องจากสามารถประยุกต์ใช้กับปัญหาการจำแนกประเภท (Classification) ได้หลากหลาย เช่น

- การจำแนกประเภทลูกค้าว่าจะเป็นลูกค้าที่มีแนวโน้มจะซื้อสินค้าหรือไม่
- การจำแนกประเภทโรคว่าเป็นโรคร้ายแรงหรือไม่
- การจำแนกประเภทข้อความว่าเป็นสแปมหรือไม่
- การจำแนกประเภทเว็บไซต์ว่ามีความน่าเชื่อถือหรือไม่

การถดถอยโลจิสติกทำงานโดยสร้างสมการเชิงเส้น (Linear equation) ขึ้นมา ซึ่งสมการนี้สามารถอธิบายความสัมพันธ์ระหว่างตัวแปรตาม (Dependent variable) ซึ่งคือตัวแปรที่ต้องการทำนาย และตัวแปรอิสระ (Independent variable) ซึ่งคือตัวแปรที่ใช้ในการทำนาย

6.6.3 การถดถอยโลจิสติก (Logistic regression) (ต่อ...)

■ ลักษณะสำคัญของ Logistic Regression ในเหมืองข้อมูล:

- การจำแนกข้อมูล (Classification): Logistic Regression มักถูกนำมาใช้ในงานจำแนกข้อมูล, เช่น การทำนายการซื้อสินค้า (1 หรือ 0), การจำแนกหมวดหมู่ของข้อมูล, หรือการจำแนกความน่าจะเป็น
- การทำนายความน่าจะเป็น (Probability): Logistic Regression สามารถให้ค่าความน่าจะเป็นของเหตุการณ์ในรูปแบบของค่าระหว่าง 0 ถึง 1
- ง่ายต่อการอธิบาย: Logistic Regression เป็นอัลกอริทึมที่ง่ายต่อการอธิบายและทำความเข้าใจ, เพราะผลลัพธ์มีการแปลงเป็นความน่าจะเป็นที่อยู่ในช่วงระหว่าง 0 ถึง 1

ปรับแต่งขั้นตอน: สามารถปรับแต่งการทำนายได้ด้วยการเพิ่มหรือลดตัวแปรที่ถูกนำมาในโมเดล

6.6.4 Naive Bayes

6.6.4 Naive Bayes

Naive Bayes เป็นอัลกอริทึมทางสถิติที่ใช้ในการจำแนกประเภทข้อมูล โดยพื้นฐาน Naive Bayes ใช้ทฤษฎีที่เรียกว่า Bayes' Theorem เพื่อทำนายความน่าจะเป็นของการเป็นสมาชิกในกลุ่มหรือประเภทต่าง ๆ ขึ้นอยู่กับลักษณะของข้อมูลที่ให้มา.

Naive Bayes

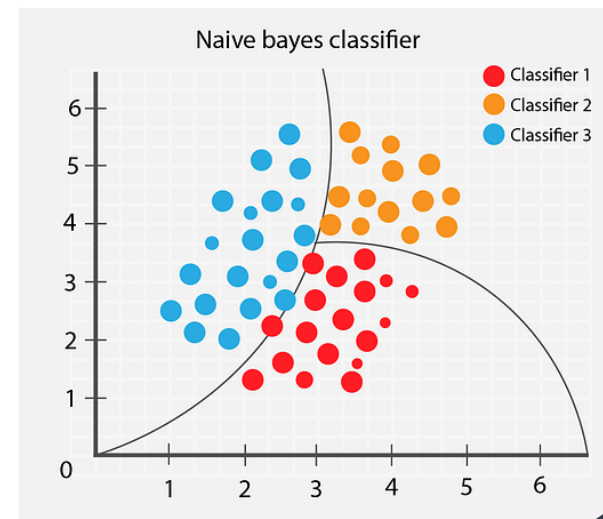


In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$



6.6.4 Naive Bayes

6.6.4 Naive Bayes

Naive Bayes เป็นอัลกอริทึมทางเลขสถิติที่มักถูกนำมาใช้ในการทำเหมืองข้อมูล (Data Mining) และการจำแนกข้อมูล. นับตั้งแต่การทำนายการเลือกตัวอย่างที่มีป้ายกำกับ (supervised learning) ไปจนถึงการจำแนกหรือการกระจายข้อมูลที่ไม่มีป้ายกำกับ (unsupervised learning).

ลักษณะสำคัญของ Naive Bayes ในการทำเหมืองข้อมูล:

- **ง่ายต่อการทำนาย:** Naive Bayes เป็นอัลกอริทึมที่ง่ายต่อการทำนายและมีการทำงานที่เร็ว.
- **ใช้งานได้ดีกับข้อมูลที่มีมิติสูง:** Naive Bayes สามารถใช้กับข้อมูลที่มีมิติสูงโดยไม่ต้องใช้ทรัพยากรคำนวณมากมาย.
- **มีประสิทธิภาพสูงในงานที่มีตัวแปรส่วนมาก:** ทำงานได้ดีในข้อมูลที่มีจำนวนตัวอย่างน้อยและมีตัวแปรมาก.
- **ทำนายและจำแนกได้ในทางไบนารี:** Naive Bayes มักถูกนำมาใช้ในการทำนายทางไบนารี, เช่น การจำแนกออกซิลาตรา, การตรวจสอบข้อความสแปม, หรือการจำแนกอยู่ในกลุ่ม A หรือ B.

6.6.4 Naive Bayes

6.6.4 Naive Bayes

■ ส่วนประกอบหลักของ Naive Bayes ประกอบไปด้วย:

1. **ความน่าจะเป็น (Probability):** Naive Bayes ใช้ความน่าจะเป็นเพื่อทำนายการจำแนกข้อมูลเป็นกลุ่มหรือประเภทที่แตกต่างกัน. ความน่าจะเป็นถูกคำนวณโดยใช้ Bayes' Theorem ซึ่งมีส่วนประกอบหลัก ๆ คือ ความน่าจะเป็นของกลุ่มที่ต้องการจำแนก (class) เทียบกับลักษณะของข้อมูล.

2. **Bayes' Theorem:** สูตรทางสถิติที่ใช้ในการคำนวณความน่าจะเป็นของเหตุการณ์ A โดยทราบความน่าจะเป็นของ B และความน่าจะเป็นของ B เทียบกับ A

โดยที่:

- ($P(A|B)$) คือความน่าจะเป็นของ A เมื่อทราบ B.
- ($P(B|A)$) คือความน่าจะเป็นของ B เมื่อทราบ A.
- ($P(A)$) คือความน่าจะเป็นของ A.
- ($P(B)$) คือความน่าจะเป็นของ B.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

6.6.4 Naive Bayes

6.6.4 Naive Bayes

- **ส่วนประกอบหลักของ Naive Bayes ประกอบด้วย:**

3. Naive Assumption: Naive Bayes ทำการทำนายด้วยการทำสมมติฐานที่เรียกว่า "Naive" หรือ "ซบซอนน้อย" ที่ความน่าจะเป็นของแต่ละลักษณะ (features) ที่ใช้ในการจำแนก ถูกสมมติว่าเป็นอิสระต่อกัน.

4. ลักษณะ (Features): ลักษณะของข้อมูลที่ใช้ในการจำแนก ที่ Naive Bayes สมมติว่าเป็นอิสระต่อกัน.

5. หลักการทำงาน: ในการจำแนก, Naive Bayes คำนวณความน่าจะเป็นของแต่ละกลุ่มตามลักษณะของข้อมูล แล้วเลือกกลุ่มที่มีความน่าจะเป็นสูงสุดเป็นคำตอบ

Naive Bayes ได้รับชื่อว่า "Naive" เนื่องจากมีสมมติฐานที่ไม่ได้คิดถึงความสัมพันธ์ระหว่างลักษณะ (features) ซึ่งทำให้การคำนวณความน่าจะเป็นง่ายและเร็ว. ถึงแม้ว่าสมมติฐานนี้จะไม่สอดคล้องกับบางสถานการณ์จริง ๆ แต่ Naive Bayes ยังมีประสิทธิภาพดีในการจำแนกข้อมูลในหลายสถานการณ์

6.6.4 Naive Bayes (ต่อ...)

■ ตัวอย่างการนำ Naive Bayes ไปใช้ เช่น

- การจำแนกประเภทลูกค้า ร้านค้าออนไลน์อาจใช้ Naive Bayes เพื่อจำแนกประเภทลูกค้าว่าจะเป็นลูกค้าที่มีแนวโน้มจะซื้อสินค้าหรือไม่ โดยพิจารณาจากข้อมูลต่างๆ ของลูกค้า เช่น อายุ เพศ ที่อยู่ ประวัติการซื้อ เป็นต้น
- การจำแนกประเภทโรค โรงพยาบาลอาจใช้ Naive Bayes เพื่อจำแนกประเภทโรคว่าเป็นโรคร้ายแรงหรือไม่ โดยพิจารณาจากข้อมูลต่างๆ ของผู้ป่วย เช่น อาการ ประวัติทางการแพทย์ เป็นต้น
- การจำแนกประเภทข้อความ ผู้ให้บริการอีเมลอาจใช้ Naive Bayes เพื่อจำแนกประเภทข้อความว่าเป็นสแปมหรือไม่ โดยพิจารณาจากเนื้อหาของข้อความ เป็นต้น
- การจำแนกประเภทเว็บไซต์ ผู้ให้บริการค้นหาอาจใช้ Naive Bayes เพื่อจำแนกประเภทเว็บไซต์ว่ามีความน่าเชื่อถือหรือไม่ โดยพิจารณาจากเนื้อหาของเว็บไซต์ เป็นต้น

โดยสรุป Naive Bayes เป็นเทคนิคการเรียนรู้ของเครื่องที่มีประสิทธิภาพสูงและสามารถประยุกต์ใช้กับปัญหาการจำแนกประเภทได้หลากหลาย จึงเป็นเทคนิคที่มีความสำคัญในงานเหมืองข้อมูล

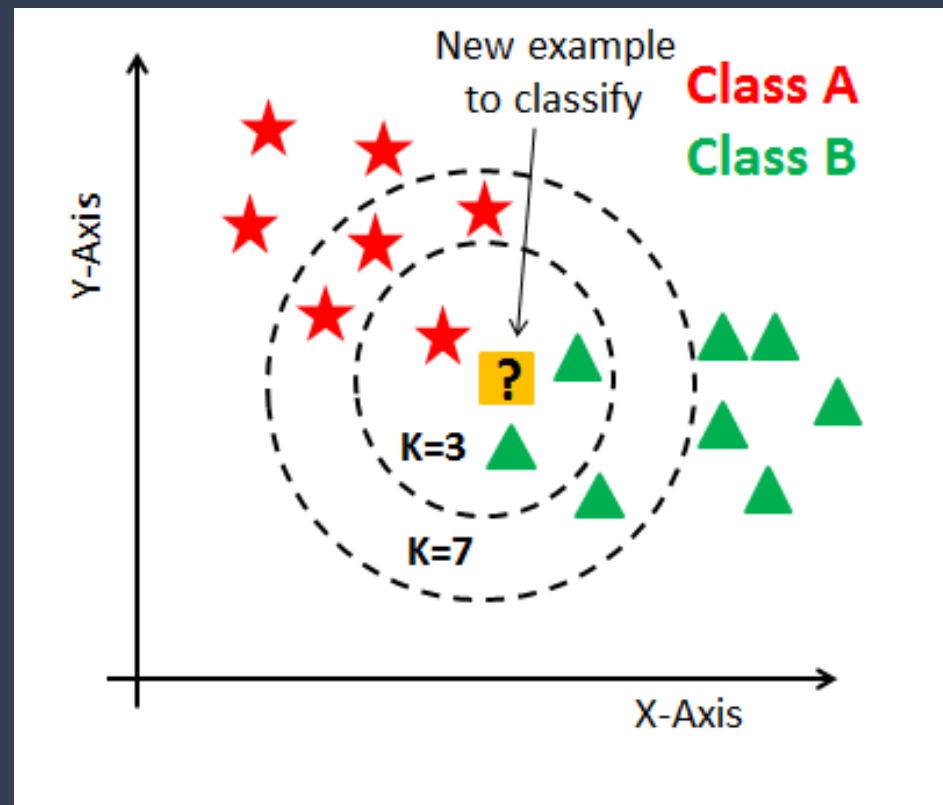
6.6.5 K-nearest neighbors

6.6.5 K-nearest neighbors ในการทำเหมืองข้อมูล

K-nearest neighbors (KNN) เป็นเทคนิคการเรียนรู้ของเครื่องประเภท supervised learning ที่ใช้สำหรับปัญหาการจำแนกประเภท (classification) โดยพิจารณาจากข้อมูลใกล้เคียงที่สุด (nearest neighbors)

KNN ทำงานโดยค้นหาข้อมูลใกล้เคียงที่สุดกับข้อมูลที่ต้องการทำนายจำนวน K ตัว จากนั้นจึงให้ข้อมูลเหล่านั้นโหวตว่าข้อมูลที่ต้องการทำนายควรอยู่ในคลาสใดคลาสหนึ่ง

ค่า K เป็นตัวกำหนดว่าต้องการค้นหาข้อมูลใกล้เคียงที่สุดจำนวนเท่าใด โดยค่า K ที่สูงจะให้ความสำคัญกับข้อมูลที่อยู่ไกลออกไปมากขึ้น ในขณะที่ค่า K ที่ต่ำจะให้ความสำคัญกับข้อมูลที่อยู่ใกล้เข้ามามากขึ้น



6.6.5 K-nearest neighbors

6.6.5 K-nearest neighbors ในการทำเหมืองข้อมูล

K-nearest neighbors (KNN) เป็นเทคนิคการเรียนรู้ของเครื่องประเภท supervised learning ที่ใช้สำหรับปัญหาการจำแนกประเภท (classification) โดยพิจารณาจากข้อมูลใกล้เคียงที่สุด (nearest neighbors)

KNN ทำงานโดยค้นหาข้อมูลใกล้เคียงที่สุดกับข้อมูลที่ต้องการทำนายจำนวน K ตัว จากนั้นจึงให้ข้อมูลเหล่านั้นโหวตว่าข้อมูลที่ต้องการทำนายควรอยู่ในคลาสใดคลาสหนึ่ง

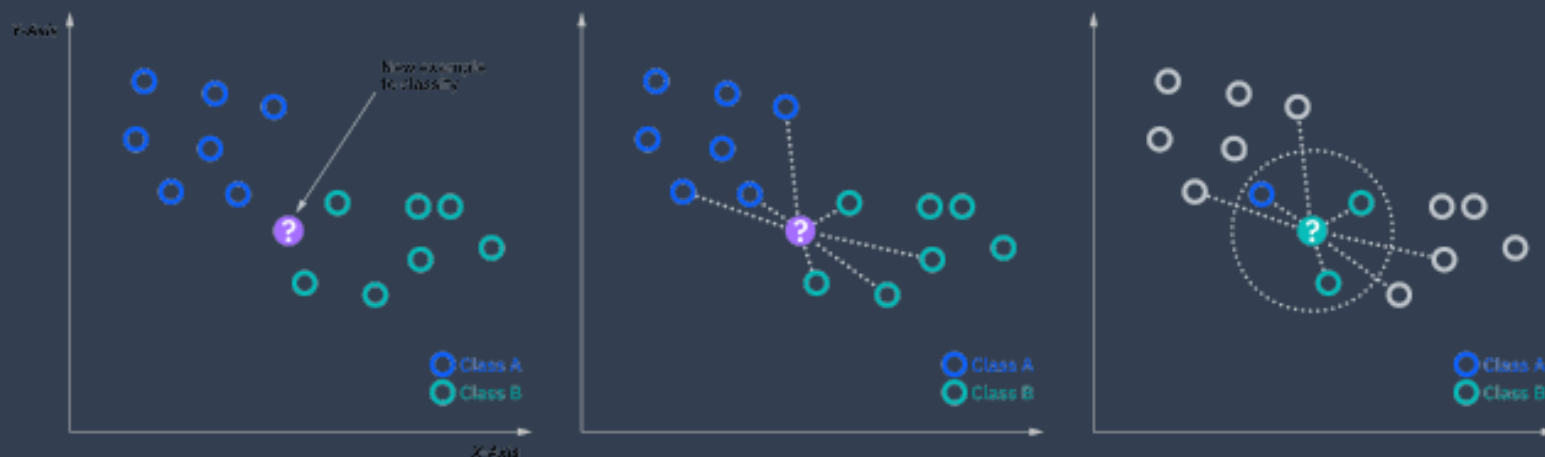
ค่า K เป็นตัวกำหนดว่าต้องการค้นหาข้อมูลใกล้เคียงที่สุดจำนวนเท่าใด โดยค่า K ที่สูงจะให้ความสำคัญกับข้อมูลที่อยู่ไกลออกไปมากขึ้น ในขณะที่ค่า K ที่ต่ำจะให้ความสำคัญกับข้อมูลที่อยู่ใกล้เข้ามามากขึ้น

■ ลักษณะสำคัญของ K-NN ในการทำเหมืองข้อมูล:

- ง่ายต่อการทำนาย: K-NN เป็นอัลกอริทึมที่ง่ายต่อการทำนายและไม่ต้องการการฝึกล่วงหน้า.
- ใช้งานได้ดีกับข้อมูลมีมิติสูง: K-NN สามารถใช้กับข้อมูลที่มีมิติสูงโดยไม่ต้องใช้ทรัพยากรคำนวณมากมาย.
- ไม่ต้องการการสร้างโมเดล: K-NN ไม่ต้องการขั้นตอนการฝึกและสร้างโมเดล มันใช้ข้อมูลโดยตรงในการทำนาย.
- การให้คะแนนเส้นทาง: ในบางกรณี, K-NN สามารถให้คะแนนเส้นทางแบบชัดเจน.

6.6.5 K-nearest neighbors

- <https://www.ibm.com/topics/knn>



โดยจากรูปจะเห็นว่า มีตัวข้อมูลสีม่วงที่เราไม่รู้ว่า เป็นข้อมูลกลุ่มใด เราจึงลองดูว่าในระยะใกล้ ๆ มีข้อมูลใดอยู่บ้างเพื่อดูว่าตัวที่เราสนใจใกล้เคียงกับข้อมูลชุดไหนมากที่สุด ซึ่งจากรูปจะเห็นว่าจากระยะในจุดสีม่วงจะมีสีเขียวมากกว่า ดังนั้นระบบจึงให้คำตอบเป็นสีเขียว เมื่อเราได้รู้จักมันคร่าว ๆ กันแล้ววันนี้ DIGI จะพาทุกคนไปรู้เจ้าตัว K-Nearest Neighbors ให้มากขึ้นกันครับ

6.6.5 K-nearest neighbors

หลักการทำงานของ K Nearest Neighbor Algorithm

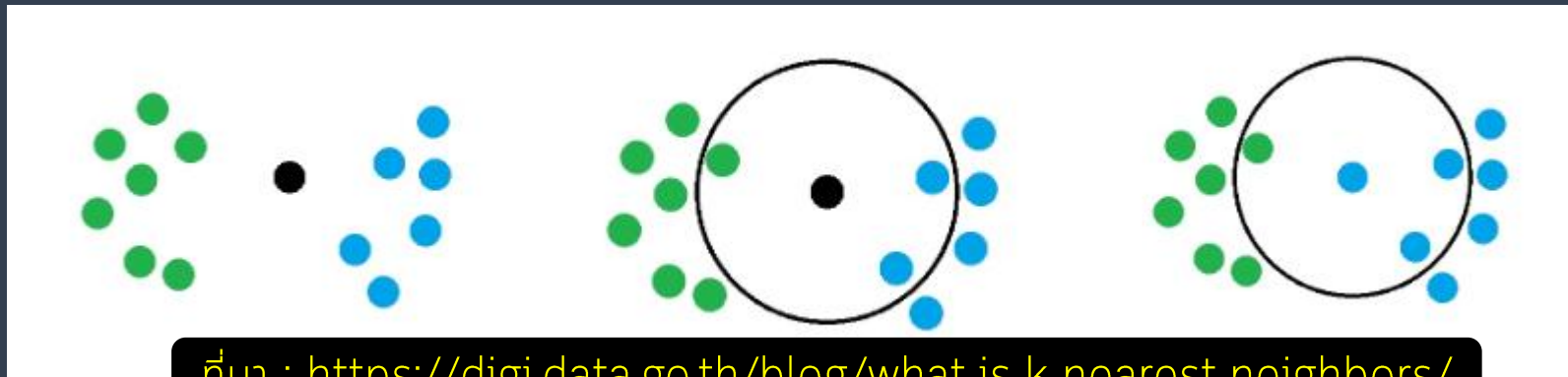
หลักการทำงานของอัลกอริทึม K-Nearest Neighbors (KNN) คือการจัดกลุ่มหรือจำแนกประเภทข้อมูลใหม่โดยอิงถึงข้อมูลที่ใกล้เคียงที่สุดกับข้อมูลทดสอบ เพื่อคาดเดาหรือจำแนกประเภทข้อมูลใหม่ อัลกอริทึมนี้สนับสนุนการเรียนรู้เชิงกระบวนการ (instance-based learning) ซึ่งไม่มีขั้นตอนการเรียนรู้ที่ซับซ้อน แต่จะใช้ข้อมูลที่มีอยู่เพื่อตัดสินใจในการจัดกลุ่มหรือจำแนกประเภทข้อมูลใหม่ โดยหลักการทำงานของ KNN สามารถสรุปได้ดังนี้:

- **เลือกค่า K:** เราต้องกำหนดค่า K ที่ต้องการ ซึ่งเป็นจำนวนของข้อมูลที่ใกล้เคียงที่สุดที่จะใช้ในการตัดสินใจ ค่า K นี้มีผลต่อความถูกต้องและความสามารถในการทำนายของโมเดล
- **คำนวณระยะทาง:** ในกระบวนการ KNN ส่วนใหญ่ใช้ระยะทางยูคลิดีียน (Euclidean distance) เพื่อคำนวณระยะทางระหว่างข้อมูลทดสอบกับข้อมูลในชุดข้อมูลการฝึกฝน ระยะทางนี้จะวัดความคล้ายคลึงของคุณสมบัติระหว่างข้อมูล
- **หาข้อมูลที่ใกล้เคียงที่สุด:** หลังจากคำนวณระยะทางระหว่างข้อมูลทดสอบกับข้อมูลในชุดข้อมูลการฝึกฝน เราจะเลือกข้อมูล K รายการที่มีระยะทางน้อยที่สุด
- **คำนวณผลโหวต:** เมื่อเราได้ข้อมูล K รายการที่ใกล้เคียงที่สุดแล้ว เราจะนับจำนวนรายการในแต่ละกลุ่มหรือประเภทข้อมูล และกำหนดกลุ่มหรือประเภทข้อมูลของข้อมูลทดสอบตามจำนวนที่มากที่สุดภายใน K รายการนั้น
- **ทำนายผลลัพธ์:** สุดท้ายนี้ เราจะใช้กลุ่มหรือประเภทข้อมูลที่ได้มาจากขั้นตอนก่อนหน้านี้ในการทำนายผลลัพธ์ของข้อมูลทดสอบ

สมมุติมีกลุ่ม 2 กลุ่ม และให้ค่า $K = 3$ (K คือค่าที่ไว้ดูจำนวนข้อมูลที่ใกล้เคียงที่สุดว่าจะดูกี่จำนวนในการตัดสินใจเลือกกลุ่ม ซึ่งส่วนใหญ่นิยมใช้เป็นเลขคี่) จะดูว่าใกล้เคียงกับข้อมูลชุดไหนมากที่สุดก็จะได้คำตอบตามข้อมูลนั้น ๆ ตามรูป

6.6.5 K-nearest neighbors

■ หลักการทำงานของ K Nearest Neighbor Algorithm



ที่มา : <https://digi.data.go.th/blog/what-is-k-nearest-neighbors/>

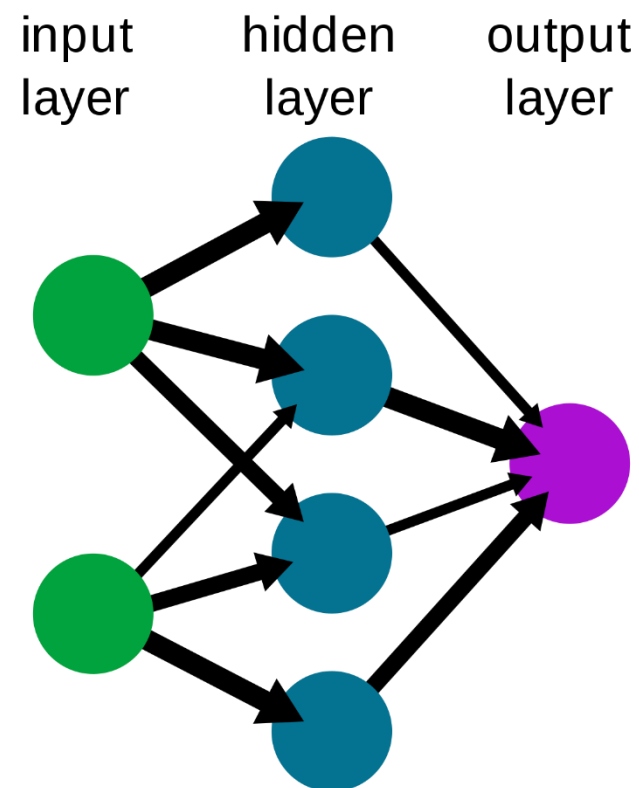
6.6.6 Neural Networks (NN)

6.6.6 Neural Networks (NN)

Neural Networks (NN) หรือโครงข่ายประสาทเทียม, เป็นเครื่องมือที่มีประสิทธิภาพสูงในการทำเหมืองข้อมูล (data mining) โดยเฉพาะการจำแนกข้อมูลหรือการพยากรณ์. นี่คือนางวิธีที่ Neural Networks ถูกนำมาใช้ในการทำเหมืองข้อมูล:

Neural Networks (NN) หรือโครงข่ายประสาทเทียมเป็นโมเดลทางคณิตศาสตร์ที่จำลองการทำงานของระบบประสาทของมนุษย์ในการประมวลผลข้อมูลและการเรียนรู้ โครงสร้างของ Neural Networks ประกอบไปด้วยหลายส่วนที่ทำงานร่วมกันเพื่อดำเนินการต่างๆ ต่อไปนี้คือส่วนประกอบหลักของ Neural Networks:

A simple neural network



https://en.wikipedia.org/wiki/Neural_network

6.6.6 Neural Networks (NN)

■ Neural Networks (NN)

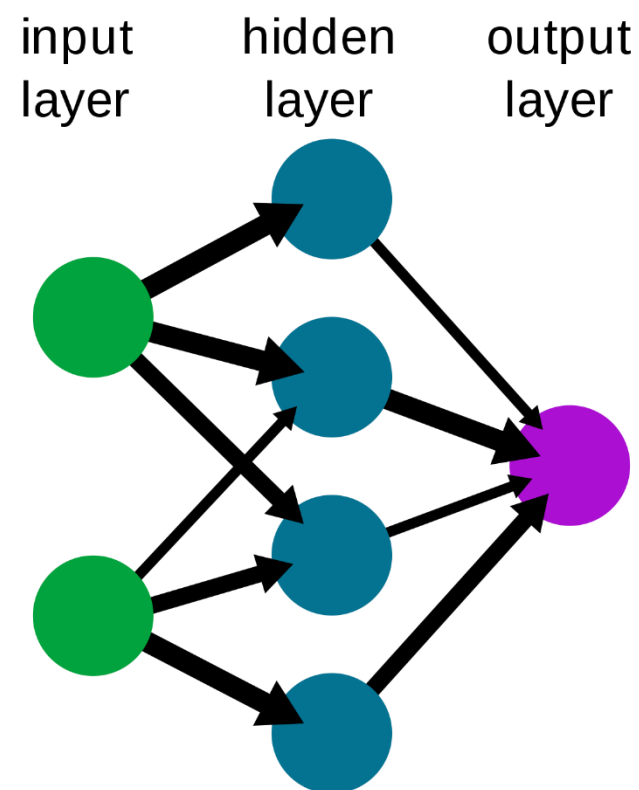
1. Input Layer (ชั้นนำเข้า): ชั้นนำเข้ารับข้อมูลจากภายนอก เช่น ภาพ, เสียง, ข้อความ, หรือข้อมูลทางตัวเลขอื่น ๆ ซึ่งจะถูกนำเข้าไปในโมเดลสำหรับการประมวลผล.

2. Hidden Layers (ชั้นซ่อน): ชั้นที่ไม่ถูกเปิดเผยต่อภายนอก แต่ทำหน้าที่ประมวลผลข้อมูลจากชั้นนำเข้า และส่งผลลัพธ์ไปยังชั้นถัดไป. Neural Networks สามารถมีหลายชั้นซ่อน (hidden layers) ขึ้นอยู่กับความซับซ้อนของโครงสร้าง

3. Weights (น้ำหนัก): น้ำหนักถูกกำหนดให้กับแต่ละเชื่อมโยง (connection) ระหว่างโหนดในชั้นที่ i กับโหนดในชั้นที่ $i+1$. น้ำหนักนี้ถูกปรับเปลี่ยนในขั้นตอนการฝึก (training) เพื่อให้ระบบสามารถทำนายผลลัพธ์ที่ถูกต้อง.

4. Activation Function (ฟังก์ชันกระตุ้น): ฟังก์ชันที่ใช้ในแต่ละโหนด (node) เพื่อทำให้มีการเรียนรู้และประมวลผลได้. ฟังก์ชันนี้มีหน้าที่ให้ผลลัพธ์ที่ไม่เป็นเชิงเส้น ทำให้ Neural Networks สามารถจับลักษณะที่ซับซ้อนของข้อมูลได้.

A simple neural network



https://en.wikipedia.org/wiki/Neural_network

6.6.6 Neural Networks (NN)

■ Neural Networks (NN)

5. Bias (ไบแอส): ไบแอสเป็นค่าคงที่ที่เพิ่มเข้าไปในผลรวมของน้ำหนักและข้อมูลที่มีไบเอนด์ มีหน้าที่ให้ความยืดหยุ่นในการปรับแต่งการเรียนรู้ของระบบ.

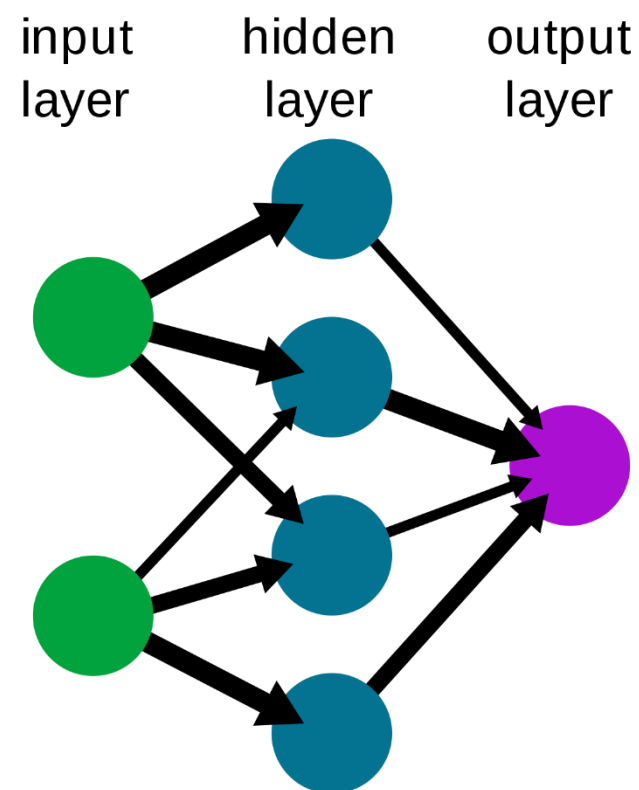
6. Output Layer (ชั้นส่งออก): ชั้นที่ได้ผลลัพธ์สุดท้ายหลังจากที่ข้อมูลได้รับการประมวลผลผ่านทางชั้นซ่อน และมีความเกี่ยวข้องกับปัญหาที่กำหนด, เช่น การจำแนก, การทำนาย, หรือการแก้ปัญหาทางคณิตศาสตร์.

7. Loss Function (ฟังก์ชันสูญเสีย): ฟังก์ชันที่ใช้วัดความคลาดเคลื่อนระหว่างผลลัพธ์ที่ได้จากระบบและผลลัพธ์ที่ควรจะได้ ฟังก์ชันนี้ถูกใช้ในกระบวนการฝึก Neural Networks.

8. Optimizer: อัลกอริทึมที่ใช้ในการปรับค่าน้ำหนักเพื่อลดค่าคลาดเคลื่อนและเพิ่มประสิทธิภาพของระบบ โดยมักจะใช้ตัวอัลกอริทึมเช่น Gradient Descent, Adam, หรือ RMSprop.

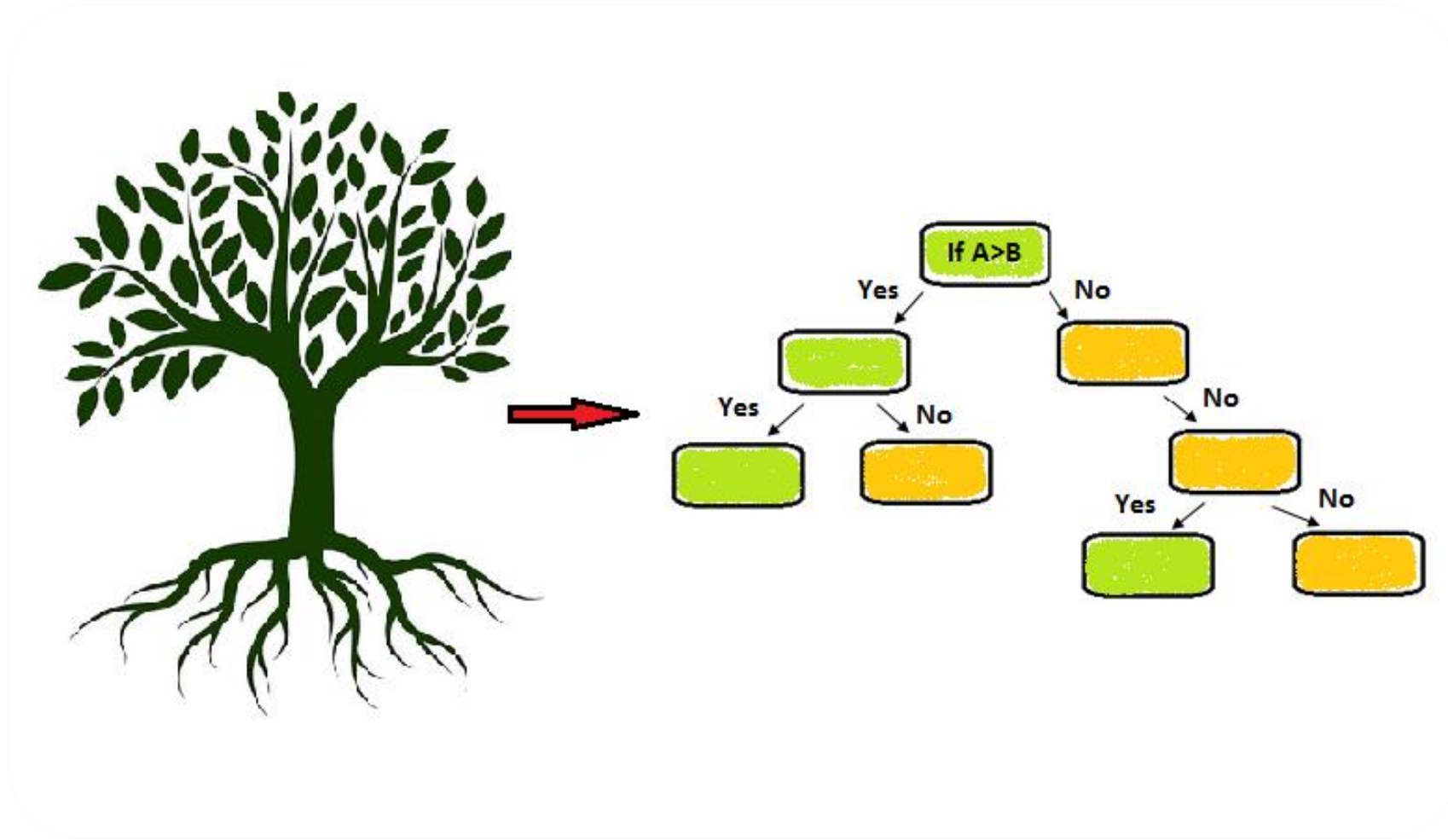
Neural Networks มีความหลากหลายและมีการปรับปรุงเพิ่มเติมขึ้นตามความต้องการของแต่ละงาน แต่ส่วนประกอบที่กล่าวถึงเป็นพื้นฐานที่สำคัญในการทำงานของ Neural Networks.

A simple neural network



https://en.wikipedia.org/wiki/Neural_network

6.7 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree)



6.7.1 การจัดเตรียมข้อมูลสำหรับการจำแนกและการทำนาย ข้อมูล

1) การทำความสะอาดข้อมูล (Data cleansing)—จะเกี่ยวข้องกับการประมวลผลข้อมูลเบื้องต้นที่จะลบหรือลดข้อมูลที่มีสิ่งรบกวน (noise) ด้วยการประยุกต์ใช้วิธีการปรับเรียบข้อมูลแบบต่าง ๆ และจัดการกับการขาดหายไปของข้อมูลด้วยการแทนค่าของข้อมูลที่ขาดหายไปด้วยค่าของข้อมูลที่ปรากฏบ่อยที่สุดหรือทำการแทนด้วยค่าของข้อมูลที่มีค่าเชิงสถิติสูงสุด เป็นต้น

2) ความเกี่ยวเนื่องของข้อมูล (Relevance analysis)—จะทำการตรวจสอบข้อมูลแอตทริบิวต์ต่าง ๆ ว่ามีความเกี่ยวเนื่องหรือซ้ำซ้อนกันมากน้อยเพียงใด ซึ่งโดยปกติของชุดข้อมูลจะมีแอตทริบิวต์ที่ซ้ำซ้อนกัน ดังนั้น เพื่อที่จะหลีกเลี่ยงความซ้ำซ้อนดังกล่าว เราสามารถประยุกต์ใช้การวิเคราะห์สหสัมพันธ์ (correlation analysis) เพื่อทำการตรวจสอบทางสถิติว่า 2 แอตทริบิวต์ใดๆ (เลือกพิจารณาทีละคู่ของแอตทริบิวต์จากแอตทริบิวต์ทั้งหมด) มีความเหมือนกันหรือแตกต่างกันมากน้อยเพียงใด ตัวอย่างเช่น แอตทริบิวต์ $A1$ และ $A2$ ที่มีความเหมือนกันค่อนข้างมากจะทำให้เราทราบว่าเราควรที่จะต้องลบแอตทริบิวต์ใดแอตทริบิวต์หนึ่งออกจากเซตของแอตทริบิวต์ที่จะทำการพิจารณา จากนั้นเราจะต้องทำการเลือกแอตทริบิวต์ที่น่าสนใจ (Attribute subset selection) เพื่อลดปริมาณแอตทริบิวต์ที่ต้องพิจารณา อันนำมาซึ่งการลดทรัพยากรในการคำนวณและอาจเป็นการเพิ่มความถูกต้องของการค้นหาลักษณะได้อีกด้วย

6.7.1 การจัดเตรียมข้อมูลสำหรับการจำแนกและการทำนายข้อมูล(ต่อ)

3) การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลและการลดจำนวนข้อมูล (Data transformation and reduction)—ข้อมูลที่เป็นอินพุตอาจมีช่วงของข้อมูลหรือค่าของข้อมูลที่มีระยะห่างค่อนข้างมาก ดังนั้น เราอาจทำการเปลี่ยนแปลง/เปลี่ยนรูปด้วยวิธีการ normalization ที่จะทำการปรับเปลี่ยนค่าในแอตทริบิวต์หนึ่ง ๆ ให้อยู่ในช่วงที่กำหนด อาทิเช่น ช่วง -1.0 ถึง 1.0 หรือ ช่วง 0.0 ถึง 1.0 เป็นต้น การปรับเปลี่ยนช่วงของข้อมูลจะช่วยให้เรื่องของการหาความแตกต่างระหว่างข้อมูลเรคคอร์ดต่างๆ (distance measure) นอกจากนั้นการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลยังสามารถทำได้ด้วยวิธีการ generalization ที่ซึ่งจะทำให้ข้อมูลมีความละเอียดน้อยลง ตัวอย่างเช่น แอททริบิวต์ที่บ่งบอกถึงรายได้ที่เป็นค่าต่อเนื่อง (ค่าเงินจริงที่แต่ละบุคคลได้รับในแต่ละเดือน) จะสามารถถูกทำ generalize ให้อยู่ในช่วงที่ไม่ต่อเนื่องได้ อาทิเช่น low, medium และ high เป็นต้น ยิ่งไปกว่านั้น เรายังสามารถลดจำนวนข้อมูลที่ต้องการการพิจารณาได้ด้วยการประยุกต์ใช้เทคนิคต่าง ๆ อาทิเช่น

wavelet transformation และ principal component analysis รวมถึงเทคนิคการทำ binning, histogram analysis และ clustering เป็นต้น

จะสามารถถูกทำ generalize ให้อยู่ในช่วงที่ไม่ต่อเนื่องได้ อาทิเช่น low, medium และ high เป็นต้น

ลำดับ	อายุ	อาชีพ	รายได้	รายจ่าย	เงินเดือน	ประวัติ	คลาส
1	36	อาจารย์	50000	30000	40000	ล่าช้า	เสี่ยง
2	40	หมอ	70000	40000	50000	ไม่มี	ปลอดภัย
3	50	ภารโรง	30000	10000	20000	ไม่มี	ปลอดภัย
4	24	พนักงานออฟฟิศ	40000	20000	30000	ค้างชำระ	เสี่ยง

ลำดับ	อายุ	อาชีพ	รายได้	รายจ่าย	เงินเดือน	ประวัติ	คลาส
1	36	อาจารย์	MEDIUM	30000	40000	ล่าช้า	เสี่ยง
2	40	หมอ	HIGH	40000	50000	ไม่มี	ปลอดภัย
3	50	ภารโรง	MEDIUM	10000	20000	ไม่มี	ปลอดภัย
4	24	พนักงานออฟฟิศ	MEDIUM	20000	30000	ค้างชำระ	เสี่ยง

1 - 10000 Low
10001 - 50000 MEDIUM
50001 - 100000 High
100001 ขึ้น Very High

6.7.2 การเปรียบเทียบประสิทธิภาพวิธีในการจำแนกและทำนายข้อมูล

ในการเปรียบเทียบประสิทธิภาพของวิธีในการจำแนกและทำนายข้อมูล จะทำการประยุกต์ใช้เกณฑ์ดังต่อไปนี้

1) ความถูกต้อง (Accuracy)—จะเกี่ยวข้องกับความสามารถของตัวจำแนกข้อมูลที่ถูกสร้างขึ้นที่จะสามารถจำแนกข้อมูลที่ไม่เคยพบเจอมาก่อนได้อย่างถูกต้อง โดยในการวัดความถูกต้องอาจประเมินได้จากการใช้ชุดข้อมูลหนึ่ง ๆ (หรือมากกว่าหนึ่งชุดก็ได้) ที่แยกจากชุดข้อมูลเรียนรู้ (training dataset)

2) ความเร็ว (Speed)—จะเกี่ยวข้องกับเวลาที่ใช้ในการคำนวณทั้งในส่วนของการสร้างตัวจำแนกข้อมูลและการจำแนก/ทำนายข้อมูล (เวลาในการเรียนรู้ + เวลาทดสอบ)

6.7.2 การเปรียบเทียบประสิทธิภาพวิธีในการจำแนกและทำนายข้อมูล (ต่อ)

3) ความทนทาน (Robustness)—จะเกี่ยวข้องกับความสามารถของตัวจำแนกหรือตัวทำนายข้อมูลที่จะทำการทำนายได้อย่างถูกต้องจากข้อมูลตั้งต้นที่มีสิ่งรบกวนหรือมีการขาดหายไปของข้อมูล

4) ความยืดหยุ่นต่อปริมาณข้อมูล (Scalability)—จะเกี่ยวข้องกับความสามารถในการสร้างตัวจำแนกข้อมูลหรือตัวทำนายข้อมูลได้อย่างมีประสิทธิภาพเมื่อมีข้อมูลที่ต้องพิจารณาเป็นปริมาณมาก

5) ความสามารถในการเข้าใจ (Interpretability)—เกี่ยวเนื่องกับระดับความสามารถที่จะถูกเข้าใจในตัวจำแนกหรือทำนายข้อมูลจากผู้ใช้งาน

ความยืดหยุ่นต่อปริมาณข้อมูล (Scalability)—จะเกี่ยวข้องกับความสามารถในการสร้างตัวจำแนกข้อมูลหรือตัวทำนายข้อมูลได้อย่างมีประสิทธิภาพเมื่อมีข้อมูลที่ต้องพิจารณาเป็นปริมาณมาก

ลำดับ	อายุ	อาชีพ	รายได้	รายจ่าย	เงินเดือน	ประวัติ	คลาส
1	36	อาจารย์	MEDIUM	30000	40000	ล่าช้า	เสี่ยง
2	40	หมอ	HIGH	40000	50000	ไม่มี	ปลอดภัย
3	50	ภารโรง	MEDIUM	10000	20000	ไม่มี	ปลอดภัย
4	24	พนักงานออฟฟิส	MEDIUM	20000	30000	ค้างชำระ	เสี่ยง
....							

หมายเหตุ:

คอลัมน์ "รายได้" แบ่งเป็น 3 ระดับ: ต่ำ (LOW), ปานกลาง (MEDIUM), สูง (HIGH)
คอลัมน์ "ประวัติ" แบ่งเป็น 2 ประเภท: ดี (ไม่มีประวัติเสีย), เสีย (มีประวัติล่าช้า ดำรงชำระ)
คอลัมน์ "คลาส" บ่งบอกว่าผู้ขอกู้มีสถานะการเงินที่ "ปลอดภัย" หรือ "เสี่ยง"

การวิเคราะห์

จากตารางข้อมูลข้างต้น เราสามารถวิเคราะห์ได้ดังนี้:

- ผู้ขอกู้ที่มีอายุน้อย มักจะมีรายได้น้อยกว่าผู้ขอกู้ที่มีอายุมาก (30,000 บาท vs. 40,000 บาท)
- ผู้ขอกู้ที่มีอาชีพเป็นหมอ มักจะมีรายได้มากกว่าผู้ขอกู้ที่มีอาชีพอื่น (50,000 บาท vs. 40,000 บาท)
- ผู้ขอกู้ที่มีรายได้มากกว่ารายจ่าย มักจะมีสถานะการเงินที่ปลอดภัย (2 จาก 3)
- ผู้ขอกู้ที่มีประวัติการชำระหนี้ล่าช้าหรือค้างชำระ มักจะมีสถานะการเงินที่เสี่ยง (2 จาก 2)

ข้อจำกัด

ข้อมูลในตารางนี้มีจำนวนจำกัด ไม่สามารถสรุปผลได้อย่างแน่ชัด จำเป็นต้องมีข้อมูลเพิ่มเติมเพื่อวิเคราะห์อนาคต

3) ความทนทาน (Robustness)—จะเกี่ยวข้องกับความสามารถของตัวจำแนกหรือตัวทำนายข้อมูลที่จะทำการทำนายได้อย่างถูกต้องจากข้อมูลตั้งต้นที่มีสิ่งรบกวนหรือมีการขาดหายไปของข้อมูล

ลำดับ	อายุ	อาชีพ	รายได้	รายจ่าย	เงินเดือน	ประวัติ	คลาส
1	36	อาจารย์	MEDIUM	30000	40000	ล่าช้า	เสี่ยง
2	40	หมอ	HIGH	40000	50000	ไม่มี	ปลอดภัย
3	50	ภารโรง	MEDIUM	10000	20000	ไม่มี	ปลอดภัย
4	24	พนักงานออฟฟิศ	MEDIUM	20000	30000	ค้างชำระ	เสี่ยง
5	??	พนักงานโรงงาน	MEDIUM	20000	30000	ค้างชำระ	เสี่ยง
100000							???

3) ความทนทาน (Robustness)--จะเกี่ยวข้องกับความสามารถของตัวจำแนกหรือตัวทำนายข้อมูลที่จะทำการทำนายได้อย่างถูกต้องจากข้อมูลตั้งต้นที่มีสิ่งรบกวนหรือมีการขาดหายไปของข้อมูล

ลำดับ	อายุ	อาชีพ	รายได้	รายจ่าย	เงินเดือน	ประวัติ	คลาส
1	36	อาจารย์	MEDIUM	30000	40000	ล่าช้า	เสี่ยง
2	40	หมอ	HIGH	40000	50000	ไม่มี	ปลอดภัย
3	50	ภารโรง	MEDIUM	10000	20000	ไม่มี	ปลอดภัย
4	24	พนักงานออฟฟิศ	MEDIUM	20000	30000	ค้างชำระ	เสี่ยง
5	??	พนักงานโรงงาน	MEDIUM	20000	30000	ค้างชำระ	เสี่ยง
100000							???

6.7.3 การเตรียมข้อมูลประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

ตัวอย่างต่อไปนี้เป็นสร้างข้อมูลทดสอบสำหรับการใช้ Decision Tree ในการทำนาย Churn (การตกลง) ของลูกค้า โดยมีคุณลักษณะหลายประการที่สามารถมีผลต่อการตัดสินใจของลูกค้าในการเลือกที่จะตกลงหรือไม่ตกลงในการใช้บริการ หรือสินค้า โดย Churn มักถูกนิยามว่าลูกค้าที่ยุติการใช้บริการหรือสินค้า

ตัวอย่างข้อมูลที่ถูกสร้างมีคุณลักษณะดังนี้:

- 'อายุ': อายุของลูกค้าที่ถูกสุ่มมาจากการสุ่มตัวเลขในช่วง 18-64 ปี.
- 'เพศ': เพศของลูกค้าที่ถูกสุ่มมาเป็นชายหรือหญิง.
- 'รายได้': รายได้ของลูกค้าที่ถูกสุ่มมาเป็น 'น้อย' หรือ 'มาก'.
- 'การศึกษา': ระดับการศึกษาของลูกค้าที่ถูกสุ่มมาเป็น 'ปริญญาตรี', 'มัธยม', หรือ 'ปริญญาโท'.
- 'การทำงาน': สถานะการทำงานของลูกค้าที่ถูกสุ่มมาเป็น 'ทำงาน' หรือ 'ไม่ทำงาน'.
- 'Churn': ค่าเป้าหมายที่ถูกสุ่มมาเพื่อแสดงว่าลูกค้าตกลง ('ใช่') หรือไม่ตกลง ('ไม่').

File :

Lec06_DT_Gen_dataset.py

6.7.3 การเตรียมข้อมูลประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

```
import pandas as pd
import numpy as np
np.random.seed(42)
# สร้าง DataFrame สำหรับตัวอย่าง 1000 รายการ
data = {
    'อายุ': np.random.randint(18, 65, size=1000),
    'เพศ': np.random.choice(['ชาย', 'หญิง'], size=1000),
    'รายได้': np.random.choice(['น้อย', 'มาก'], size=1000),
    'การศึกษา': np.random.choice(['ปริญญาตรี', 'มัธยม', 'ปริญญาโท'], size=1000),
    'การทำงาน': np.random.choice(['ทำงาน', 'ไม่ทำงาน'], size=1000),
    'Churn': np.random.choice(['ใช่', 'ไม่'], p=[0.2, 0.8], size=1000)
}
df = pd.DataFrame(data)
# บันทึก DataFrame เป็นไฟล์ Excel
df.to_excel('decision_tree_dataset01.xlsx', index=False)
```

File : Lec06_DT_Gen_dataset.py

หลังจากสร้างข้อมูล, คำสั่ง `df.to_excel('decision_tree_dataset.xlsx', index=False)` ถูกใช้เพื่อบันทึก DataFrame ลงในไฟล์ Excel ที่ชื่อ `decision_tree_dataset.xlsx` โดยไม่รวม index ลงไป นำไปใช้เป็นข้อมูลในการฝึกและทดสอบ Decision Tree สำหรับการทำนาย Churn ของลูกค้า

6.7.3 การเตรียมข้อมูลประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

	A	B	C	D	E	F
1	อายุ	เพศ	รายได้	การศึกษา	การทำงาน	Churn
2	56	ชาย	มาก	มัธยม	ไม่ทำงาน	ใช่
3	46	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
4	32	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
5	60	หญิง	มาก	มัธยม	ไม่ทำงาน	ไม่
6	25	ชาย	มาก	มัธยม	ไม่ทำงาน	ไม่
7	38	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
8	56	ชาย	น้อย	ปริญญาโท	ทำงาน	ไม่
9	36	ชาย	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
10	40	ชาย	น้อย	ปริญญาโท	ไม่ทำงาน	ไม่
11	28	ชาย	น้อย	ปริญญาตรี	ไม่ทำงาน	ไม่
12	28	ชาย	น้อย	ปริญญาโท	ทำงาน	ใช่
13	41	ชาย	น้อย	มัธยม	ไม่ทำงาน	ไม่
14	53	หญิง	น้อย	มัธยม	ทำงาน	ไม่
15	57	หญิง	น้อย	ปริญญาตรี	ไม่ทำงาน	ไม่
16	41	ชาย	น้อย	มัธยม	ไม่ทำงาน	ไม่
17	20	ชาย	น้อย	มัธยม	ไม่ทำงาน	ใช่
18	39	หญิง	น้อย	ปริญญาโท	ทำงาน	ใช่
19	19	ชาย	น้อย	มัธยม	ทำงาน	ไม่
20	41	ชาย	มาก	ปริญญาตรี	ไม่ทำงาน	ใช่
21	61	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
22	47	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
23	55	หญิง	น้อย	ปริญญาตรี	ทำงาน	ใช่
24	19	หญิง	มาก	ปริญญาโท	ทำงาน	ไม่
25	38	หญิง	มาก	ปริญญาโท	ทำงาน	ไม่
26	50	หญิง	มาก	ปริญญาตรี	ไม่ทำงาน	ไม่
27	29	ชาย	น้อย	ปริญญาโท	ทำงาน	ไม่
28	39	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
29	61	ชาย	น้อย	ปริญญาตรี	ไม่ทำงาน	ใช่
30	42	ชาย	น้อย	มัธยม	ไม่ทำงาน	ไม่
31	44	หญิง	มาก	มัธยม	ไม่ทำงาน	ไม่
32	59	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
33	45	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
34	33	ชาย	มาก	ปริญญาตรี	ทำงาน	ไม่
35	32	ชาย	น้อย	ปริญญาโท	ทำงาน	ไม่
36	64	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
37	61	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
38	20	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
39	54	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่

File : Lec06_DT_Gen_dataset.py

หลังจากสร้างข้อมูล, คำสั่ง `df.to_excel('decision_tree_dataset.xlsx', index=False)` ถูกใช้เพื่อบันทึก DataFrame ลงในไฟล์ Excel ที่ชื่อ `decision_tree_dataset.xlsx` โดยไม่รวม index ลงไป. นำไปใช้เป็นข้อมูลในการฝึกและทดสอบ Decision Tree สำหรับการทำนาย Churn ของลูกค้า.

6.7.3 การเตรียมข้อมูลประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

- การประยุกต์ใช้งานต้นไม้ตัดสินใจมักถูกนำมาใช้ในงานที่ต้องการการตัดสินใจหรือการจำแนกหมวดหมู่ ตัวอย่างนี้จะใช้ **scikit-learn** ในการสร้างต้นไม้ตัดสินใจและประยุกต์ใช้งาน
- สมมติว่าเรามีข้อมูลตัวอย่างเกี่ยวกับการตัดสินใจของลูกค้าที่จะซื้อหรือไม่ซื้อสินค้า โดยมีข้อมูลตัวอย่างนี้จะมีคอลัมน์ "อายุ", "เพศ", "รายได้", และ "การซื้อสินค้า" (Churn) ซึ่งบอกว่าลูกค้าซื้อหรือไม่ซื้อ

ตัวอย่างข้อมูล

	อายุ	เพศ	รายได้	การซื้อสินค้า
0	28	ชาย	น้อย	ไม่ซื้อ
1	45	หญิง	มาก	ซื้อ
2	33	ชาย	มาก	ซื้อ
3	22	หญิง	น้อย	ไม่ซื้อ
4	36	ชาย	มาก	ซื้อ
5	50	หญิง	มาก	ไม่ซื้อ
6	28	ชาย	น้อย	ซื้อ
7	40	หญิง	มาก	ซื้อ
8	32	ชาย	น้อย	ไม่ซื้อ
9	55	หญิง	มาก	ซื้อ

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

```
import pandas as pd
```

Part01 :

Lec06_Decision_Tree_Churn01.py

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.metrics import accuracy_score, classification_report,  
confusion_matrix
```

```
from sklearn import preprocessing
```

```
# สร้าง DataFrame สำหรับตัวอย่าง
```

หมายเหตุ : ไฟล์นี้จะมีทั้งหมด 4 ส่วนคือ part01 - part04

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision

	A	B	C	D	E	F
1	อายุ	เพศ	รายได้	การศึกษา	การทำงาน	Churn
2	56	ชาย	มาก	มัธยม	ไม่ทำงาน	ใช่
3	46	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
4	32	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
5	60	หญิง	มาก	มัธยม	ไม่ทำงาน	ไม่
6	25	ชาย	มาก	มัธยม	ไม่ทำงาน	ไม่
7	38	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
8	56	ชาย	น้อย	ปริญญาโท	ทำงาน	ไม่
9	36	ชาย	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
10	40	ชาย	น้อย	ปริญญาโท	ไม่ทำงาน	ไม่

ชื่อไฟล์ :

Lec06_Decision_Tree_Churn01.py

	A	B	C	D	E	F
1	อายุ	เพศ	รายได้	การศึกษา	การทำงาน	Churn
2	56	ชาย	มาก	มัธยม	ไม่ทำงาน	ใช่
3	46	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
4	32	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
5	60	หญิง	มาก	มัธยม	ไม่ทำงาน	ไม่
6	25	ชาย	มาก	มัธยม	ไม่ทำงาน	ไม่
7	38	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
8	56	ชาย	น้อย	ปริญญาโท	ทำงาน	ไม่
9	36	ชาย	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
10	40	ชาย	น้อย	ปริญญาโท	ไม่ทำงาน	ไม่
11	28	ชาย	น้อย	ปริญญาตรี	ไม่ทำงาน	ไม่
12	28	ชาย	น้อย	ปริญญาโท	ทำงาน	ใช่
13	41	ชาย	น้อย	มัธยม	ไม่ทำงาน	ไม่
14	53	หญิง	น้อย	มัธยม	ทำงาน	ไม่
15	57	หญิง	น้อย	ปริญญาตรี	ไม่ทำงาน	ไม่
16	41	ชาย	น้อย	มัธยม	ไม่ทำงาน	ไม่
17	20	ชาย	น้อย	มัธยม	ไม่ทำงาน	ใช่
18	39	หญิง	น้อย	ปริญญาโท	ทำงาน	ใช่
19	19	ชาย	น้อย	มัธยม	ทำงาน	ไม่
20	41	ชาย	มาก	ปริญญาตรี	ไม่ทำงาน	ใช่
21	61	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
22	47	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
23	55	หญิง	น้อย	ปริญญาตรี	ทำงาน	ใช่
24	19	หญิง	มาก	ปริญญาโท	ทำงาน	ไม่
25	38	หญิง	มาก	ปริญญาโท	ทำงาน	ไม่
26	50	หญิง	มาก	ปริญญาตรี	ไม่ทำงาน	ไม่
27	29	ชาย	น้อย	ปริญญาโท	ทำงาน	ไม่
28	39	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
29	61	ชาย	น้อย	ปริญญาตรี	ไม่ทำงาน	ใช่
30	42	ชาย	น้อย	มัธยม	ไม่ทำงาน	ไม่
31	44	หญิง	มาก	มัธยม	ไม่ทำงาน	ไม่
32	59	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ไม่
33	45	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
34	33	ชาย	มาก	ปริญญาตรี	ทำงาน	ไม่
35	32	ชาย	น้อย	ปริญญาโท	ทำงาน	ไม่
36	64	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
37	61	หญิง	น้อย	ปริญญาตรี	ทำงาน	ไม่
38	28	ชาย	น้อย	ปริญญาตรี	ไม่ทำงาน	ไม่

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
# อ่านไฟล์ Excel
df = pd.read_excel('decision_tree_dataset01.xlsx')
```

Part01 :

Lec06_Decision_Tree_Churn01.py

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

แปลงค่าข้อความเป็นตัวเลข

```
label_encoders = {}
```

```
for col in ['เพศ', 'รายได้', 'การศึกษา', 'การทำงาน', 'Churn']:
```

```
    le = LabelEncoder()
```

```
    df[col] = le.fit_transform(df[col])
```

```
    label_encoders[col] = le # เก็บตัวแปลงค่าไว้ใช้ภายหลัง
```

แยกคุณลักษณะและเป้าหมาย

```
X = df.drop('Churn', axis=1)
```

```
y = df['Churn']
```

แบ่งข้อมูล train/test

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Part02 :

Lec06_Decision_Tree_Churn01.py

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

สร้างและฝึกโมเดล Decision Tree

Part03 :

Lec06_Decision_Tree_Churn01.py

```
model = DecisionTreeClassifier(max_depth=5, random_state=42)
```

```
model.fit(X_train, y_train)
```

ข้อมูล unseen 3 รายการ

```
unseen_data = pd.DataFrame([
```

```
    {'อายุ': 30, 'เพศ': 'ชาย', 'รายได้': 'มาก', 'การศึกษา': 'ปริญญาตรี', 'การทำงาน': 'ทำงาน'},
```

```
    {'อายุ': 50, 'เพศ': 'หญิง', 'รายได้': 'น้อย', 'การศึกษา': 'มัธยม', 'การทำงาน': 'ไม่ทำงาน'},
```

```
    {'อายุ': 40, 'เพศ': 'ชาย', 'รายได้': 'น้อย', 'การศึกษา': 'ปริญญาโท', 'การทำงาน': 'ทำงาน'}]
```

```
)
```

แปลงข้อมูล unseen ให้เป็นตัวเลข

```
for col in ['เพศ', 'รายได้', 'การศึกษา', 'การทำงาน']:
```

```
    unseen_data[col] = label_encoders[col].transform(unseen_data[col])
```

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

ทำนายผล

```
unseen_predictions = model.predict(unseen_data)
```

แปลงผลลัพธ์กลับเป็นข้อความ

```
unseen_data['Churn'] = label_encoders['Churn'].inverse_transform(unseen_predictions)
```

แสดงผลลัพธ์

```
print(unseen_data)
```

Part04 :

Lec06_Decision_Tree_Churn01.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> & C:/Users/ec06_DT_Decision_tree_Churn.py
```

```
อายุ เพศ รายได้ การศึกษา การทำงาน Churn
```

```
0 30 0 1 0 0 ไม่
```

```
1 50 1 0 2 1 ไม่
```

```
2 40 0 0 1 0 ไม่
```

```
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> []
```


A	B	C	D	E	F
อายุ	เพศ	รายได้	การศึกษา	การทำงาน	Churn
56	ชาย	มาก	มัธยม	ไม่ทำงาน	ใช่
46	หญิง	น้อย	ปริญญาตรี	ทำงาน	ใช่
32	หญิง	มาก	ปริญญาโท	ไม่ทำงาน	ใช่
60	หญิง	มาก	มัธยม	ไม่ทำงาน	ใช่
25	ชาย	มาก	มัธยม	ไม่ทำงาน	ไม่
38	ชาย	มาก	ปริญญาโท	ทำงาน	ไม่
56	ชาย	น้อย	ปริญญาโท	ทำงาน	ใช่
36	ชาย	มาก	ปริญญาโท	ไม่ทำงาน	ใช่
40	ชาย	น้อย	ปริญญาโท	ไม่ทำงาน	ใช่

```
unseen_data = pd.DataFrame([
{'อายุ': 30, 'เพศ': 'ชาย', 'รายได้': 'มาก', 'การศึกษา': 'ปริญญาตรี', 'การทำงาน': 'ทำงาน'},
{'อายุ': 50, 'เพศ': 'หญิง', 'รายได้': 'น้อย', 'การศึกษา': 'มัธยม', 'การทำงาน': 'ไม่ทำงาน'},
{'อายุ': 40, 'เพศ': 'ชาย', 'รายได้': 'น้อย', 'การศึกษา': 'ปริญญาโท', 'การทำงาน': 'ทำงาน'},
{'อายุ': 22, 'เพศ': 'ชาย', 'รายได้': 'มาก', 'การศึกษา': 'มัธยม', 'การทำงาน': 'ทำงาน'}
])
# แปลงข้อมูล unseen ให้เป็นตัวเลข
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Data_Mining_Project> & C:/Users/ASUS/AppData/Local
    อายุ  เพศ  รายได้  การศึกษา  การทำงาน  Churn
0      30    0      1          0          0    ไม่
1      50    1      0          2          1    ใช่
2      40    0      0          1          0    ใช่
3      22    0      1          2          0    ไม่
PS D:\Data_Mining_Project>
```

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

```
import pandas as pd
```

Part01 :

Lec06_DT_Churn01.py

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.metrics import accuracy_score, classification_report,  
confusion_matrix
```

```
from sklearn import preprocessing
```

```
# สร้าง DataFrame สำหรับตัวอย่าง
```

หมายเหตุ : ไฟล์นี้จะมีทั้งหมด 4 ส่วนคือ part01 - part04

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
df = pd.read_excel('decision_tree_dataset01.xlsx')
X = df.drop('Churn', axis=1)
y = df['Churn']
cat_cols = ['เพศ', 'รายได้', 'การฝึกหัด', 'การทำงาน']
num_cols = ['อายุ']
preprocess = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), cat_cols),
        ('num', 'passthrough', num_cols)
    ]
)
```

Part01 :

Lec06_DT_Churn02.py

```

model = DecisionTreeClassifier(max_depth=5, random_state=42, class_weight='balanced')
pipe = Pipeline(steps=[('preprocess', preprocess),
                        ('model', model)])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
pipe.fit(X_train, y_train)
unseen_data = pd.DataFrame([
    {'อายุ': 30, 'เพศ': 'ชาย', 'รายได้': 'มาก', 'การศึกษา': 'มัธยม', 'การทำงาน': 'ไม่ทำงาน'},
    {'อายุ': 50, 'เพศ': 'หญิง', 'รายได้': 'น้อย', 'การศึกษา': 'มัธยม', 'การทำงาน': 'ไม่ทำงาน'},
    {'อายุ': 22, 'เพศ': 'หญิง', 'รายได้': 'มาก', 'การศึกษา': 'ปริญญาโท', 'การทำงาน': 'ไม่ทำงาน'}
])
pred = pipe.predict(unseen_data)
unseen_data['Churn'] = pred
print(unseen_data)

```

Part02 :

Lec06_DT_Churn02.py

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Data_Mining_Project> & C:/Users/ASUS/AppData/Local/Microsoft
• อายุ เพศ รายได้ การศึกษา การทำงาน Churn
0 30 ชาย มาก มัธยม ไม่ทำงาน ไม่
1 50 หญิง น้อย มัธยม ไม่ทำงาน ไม่
2 22 หญิง มาก ปริญญาโท ไม่ทำงาน ใช่
○ PS D:\Data_Mining_Project>

```

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

Part02 :

Lec06_DT_Churn01.py

```
data = {  
    'อายุ': [25, 35, 22, 30, 21, 45, 32, 28, 50, 40],  
    'เพศ': ['ชาย', 'หญิง', 'ชาย', 'หญิง', 'ชาย', 'หญิง', 'ชาย', 'หญิง', 'ชาย', 'หญิง'],  
    'รายได้': ['น้อย', 'มาก', 'มาก', 'น้อย', 'มาก', 'มาก', 'น้อย', 'มาก', 'น้อย', 'มาก'],  
    'การซื้อสินค้า': ['ไม่ซื้อ', 'ซื้อ', 'ซื้อ', 'ไม่ซื้อ', 'ซื้อ', 'ไม่ซื้อ', 'ซื้อ', 'ซื้อ', 'ไม่ซื้อ', 'ซื้อ']  
}  
  
df = pd.DataFrame(data)  
# แปลงข้อมูลแบบ categorical เป็นตัวเลข  
le = preprocessing.LabelEncoder()
```

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

Part03 :

Lec06_DT_Churn01.py

```
df['เพศ'] = le.fit_transform(df['เพศ'])
df['รายได้'] = le.fit_transform(df['รายได้'])
df['การซื้อสินค้า'] = le.fit_transform(df['การซื้อสินค้า'])
# แบ่งข้อมูลเป็น train และ test
X = df[['อายุ', 'เพศ', 'รายได้']]
y = df['การซื้อสินค้า']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# สร้างต้นไม้ตัดสินใจ
tree_model = DecisionTreeClassifier(random_state=42)
tree_model.fit(X_train, y_train)
```

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

ทำนายบนข้อมูลทดสอบ

```
y_pred = tree_model.predict(X_test)
```

วัดประสิทธิภาพ

```
accuracy = accuracy_score(y_test, y_pred)
```

```
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
class_report = classification_report(y_test, y_pred)
```

```
print(f'Accuracy: {accuracy}')
```

```
print(f'Confusion Matrix:\n{conf_matrix}')
```

```
print(f'Classification Report:\n{class_report}')
```

Part04 :

Lec06_DT_Churn01.py

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

วิเคราะห์ผลลัพธ์

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/L
Accuracy: 1.0
Confusion Matrix:
[[1 0]
 [0 1]]
Classification Report:
              precision    recall  f1-score   support

     0           1.00        1.00        1.00         1
     1           1.00        1.00        1.00         1

 accuracy          1.00          1.00          1.00         2
  macro avg           1.00          1.00          1.00         2
 weighted avg          1.00          1.00          1.00         2

PS C:\AppServ\www\Python_DataMining>
```

การวิเคราะห์ผลลัพธ์จากการทดสอบโมเดลต้นไม้ตัดสินใจที่คุณได้แสดงมีลักษณะดังนี้:

1. Accuracy: 1.0 (หรือ 100%)

- Accuracy คือสัดส่วนของการทำนายที่ถูกต้องทั้งหมดต่อจำนวนทั้งหมดของการทำนาย
- ในกรณีนี้, โมเดลทำนายถูกทั้งหมด, ค่า accuracy เท่ากับ 1.0 หรือ 100%

2. Confusion Matrix:

- มีขนาด 2x2 แสดงการจำแนกที่ทำนายถูกและทำนายผิด
- มีตัวเลข 1 อยู่ในแท่งที่ (1, 1) และ (2, 2) แสดงว่าทั้ง 2 กรณีทั้งหมดถูกทำนายถูกต้อง

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

วิเคราะห์ผลลัพธ์

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/L
Accuracy: 1.0
Confusion Matrix:
[[1 0]
 [0 1]]
Classification Report:
              precision    recall  f1-score   support

     0               1.00      1.00      1.00         1
     1               1.00      1.00      1.00         1

 accuracy               1.00                1.00         2
 macro avg              1.00      1.00      1.00         2
 weighted avg           1.00      1.00      1.00         2

PS C:\AppServ\www\Python_DataMining> |
```

3. Classification Report:

- Precision: คือสัดส่วนของ True Positive (TP) ต่อ (TP + False Positive (FP))
- Recall (Sensitivity): คือสัดส่วนของ TP ต่อ (TP + False Negative (FN))
- F1-Score: เป็นค่าเฉลี่ยที่นำ precision และ recall มารวมกัน
- Support: คือจำนวนของตัวอย่างในแต่ละคลาส
- Macro Avg: คือค่าเฉลี่ยของ precision, recall, และ F1-score ที่คำนวณโดยไม่คำนึงถึงขนาดของแต่ละคลาส
- Weighted Avg: คือค่าเฉลี่ยของ precision, recall, และ F1-score ที่คำนวณโดยคำนึงถึงขนาดของแต่ละคลาส

ในกรณีนี้, ผลลัพธ์ทั้งหมดทำให้เห็นว่าโมเดลทำนายทุกรายการถูกต้องทั้งหมด, ซึ่งส่งผลให้ accuracy เป็น 1.0 หรือ 100 %. อย่างไรก็ตาม, ควรระมัดระวังเมื่อวิเคราะห์ผลลัพธ์นี้, โดยการตรวจสอบข้อมูลทดสอบและระวังปัญหา overfitting

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

วิเคราะห์ผลลัพธ์

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/L
Accuracy: 1.0
Confusion Matrix:
[[1 0]
 [0 1]]
Classification Report:
              precision    recall  f1-score   support

     0           1.00        1.00        1.00         1
     1           1.00        1.00        1.00         1

 accuracy          1.00          1.00          1.00         2
  macro avg          1.00          1.00          1.00         2
 weighted avg          1.00          1.00          1.00         2

PS C:\AppServ\www\Python_DataMining> |
```

ในทางทฤษฎี, ผลลัพธ์ที่ได้มีความแม่นยำเต็มรูปแบบ (accuracy เป็น 1.0) และค่า precision, recall, และ F1-score ที่สูงสุด (ทั้งในแต่ละคลาส) จะมีสิ่งที่เรียกว่า "overfitting"

Overfitting เกิดขึ้นเมื่อโมเดลทำนายข้อมูลที่ใช้ในการฝึก (training data) ได้ถูกต้องเกินไปถึงขั้นที่ไม่สามารถทำนายข้อมูลทดสอบ (test data) ได้ดีเท่านั้น. ในกรณีที่ accuracy เป็น 1.0 บนข้อมูลทดสอบ, มีความเสี่ยงที่โมเดลได้ทำการจำแนกข้อมูลที่ใช้ในการฝึกได้ดีเกินไป (overfit) และอาจไม่สามารถทำนายข้อมูลที่ไม่เคยเห็นมาก่อนได้

6.7.4 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

วิเคราะห์ผลลัพธ์

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/L
Accuracy: 1.0
Confusion Matrix:
[[1 0]
 [0 1]]
Classification Report:
              precision    recall  f1-score   support

     0           1.00        1.00        1.00         1
     1           1.00        1.00        1.00         1

 accuracy          1.00          1.00          1.00         2
  macro avg           1.00          1.00          1.00         2
 weighted avg          1.00          1.00          1.00         2

PS C:\AppServ\www\Python_DataMining> |
```

การป้องกัน overfitting มีหลายวิธี, บางวิธีที่สามารถลองทำได้รวมถึง:

1. การใช้ข้อมูลทดสอบแยกออกมาในขั้นตอนการฝึก: ใช้ข้อมูลทดสอบที่ไม่เคยมีในขั้นตอนการฝึกเพื่อวัดประสิทธิภาพของโมเดล
2. การใช้ Cross-Validation: ใช้เทคนิค cross-validation เพื่อให้การวัดประสิทธิภาพมีความเท่าเทียมกัน.
3. การลดความซับซ้อนของโมเดล: ลดจำนวนของ feature หรือลดความซับซ้อนของโมเดล

การทดลองด้วยข้อมูลทดสอบที่ไม่เคยเห็นมาก่อนและการใช้ cross-validation เป็นวิธีที่ช่วยลดความเสี่ยงจาก overfitting

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> &
_Learning/DataMining/DM_Projects/Lec06_DT_Churn02.py
Accuracy: 0.68
Confusion Matrix:
[[ 12  32]
 [ 32 124]]
Cla
```

ผลลัพธ์ของการทดสอบโมเดล Decision Tree ที่คุณได้รับแสดงรายละเอียดดังนี้:

Accuracy: 0.68

ความแม่นยำของโมเดลที่ 68% หมายความว่าโมเดลทำนายผลถูกต้อง 68% ของข้อมูลทั้งหมด แต่ในกรณีที่ข้อมูลไม่สมดุล (มีความแตกต่างในจำนวนตัวอย่างของแต่ละคลาส) ค่า accuracy อาจไม่ใช่ตัวชี้วัดที่ดีที่สุดในการประเมินประสิทธิภาพของโมเดล

Confusion Matrix:

[12 32]: จาก 44 ตัวอย่างในคลาส 0 (ไม่ Churn), โมเดลทำนายถูกต้อง 12 ครั้งและทำนายผิดเป็นคลาส 1 จำนวน 32 ครั้ง

[32 124]: จาก 156 ตัวอย่างในคลาส 1 (Churn), โมเดลทำนายถูกต้อง 124 ครั้งและทำนายผิดเป็นคลาส 0 จำนวน 32 ครั้ง

Confusion Matrix นี้บ่งบอกว่าโมเดลมีปัญหาในการทำนายคลาส 0 (ไม่ Churn) และมักจะทำนายเป็นคลาส 1 (Churn) มากกว่า

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

- ตัวอย่างการประยุกต์ใช้งาน decision tree สำหรับ พนักงานสินเชื่อของธนาคารต้องการที่จะทำการวิเคราะห์ข้อมูลเพื่อที่จะทำการศึกษาว่าการกู้ยืมในครั้งหนึ่ง ๆ มีครั้งไหนบ้างที่ปลอดภัยและครั้งไหนบ้างที่มีความเสี่ยง
- ในตารางนี้ มีข้อมูลเกี่ยวกับอายุ, รายได้, การศึกษา, การทำงาน, และการกู้ยืมของพนักงาน. ข้อมูล "การกู้ยืม" เป็นตัวแปรที่เราต้องการทำนายว่าจะปลอดภัยหรือไม่ปลอดภัย. ข้อมูลนี้สามารถใช้ในการฝึก Decision Tree และทดสอบความสามารถในการทำนาย

ตัวอย่างข้อมูล

อายุ	รายได้	การศึกษา	การทำงาน	การกู้ยืม
25	น้อย	ปริญญาตรี	ทำงาน	ปลอดภัย
35	มาก	มัธยม	ไม่ทำงาน	ไม่ปลอดภัย
22	มาก	ปริญญาโท	ทำงาน	ไม่ปลอดภัย
30	น้อย	ปริญญาตรี	ไม่ทำงาน	ปลอดภัย
21	มาก	มัธยม	ทำงาน	ไม่ปลอดภัย
45	มาก	ปริญญาโท	ทำงาน	ปลอดภัย
32	น้อย	มัธยม	ไม่ทำงาน	ไม่ปลอดภัย
28	มาก	ปริญญาตรี	ทำงาน	ปลอดภัย
50	น้อย	มัธยม	ทำงาน	ปลอดภัย
40	มาก	ปริญญาโท	ไม่ทำงาน	ไม่ปลอดภัย

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

- | อายุ | รายได้ | การศึกษา | การทำงาน | การกู้ยืม |
- |-----|-----|-----|-----|-----|
- | 38 | น้อย | ปริญญาตรี | ทำงาน | ? |
- | 28 | มาก | มัธยม | ไม่ทำงาน | ? |
- | 33 | มาก | ปริญญาโท | ทำงาน | ? |

- ผลลัพธ์ของการทำนายข้อมูล unknow 2 ชุดคือ
- ข้อมูล unknown1: ข้อมูลนี้มีรายได้สูง การศึกษาระดับปริญญาตรี และทำงานอยู่ ซึ่งตรงกับเงื่อนไขที่โมเดลตัดสินใจให้กู้ยืม จึงให้ผลลัพธ์ว่า "ปลอดภัย"
- ข้อมูล unknown2: ข้อมูลนี้มีรายได้ต่ำ การศึกษาระดับมัธยม และไม่ได้ทำงานอยู่ ซึ่งตรงกับเงื่อนไขที่โมเดลตัดสินใจไม่ให้กู้ยืม จึงให้ผลลัพธ์ว่า "ไม่ปลอดภัย"

ตัวอย่างข้อมูล

อายุ	รายได้	การศึกษา	การทำงาน	การกู้ยืม
38	น้อย	ปริญญาตรี	ทำงาน	?
28	มาก	มัธยม	ไม่ทำงาน	?
33	มาก	ปริญญาโท	ทำงาน	?

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

ผลลัพธ์ของการทำนายข้อมูล unknow 2 ชุดคือ

ข้อมูล unknown1: ข้อมูลนี้มีรายได้สูง การศึกษาระดับปริญญาตรี และทำงานอยู่ ซึ่งตรงกับเงื่อนไขที่โมเดลตัดสินใจให้ถูก จึงให้ผลลัพธ์ว่า "ปลอดภัย"

ข้อมูล unknown2: ข้อมูลนี้มีรายได้ต่ำ การศึกษาระดับมัธยม และไม่ได้ทำงานอยู่ ซึ่งตรงกับเงื่อนไขที่โมเดลตัดสินใจไม่ให้ถูก จึงให้ผลลัพธ์ว่า "ไม่ปลอดภัย"

ตัวอย่างข้อมูล

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

Accuracy: 1.0

Confusion Matrix:

```
[[1 0]
 [0 1]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	1.00	1.00	1.00	1
accuracy			1.00	2
macro avg	1.00	1.00	1.00	2
weighted avg	1.00	1.00	1.00	2

```
C:\Users\User\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8
; returning scalar instead, but in the future will perform elementwise comparison
```

```
mask |= (ar1 == a)
```

```
การกู้ยืมสำหรับข้อมูล unknown1: [1]
```

```
การกู้ยืมสำหรับข้อมูล unknown2: [1]
```

```
PS C:\AppServ\www\Python_DataMining>
```

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

1. ในการใช้ Decision Tree สำหรับการวินิจฉัยโรคในภาษา Python นั้น คุณสามารถใช้ไลบรารี scikit-learn ได้เป็นที่เรียบร้อยแล้ว เนื่องจาก scikit-learn มีฟังก์ชันการสร้าง Decision Tree ที่สะดวกและมีประสิทธิภาพมากมาย
2. ตัวอย่างด้านล่างนี้จะแสดงตัวอย่างการใช้ Decision Tree Classifier ใน scikit-learn สำหรับการวินิจฉัยโรคโดยใช้ข้อมูลที่มีอยู่แล้ว

ตัวอย่างข้อมูล

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

```
# Import ไลบรารีที่จำเป็น
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn import datasets
# โหลดข้อมูล
data = datasets.load_iris()
X = data.data
y = data.target
# แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# สร้าง Decision Tree Classifier
clf = DecisionTreeClassifier()
# ฝึกโมเดล
clf.fit(X_train, y_train)
# ทำนายผลลัพธ์ในชุดทดสอบ
```

Part01 :

Lec06_DT_Sick.py

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

```
y_pred = clf.predict(X_test)
```

```
# ประเมินประสิทธิภาพ
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print(f'Accuracy: {accuracy:.2f}')
```

```
# แสดงรายงานการประเมิน
```

```
print("\nClassification Report:")
```

```
print(classification_report(y_test, y_pred))
```

```
# สมมติข้อมูล unknown
```

```
unknown_data = [[5.0, 3.5, 1.5, 0.2]] # ปรับค่าตามลักษณะของข้อมูล iris
```

```
# ทำนายผลลัพธ์
```

```
prediction = clf.predict(unknown_data)
```

```
# แสดงผลลัพธ์
```

```
print(f'Prediction for unknown data: {prediction}')
```

Part02 :

Lec06_DT_Sick.py

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

```
1. # สมมติข้อมูล unknown
2. unknown_data = [[5.0, 3.5, 1.5, 0.2]] # ปรับค่าตาม
   ลักษณะของข้อมูล iris
3. # ทำนายผลลัพธ์
4. prediction = clf.predict(unknown_data)
5. # แสดงผลลัพธ์
6. print(f'Prediction for unknown data:
   {prediction}')
```

ตัวอย่างข้อมูล

Accuracy: 1.00

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	9
2	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Prediction for unknown data: [0]

6.7.5 การประยุกต์ใช้งาน ต้นไม้ตัดสินใจ (Decision Tree) ในภาษาไพธอน

1. Accuracy (ความแม่นยำ): 1.00 หมายถึงโมเดลมีความแม่นยำทั้งในชุดฝึกและชุดทดสอบเท่ากับ 100%.
2. Classification Report (รายงานการประเมิน):
 1. Precision (ความแม่นยำของการทำนาย): ทุกระดับคลาสมี precision เท่ากับ 1.00, หมายถึงไม่มีการทำนายคลาสนั้นๆ ผิดพลาด.
 2. Recall (ความแม่นยำในการค้นค่า): ทุกระดับคลาสมี recall เท่ากับ 1.00, หมายถึงทุกรายการที่เป็นของคลาสนั้นถูกต้อง.
 3. F1-score (คะแนน F1): ทุกระดับคลาสมี F1-score เท่ากับ 1.00, หมายถึงการรวมความแม่นยำของการทำนายและการค้นค่า.
 4. Support (จำนวนข้อมูลในแต่ละคลาส): แสดงจำนวนข้อมูลในแต่ละคลาส.
3. Prediction for unknown data (การทำนายสำหรับข้อมูลที่รู้จัก): โมเดลทำนายว่าข้อมูล unknown อยู่ในคลาสที่ 0.
 - ในรายงานการประเมิน, ค่า Precision, Recall, และ F1-score ทุกระดับคลาสนี้เท่ากับ 1.00 ทำให้เห็นว่าโมเดลมีประสิทธิภาพมากในการทำนาย. การทำนายสำหรับข้อมูลที่รู้จัก (unknown data) ได้ว่าอยู่ในคลาสที่ 0.

ตัวอย่างข้อมูล

Accuracy: 1.00

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	9
2	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Prediction for unknown data: [0]

6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

ตัวอย่างข้อมูล สำหรับการทำต้นไม้ตัดสินใจ

Part01 :

Lec06_DT_decision_tree_loan.py

ข้อมูลที่เหมาะสมสำหรับการทำต้นไม้ตัดสินใจ (Decision Tree) ต้องประกอบด้วยคุณลักษณะ (features) และป้ายกำกับ (labels) ที่เกี่ยวข้องกัน. นี่คือนตัวอย่างข้อมูลที่สามารถใช้ในการสร้างโมเดลต้นไม้ตัดสินใจ:

ตัวอย่าง: การตัดสินใจว่าลูกค้าซื้อประกันหรือไม่

สมมติว่าคุณมีข้อมูลเกี่ยวกับลูกค้าที่มาซื้อประกันรถยนต์ โดยมีคุณลักษณะต่าง ๆ ที่สามารถมีผลต่อการตัดสินใจของลูกค้าที่จะซื้อประกันหรือไม่:

อายุ	รายได้	ประวัติเครดิต	ประสบการณ์ขับขี่	ป้ายทะเบียนรถ	การซื้อประกัน
25	50000	ดี	3 ปี	มี	ซื้อ
30	80000	ปานกลาง	5 ปี	ไม่มี	ไม่ซื้อ
22	35000	ไม่ดี	1 ปี	มี	ไม่ซื้อ
45	120000	ดี	10 ปี	มี	ซื้อ
28	60000	ปานกลาง	4 ปี	ไม่มี	ซื้อ

6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

ตัวอย่างข้อมูล สำหรับการทำต้นไม้ตัดสินใจ

Part01 : Lec06_DT_decision_tree_loan.py

ในที่นี้: อายุ, รายได้, ประวัติเครดิต, ประสิทธิภาพขับขี่, และป้ายทะเบียนรถ เป็นคุณลักษณะของลูกค้า (features).
การซื้อประกัน เป็นป้ายกำกับ (label) ที่ต้องการทำนาย.

โมเดลต้นไม้ตัดสินใจจะพยายามสร้างกฎตามคุณลักษณะเหล่านี้เพื่อทำนายว่าลูกค้าใดจะซื้อประกันและลูกค้าใดไม่ซื้อ.
หลังจากฝึกฝนโมเดล, คุณสามารถใช้มันในการทำนายการตัดสินใจของลูกค้าใหม่ที่ไม่มีป้ายกำกับ

อายุ	รายได้	ประวัติเครดิต	ประสิทธิภาพขับขี่	ป้ายทะเบียนรถ	การซื้อประกัน
25	50000	ดี	3 ปี	มี	ซื้อ
30	80000	ปานกลาง	5 ปี	ไม่มี	ไม่ซื้อ
22	35000	ไม่ดี	1 ปี	มี	ไม่ซื้อ
45	120000	ดี	10 ปี	มี	ซื้อ
28	60000	ปานกลาง	4 ปี	ไม่มี	ซื้อ

6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

Part01 : Lec06_DT_decision_tree_loan.py

```
# Import ไลบรารีที่จำเป็น
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import preprocessing
# กำหนดข้อมูล
data = {
    'อายุ': [25, 30, 22, 45, 28],
    'รายได้': [50000, 80000, 35000, 120000, 60000],
    'ประวัติเครดิต': ['ดี', 'ปานกลาง', 'ไม่ดี', 'ดี', 'ปานกลาง'],
    'ประสบการณ์ขับขี่': ['3 ปี', '5 ปี', '1 ปี', '10 ปี', '4 ปี'],
    'ป้ายทะเบียนรถ': ['มี', 'ไม่มี', 'มี', 'มี', 'ไม่มี'],
    'การซื้อประกัน': ['ซื้อ', 'ไม่ซื้อ', 'ไม่ซื้อ', 'ซื้อ', 'ซื้อ']
}
```

```
# แปลงข้อมูลจากข้อมูลแบบ dictionary เป็น DataFrame
import pandas as pd

df = pd.DataFrame(data)
# แปลงข้อมูลตัวแปรที่เป็นข้อความเป็นตัวเลข

le = preprocessing.LabelEncoder()

df['ประวัติเครดิต'] = le.fit_transform(df['ประวัติเครดิต'])
df['ประสบการณ์ขับขี่'] = le.fit_transform(df['ประสบการณ์ขับขี่'])
df['ป้ายทะเบียนรถ'] = le.fit_transform(df['ป้ายทะเบียนรถ'])
df['การซื้อประกัน'] = le.fit_transform(df['การซื้อประกัน'])
# แบ่งข้อมูลเป็นชุดฝึกฝนและชุดทดสอบ

X = df.drop('การซื้อประกัน', axis=1)
y = df['การซื้อประกัน']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

Part02 : Lec06_DT_decision_tree_loan.py

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
# สร้างและฝึกฝนโมเดล Decision Tree

model = DecisionTreeClassifier()

model.fit(X_train, y_train)
# ทำนายผลบนชุดทดสอบ

y_pred = model.predict(X_test)
# คำนวณความแม่นยำ

accuracy = accuracy_score(y_test, y_pred)

print(f'ความแม่นยำ: {accuracy}')
```

```
# นำโมเดล Decision Tree ไปใช้งาน
new_data = pd.DataFrame({
    'อายุ': [35],
    'รายได้': [70000],
    'ประวัติเครดิต': ['ดี'],
    'ประสบการณ์ขับขี่': ['6 ปี'],
    'ป้ายทะเบียนรถ': ['มี']
})

new_data['ประวัติเครดิต'] = le.transform(new_data['ประวัติเครดิต'])
new_data['ประสบการณ์ขับขี่'] = le.transform(new_data['ประสบการณ์ขับขี่'])
new_data['ป้ายทะเบียนรถ'] = le.transform(new_data['ป้ายทะเบียนรถ'])
prediction = model.predict(new_data)

print(f'การทำนาย: {prediction}')
```


6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

ข้อมูล Iris เป็นชุดข้อมูลทดสอบที่ถูกใช้ในงาน Machine Learning และสถิติทางการคำนวณเพื่อตรวจสอบและทดสอบวิธีการแบบต่าง ๆ ในการจำแนกข้อมูลหรือคาดการณ์.

ชุดข้อมูล Iris ได้รับชื่อจากชนิดของดอกไม้ Iris ที่มีทั้งหมด 3 ชนิด: Iris setosa, Iris versicolor, และ Iris virginica. แต่ละชนิดมีลักษณะทางกายภาพที่แตกต่างกัน.

ข้อมูล Iris มักถูกนำมาใช้ในการฝึกสอนและทดสอบแบบจำลองการจำแนก (classification) เพื่อสร้าง Decision Tree, k-Nearest Neighbors, Support Vector Machines, หรือแบบจำลองอื่น ๆ ที่ใช้ในการจำแนกประเภทข้อมูล.

ผลลัพธ์ที่แสดงเป็นข้อมูลที่ได้จากการทดสอบแบบจำลอง Decision Tree ใน Python โดยใช้ข้อมูล Iris โดยสรุปดังนี้:

ตัวอย่างลักษณะทางกายภาพของดอกไม้ Iris 3 ชนิด:

Iris setosa: มีดอกไม้ที่เล็กและใบกลม.

Iris versicolor: มีดอกไม้ขนาดกลางและใบเรียบ.

Iris virginica: มีดอกไม้ใหญ่และใบยาว.

ชุดข้อมูล Iris ประกอบด้วยลักษณะทางกายภาพของดอกไม้ต่าง ๆ เช่น ความยาวและความกว้างของกลีบดอก (sepal) และกลีบดอกจริง (petal) ของ Iris 3 ชนิด. ข้อมูลนี้ทำให้เป็นที่น่าสนใจสำหรับการศึกษาและทดลองแบบจำลองที่ใช้ในการจำแนกประเภทข้อมูล.

6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

File : Lec06_DT_Iris.py

```
# นำเข้าไลบรารีที่จำเป็น
from sklearn.datasets import load_iris
from sklearn.model_selection import
train_test_split
from sklearn.tree import
DecisionTreeClassifier, export_text
# โหลดข้อมูล Iris
iris = load_iris()
X = iris.data
y = iris.target
# แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
# สร้างแบบจำลอง Decision Tree
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
# ทดสอบโมเดล
accuracy = clf.score(X_test, y_test)
print(f"Accuracy: {accuracy:.2f}")
# แสดงลักษณะของ Decision Tree
tree_rules = export_text(clf,
feature_names=iris.feature_names)
print("Decision Tree Rules:\n", tree_rules)
```

6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

File :

Lec06_DT_Iris.py

Accuracy: 1.00

Decision Tree Rules:

```
|--- petal width (cm) <= 0.80
|   |--- class: 0
|--- petal width (cm) > 0.80
|   |--- petal length (cm) <= 4.75
|   |   |--- petal width (cm) <= 1.65
|   |   |   |--- class: 1
|   |   |   |--- petal width (cm) > 1.65
|   |   |   |   |--- class: 2
|   |   |--- petal length (cm) > 4.75
|   |   |   |--- petal width (cm) <= 1.75
|   |   |   |   |--- petal length (cm) <= 4.95
|   |   |   |   |   |--- petal width (cm) <= 1.55
|   |   |   |   |   |   |--- class: 2
```

```
| | | | |--- petal width (cm) > 1.55
| | | | |   |--- sepal length (cm) <= 6.95
| | | | |   |   |--- class: 1
| | | | |   |   |--- sepal length (cm) > 6.95
| | | | |   |   |   |--- class: 2
| | | | |   |   |   |--- petal width (cm) > 1.75
| | | | |   |   |   |   |--- petal length (cm) <= 4.85
| | | | |   |   |   |   |   |--- sepal width (cm) <= 3.10
| | | | |   |   |   |   |   |   |--- class: 2
| | | | |   |   |   |--- sepal width (cm) > 3.10
| | | | |   |   |   |   |--- class: 1
| | | | |   |   |   |   |--- petal length (cm) > 4.85
| | | | |   |   |   |   |   |--- class: 2
```

6.7.6 การประยุกต์ใช้งานต้นไม้ตัดสินใจ (Decision Tree) ในการปล่อยสินเชื่อ(Loan) ในภาษาไพธอน

ผลลัพธ์

File :

Lec06_DT_Iris.py

ผลลัพธ์ที่แสดงเป็นข้อมูลที่ได้จากการทดสอบแบบจำลอง Decision Tree ใน Python โดยใช้ข้อมูล Iris โดยสรุป ดังนี้:

Accuracy: 1.00

ความแม่นยำของโมเดลที่ทดสอบบนชุดข้อมูลทดสอบมีค่าเท่ากับ 1.00 หรือ 100%, ซึ่งหมายถึงโมเดลทำนายถูกทุกรายการในชุดข้อมูลทดสอบ.

Decision Tree Rules:

แสดงลักษณะของ Decision Tree ที่ได้จาก การฝึกสอนด้วยข้อมูล Iris.

ตัวอย่าง: ถ้า petal width (cm) ≤ 0.80 แล้วก็จะ เป็นคลาส 0 (Iris setosa), แต่ถ้า petal width (cm) > 0.80 และ petal length (cm) ≤ 4.75 แล้วจะตรวจสอบเงื่อนไขเพิ่มเติม.

การอธิบาย Decision Tree สามารถทำได้ตามลำดับของเงื่อนไขที่ปรากฏในผลลัพธ์:

ถ้า petal width (cm) ≤ 0.80 แล้วคลาสคือ 0 (Iris setosa).

ถ้า petal width (cm) > 0.80 และ petal length (cm) ≤ 4.75 :

ถ้า petal width (cm) ≤ 1.65 แล้วคลาสคือ 1 (Iris versicolor).

ถ้า petal width (cm) > 1.65 แล้วคลาสคือ 2 (Iris virginica).

ถ้า petal width (cm) > 0.80 และ petal length (cm) > 4.75 :

ถ้า petal width (cm) ≤ 1.75 :

ถ้า petal length (cm) ≤ 4.95 แล้วคลาสคือ 1 (Iris versicolor).

ถ้า petal length (cm) > 4.95 และ petal width (cm) ≤ 1.55 แล้วคลาสคือ 2 (Iris virginica).

ถ้า petal length (cm) > 4.95 และ petal width (cm) > 1.55 แล้วตรวจสอบเงื่อนไขเพิ่มเติม.

... และสายเวลาของ Decision Tree จะทำงานไปเรื่อย ๆ ตามเงื่อนไขที่กำหนด. ข้อมูลใน Decision Tree Rules นี้เป็นตัวช่วยในการอธิบายการตัดสินใจของ Decision Tree ในการจำแนกข้อมูล Iris ออกเป็นกลุ่มต่าง ๆ ตามลักษณะของดอกไม้.

6.8 ตัววัดประสิทธิภาพของโมเดล (Classification)

- **Confusion Matrix** เป็นการทำนายไว้ว่าถูกผิดเท่าไร เป็นตาราง ขนาดเท่ากับจำนวนแถวคูณจำนวนคอลัมน์ที่มี กำหนดสถานะผลดังนี้
 - True Positive (TP)
 - True Negative (TN)
 - False Positive (FP)
 - False Negative (FN)

6.8 ตัววัดประสิทธิภาพของโมเดล (Classification)

- Precision ดูสิ่งที่เรา Predict ออกมา แล้วทายถูกได้ที่เปอร์เซ็นต์ ผลการคำนวณได้ทายถูกที่ตัวจากผลที่ได้ออกมาจากตัวอย่าง Precision สำหรับ normal มีค่า True Positiveหาร ด้วย TP+FP เช่น $4/7 \times 100 = 57.12\%$
Precision สำหรับ spam มีค่าเท่ากับ $6/8 \times 100 = 75\%$
- Recall จำนวนที่ทำนายถูกที่ตัว ความถูกต้องของแต่ละคำตอบในการทายว่าสิ่งที่ต้องการคืออะไร ต้องการทำนายจำนวนคนที่ป่วยเยอะกว่า จะดูได้จาก Recall ออกมาว่าตัวไหนให้คำตอบ มากกว่าน้อยกว่าอีกตัวจะมาช่วยในการทำนายชื่อ

6.8 ตัววัดประสิทธิภาพของโมเดล (Classification) (ต่อ)

- **F-Measure** ช่วยหาค่าเฉลี่ยของ Precision และ Recall เช่นคำนวณจากค่าเฉลี่ยของทั้งสองก่อนหน้าคือ $2 \times \text{Precision} \times \text{Recall}$ หรือ $\text{Precision} + \text{Recall}$
- **Accuracy** จำนวนข้อมูลที่ทำนายถูกของทุกคลาส จากผลตัวอย่างที่มีการถูกต้องทั้งหมด $10/15 \times 100 = 66.6\%$
- **ROC Graph & Area** แสดงกราฟความสัมพันธ์ ทำนายถูกไปทางแกนตั้ง (Y) ถ้าทายผิดไปแนวแกนนอน (X) มากสุดอยู่ด้านล่าง จะเริ่มจากด้านล่างก่อน มีเส้นขึ้นมาจากเบอร์ 12 (หน้า 380) ถ้ามีข้อมูลผิดเกิดเป็นแนวนอนในเบอร์ 7 กราฟที่ได้จะขึ้นมาแบบบันได ขึ้นมาเป็น Curve คำถามคือ ถ้าเราทำนายถูกเยอะ ๆ กราฟจะขึ้นไปข้างบน เป็นโมเดลที่ดี แต่ถ้าเป็นเส้นทะแยงมุม กราฟเริ่มไม่ดีแล้ว แสดงถึงข้อมูลไม่ค่อยดี จากตัวอย่างจะเห็นได้จากคอลัมน์ที่ทำนายได้ผลเพิ่มขึ้นทีละ 0.2 กราฟจะค่อยๆ เพิ่มขึ้นเรื่อยๆ ในคอลัมน์ 3-4
- **ROC Curve** มีค่าเข้าใกล้ 1 จะแสดงว่ามีประสิทธิภาพดีกว่า จากกราฟ
- **Area Under Curve** พื้นที่ใต้กราฟ (AUC) ถ้าพื้นที่ใต้กราฟเข้าใกล้ 1 จะมีพื้นที่ใต้กราฟมาก ฉะนั้นมีค่ามาก (เข้าใกล้ 1) จะยิ่งดี
- สรุปแล้วเป็นตัววัดประสิทธิภาพเหล่านี้จะช่วยมองในมุมมองต่าง ๆ ได้

6.9 การแบ่งข้อมูลสำหรับการสอนและการทดสอบ

ชุดข้อมูลสอน (Training Dataset)

ชุดข้อมูลทดสอบ (Testing Dataset)



6.9 การแบ่งข้อมูลสำหรับการสอนและการทดสอบ

การแบ่งกลุ่มข้อมูลสำหรับการฝึก (training) และข้อมูลทดสอบ (testing) เป็นขั้นตอนสำคัญในการสร้างและประเมินโมเดลทางคอมพิวเตอร์ ภาษาไพธอนมีหลายวิธีที่สามารถใช้ในการทำนายนี้นี้ต่อไปนี้คือตัวอย่างการใช้ `train_test_split` จากไลบรารี `scikit-learn`:

```
from sklearn.model_selection import train_test_split
```

```
# สร้างข้อมูลตัวอย่าง
```

```
X, y = ... # กำหนดข้อมูล feature (X) และ label (y)
```

```
# แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

6.9 การแบ่งข้อมูลสำหรับการสอนและการทดสอบ

การแบ่งกลุ่มข้อมูลสำหรับการฝึก (training) และข้อมูลทดสอบ (testing) เป็นขั้นตอนสำคัญในการสร้างและประเมินโมเดลทางคอมพิวเตอร์ ภาษาไพธอนมีหลายวิธีที่สามารถใช้ในการทำนายนี้ต่อไปนี้คือตัวอย่างการใช้ `train_test_split` จากไลบรารี `scikit-learn`:

โดยทั่วไปแล้ว:

X คือ feature หรือตัวแปรอิสระที่ใช้ในการทำนาย.

y คือ label หรือตัวแปรตามที่ต้องการทำนาย.

test_size คือ สัดส่วนของข้อมูลที่จะถูกใช้เป็นข้อมูลทดสอบ.

random_state คือ seed ที่ใช้ในการสุ่มข้อมูล, ทำให้การแบ่งข้อมูลเป็นไปตามลำดับเดียวกันทุกครั้ง.

6.9 การแบ่งข้อมูลสำหรับการสอนและการทดสอบ

การแบ่งกลุ่มข้อมูลสำหรับการฝึก (training) และข้อมูลทดสอบ (testing) เป็นขั้นตอนสำคัญในการสร้างและประเมินโมเดลทางคอมพิวเตอร์ ภาษาไพธอนมีหลายวิธีที่สามารถใช้ในการทำนายนี ต่อไปนี้คือตัวอย่างการใช้ `train_test_split` จากไลบรารี `scikit-learn`:

```
# สมมติมีข้อมูลขนาด 1000 รายการ
X, y = np.arange(1000).reshape((500, 2)), range(500)

# แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

ตามตัวอย่างนี้, `X_train` และ `y_train` จะเป็นข้อมูลที่ใช้ฝึกโมเดล, และ `X_test` และ `y_test` จะเป็นข้อมูลที่ใช้ทดสอบโมเดล.

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล

วิธีการทดสอบโมเดลแบ่งออกเป็น 3 ตัว

- **Self-consistency test (use training set)** เอา Training Data มาเป็นตัว Test เลย คือ ใช้ข้อมูลเดิมแล้วมันมีความถูกต้องมากน้อยแค่ไหน
- **Split test** แบ่งออกเป็น 2 ส่วนเลย คือ เป็นโมเดล และส่วนทดสอบ เช่น 70% สร้างโมเดล 30% ใช้ทดสอบ หรือ 80:20 ใช้ข้อมูล 2 ชุด เป็น Training data สำหรับสร้าง Model และ Testing Data สำหรับทดสอบ ถ้า Sampling มาดีก็ได้ข้อมูลทดสอบดี Split Test จะดีเมื่อมีข้อมูลหลายๆ หมั่นหรือแสนตัวขึ้นไป
- **Cross-validation test** ใช้ค่อนข้างเยอะเหมือนกัน การทำ Split หลาย ๆ รอบ การทำหลายๆ กับ Split แต่แบ่งจำนวน N ชุดเท่าๆ กัน เช่น $N=5$, $N=10$ แล้วทำงานโดยการสร้างโมเดลทั้งหมด N ตัว จนครบ Split Test ทำรอบเดียวแล้วจบไป คือเก็บ 1,2 ไว้ แต่ใน Cross จะเอา 3 ไปเป็นตัวทดสอบ แล้วใส่ 2 กลับเข้าไป แล้วเอา 1 เป็นตัวทดสอบ จะพบว่าทุกตัวจะถูกนำมาทดสอบหมด เราจะได้ความถูกต้องเฉลี่ยออกมาในแต่ละรอบ แต่ไม่เหมาะกับการทำแบบนี้กับกรณีที่มีข้อมูลมาก ๆ ไม่เหมาะ

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

ตัวอย่างของ 5-fold cross-validation

มีข้อมูล 5 ส่วนแบ่งออกเป็นกลุ่มเท่า ๆ กัน เหลือง เขียว ฟ้า แดง ม่วง แล้ววนทำให้ครบ 5 รอบ เช่นรอบที่ 1 ทำแล้วเก็บเป็นตัวอย่าง Test ไว้ รอบที่ 2 เอาสีเขียวใส่เข้าไป แล้วทดสอบด้วยกลุ่มสีเหลือง กาแบบนั้นวนไปให้ครบทุกรอบสี แล้วสุดท้ายเอาทุกสีมาเฉลี่ยกัน คาทามจะเลือกใช้โมเดลไหนดีจากผลการรันทดสอบ ดูตัวโมเดลที่ได้ตัวไหนให้ผลเฉลี่ยออกมาได้ดีที่สุดเลือกตัวนั้น แนวทางที่ 2 ไม่ใช้งาน 5 ตัวนี้เลย แต่จะเอาทั้งหมด หนึ่งแสนตัวมาสร้างโมเดลอีกที คือ ถ้ามีข้อมูล 5 แสนตัว แต่ละรอบแบ่งโมเดล แสนตัว กับแปดแสนตัว (หน้า 395)

สรุปคือวัดประสิทธิภาพก่อนแล้วหาเทคนิคไหนเหมาะสม ระหว่าง A ,B, C ถ้าได้เทคนิคไหนดีที่สุดในที่สุดก็เอาไปสร้างเป็นโมเดลที่ใช้งานจริง

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

```
from sklearn.model_selection import cross_val_score, KFold
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
import numpy as np
```

```
# สร้างข้อมูลตัวอย่าง (แบบจำลองของคุณ)
```

```
X = [[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12], [13, 14], [15, 16]]
```

```
y = [0, 0, 0, 0, 1, 1, 1, 1]
```

```
# สร้างแบบจำลอง (ในที่นี้ใช้ RandomForestClassifier เป็นตัวอย่าง)
```

```
model = RandomForestClassifier()
```

```
# กำหนด KFold สำหรับ 5-fold cross-validation
```

```
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
```

```
# ใช้ cross_val_score เพื่อทำ 5-fold cross-validation
```

```
# นับคะแนนทั้งหมดที่ได้จากแต่ละ fold
```

Part01

File : dm_06_crossValidation_5fold.py

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

```
scores = cross_val_score(model, X, y, cv=kfold)
```

```
# แสดงผลลัพธ์คะแนนทั้งหมด
```

```
print("Cross-Validation Scores:", scores)
```

```
# คำนวณค่าเฉลี่ยของคะแนน
```

```
average_score = np.mean(scores)
```

```
print("Average Cross-Validation Score:", average_score)
```

Part02

File : dm_06_crossValidation_5fold.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/Local/Micr
Cross-Validation Scores: [1.  1.  0.5 1.  1. ]
Average Cross-Validation Score: 0.9
PS C:\AppServ\www\Python_DataMining> █
```

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

ตัวอย่างนี้:

File :: File : dm_06_crossValidation_5fold.py

X คือ features (ตัวแปรอิสระ)

y คือ labels (ตัวแปรตาม)

RandomForestClassifier คือแบบจำลองที่เราใช้ในตัวอย่างนี้

KFold ถูกกำหนดให้ทำ 5-fold cross-validation, **shuffle=True** คือการสลับข้อมูลก่อนแบ่ง, **random_state** คือการกำหนด seed สำหรับการสุ่ม

cross_val_score ใช้สำหรับทำ cross-validation และคืนคะแนนที่ได้จากแต่ละ fold

หลังจากที่ทำ cross-validation เสร็จสิ้น, เราสามารถดูคะแนนทั้งหมดและคำนวณค่าเฉลี่ยของคะแนนได้.

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/Local/Micro
Cross-Validation Scores: [1. 1. 0.5 1. 1. ]
Average Cross-Validation Score: 0.9
PS C:\AppServ\www\Python_DataMining> █
```

File :: File : dm_06_crossValidation_5fold.py

คะแนน cross-validation ที่คุณได้คือ [1. 1. 0.5 1. 1.] และค่าเฉลี่ยของคะแนน cross-validation คือ 0.9.

การได้คะแนนเฉลี่ยสูงแสดงว่าแบบจำลองของคุณมีประสิทธิภาพที่ดีในการทำนายข้อมูล โดย cross-validation เป็นเทคนิคที่ดีในการประเมินประสิทธิภาพของแบบจำลองเนื่องจากมันช่วยลดความเอนเอียง (bias) และความคลาดเคลื่อน (variance) ของการประเมินประสิทธิภาพ.

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

```
from sklearn.model_selection import cross_val_score, KFold
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
import numpy as np
```

```
# สร้างข้อมูลตัวอย่าง (แบบจำลองของคุณ)
```

```
X = [[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12], [13, 14], [15, 16],  
[17, 18], [19, 20]]
```

```
y = [0, 0, 0, 0, 1, 1, 1, 1, 1, 1]
```

```
# สร้างแบบจำลอง (ในที่นี้ใช้ RandomForestClassifier เป็นตัวอย่าง)
```

```
model = RandomForestClassifier()
```

```
# กำหนด KFold สำหรับ 10-fold cross-validation
```

Part01

File : dm_06_crossValidation_10fold.py

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

```
kfold = KFold(n_splits=10, shuffle=True, random_state=42)
# ใช้ cross_val_score เพื่อทำ 10-fold cross-validation
```

```
# นับคะแนนทั้งหมดที่ได้จากแต่ละ fold
```

Part02

File : dm_06_crossValidation_10folde.py

```
scores = cross_val_score(model, X, y, cv=kfold)
```

```
# แสดงผลลัพธ์คะแนนทั้งหมด
```

```
print("Cross-Validation Scores:", scores)
```

```
# คำนวณค่าเฉลี่ยของคะแนน
```

```
average_score = np.mean(scores)
```

```
print("Average Cross-Validation Score:", average_score)
```

6.10 Validation การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล(ต่อ)

ผลลัพธ์

File : dm_06_crossValidation_10folde.py

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/Local/Micr
Cross-Validation Scores: [1. 1. 1. 1. 1. 1. 1. 0. 1. 1.]
Average Cross-Validation Score: 0.9
PS C:\AppServ\www\Python_DataMining> |
```

โปรแกรมทำงานได้ถูกต้องและได้คะแนน cross-validation ที่ดี. คะแนนทั้งหมดคือ [1. 1. 1. 1. 1. 1. 1. 0. 1. 1.] และค่าเฉลี่ยของคะแนน cross-validation คือ 0.9

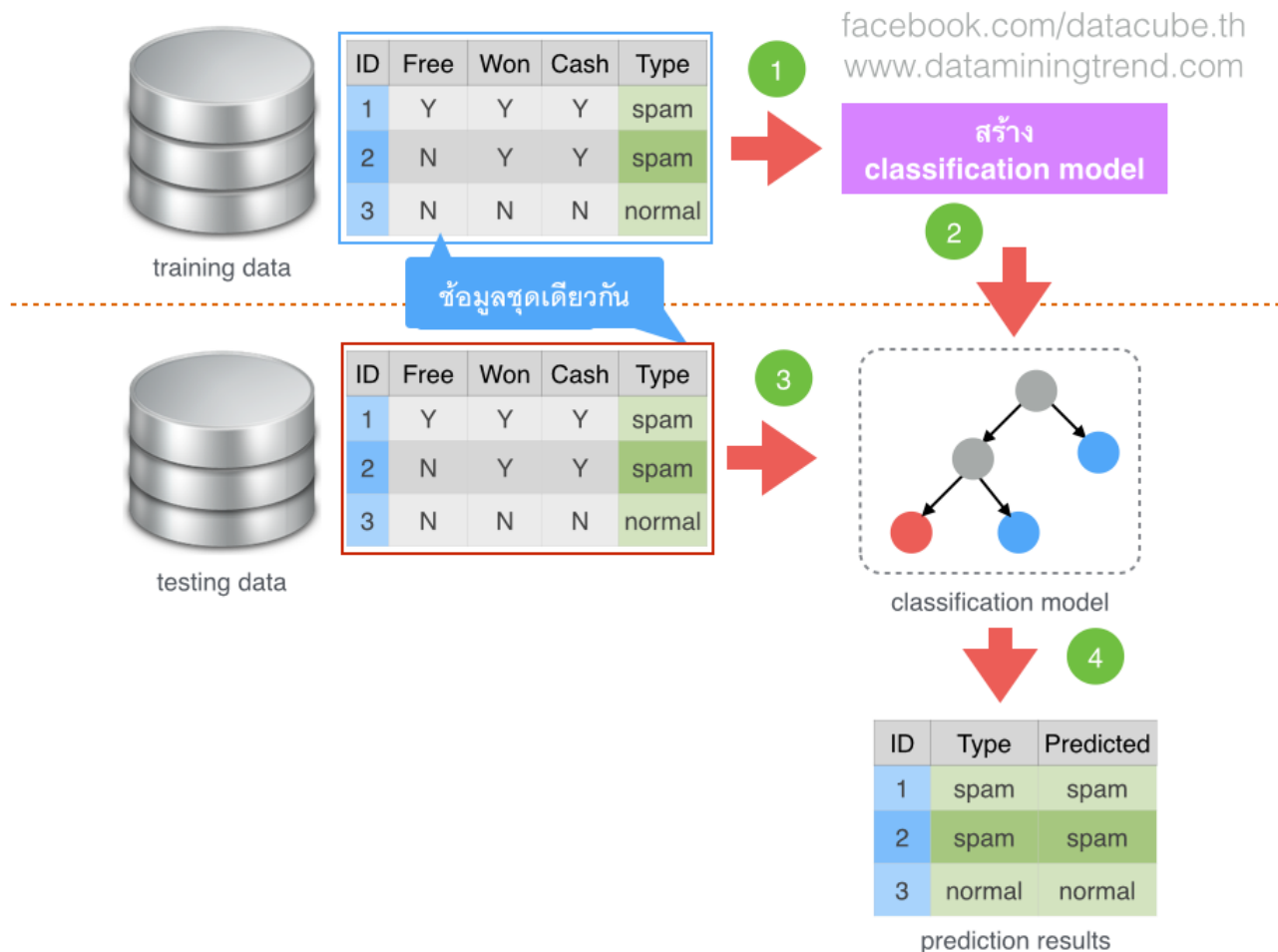
คะแนน 1 แสดงถึงความถูกต้องทั้งหมดในแต่ละ fold ในทำนายข้อมูลทดสอบ. คะแนน 0 แสดงถึงมีความผิดพลาดในการทำนายในบาง fold. ดังนั้น, ค่าเฉลี่ย 0.9 หมายถึงมีความถูกต้องทั้งหมดในส่วนมากของ fold

ทั้งนี้ผลลัพธ์ขึ้นอยู่กับข้อมูลและการสร้างแบบจำลองของคุณ. ควรตรวจสอบและปรับแต่งตัวแบบหรือข้อมูลในกรณีที่ต้องการประสิทธิภาพที่ดีขึ้น

Self-consistency Test

(data)³
base | warehouse | mining

- ใช้ข้อมูล training ในการทดสอบประสิทธิภาพของโมเดล

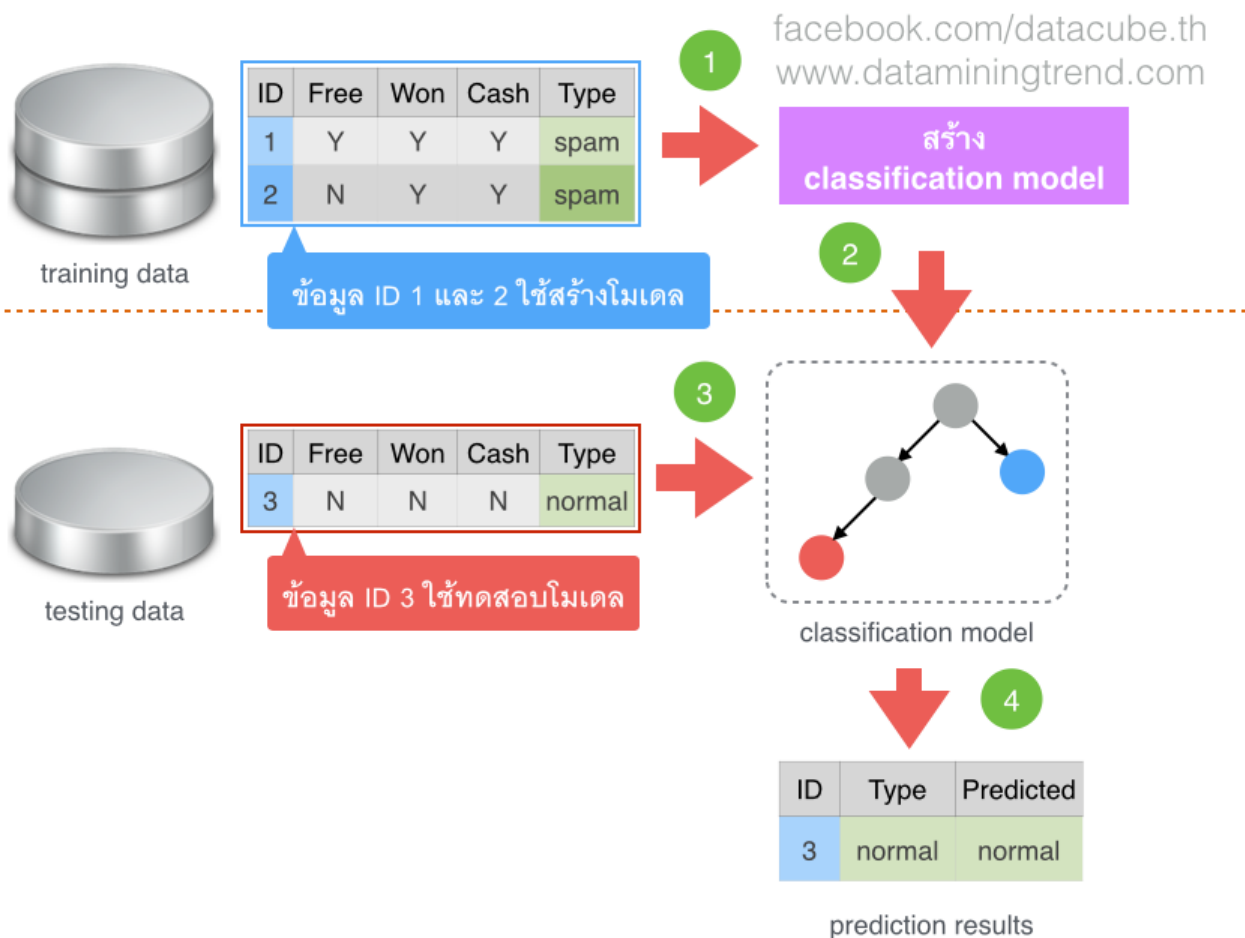


วิธี Self Consistency Test หรือบางครั้งเรียกว่า Use Training Set นี้เป็นวิธีการที่ง่ายที่สุด นั่นคือ ข้อมูลที่ใช้ในการสร้างโมเดล (model) และข้อมูลที่ใช้ในการทดสอบโมเดลเป็นข้อมูลชุดเดียวกัน กระบวนการนี้เริ่มจาก สร้างโมเดลด้วยข้อมูลเทรนนิ่งดาต้า (training data) หลังจากนั้นนำโมเดลที่สร้างได้มาทำนายข้อมูลเทรนนิ่งดาต้าชุดเดิม ตัวอย่างเช่นจากรูป นำข้อมูลเทรนนิ่งดาต้าในตาราง มาสร้างโมเดลและทดสอบโมเดลเป็นต้น การวัด ประสิทธิภาพด้วยวิธีนี้จะให้ผลการวัดประสิทธิภาพที่มีค่าสูงมาก (อาจจะเข้าใกล้ 100%) เนื่องจากเป็นข้อมูล ชุดเดิมที่ระบบได้ทำการเรียนรู้มาแล้ว แต่ผลการวัดที่ได้ไม่เหมาะที่จะนำไปรายงานในงานวิจัยต่าง ๆ ซึ่งวิธี การนี้เหมาะสำหรับใช้ในการทดสอบประสิทธิภาพเพื่อดูแนวโน้มของโมเดลที่สร้างขึ้น ถ้าได้ผลการวัดที่น้อย แสดงว่าโมเดลไม่เหมาะสมกับข้อมูล จึงไม่ควรจะนำไปทดสอบด้วยวิธีการแบ่งข้อมูลแบบต่าง ๆ

Split Test

(data)³
base | warehouse | mining

- แบ่งข้อมูลออกเป็น 2 ชุด
 - training data สำหรับสร้างโมเดล และ testing data สำหรับทดสอบ



Split Test

วิธี Split Test เป็นการแบ่งข้อมูลด้วยการสุ่มออกเป็น 2 ส่วน เช่น 70% ต่อ 30% หรือ 80% ต่อ 20% โดยข้อมูลส่วนที่หนึ่ง (70% หรือ 80%) ใช้ในการสร้างโมเดลและข้อมูลส่วนที่สอง (30% หรือ 20%) ใช้ในการทดสอบประสิทธิภาพของโมเดล ตัวอย่างเช่นในรูป แบ่งข้อมูลเทรนนิ่ง ดาต้า ในตารางออกเป็น 2 ตัวอย่างสำหรับการสร้างโมเดลและข้อมูล 1 ตัวอย่างใช้ในการทดสอบ ประสิทธิภาพของโมเดล เป็นต้น แต่การทดสอบแบบ Split Test นี้ทำการสุ่มข้อมูลเพียงครั้งเดียวซึ่งในบาง ครั้งถ้าการสุ่มข้อมูลที่ใช้ในการทดสอบที่มีลักษณะคล้ายกับข้อมูลที่ใช้สร้างโมเดลทำให้ผลการวัด ประสิทธิภาพได้ออกมาดี ในทางตรงข้ามถ้าการสุ่มข้อมูลที่ใช้ในการทดสอบที่มีลักษณะแตกต่างกับข้อมูลที่ใช้สร้างโมเดลมากทำให้ผลการวัด ประสิทธิภาพได้ออกมาแย่ ดังนั้นจึงควรใช้วิธี Split Test นี้หรือทำการสุ่ม หลายๆ ครั้ง แต่ข้อดีของวิธีการนี้คือใช้เวลาในการสร้างโมเดลน้อยซึ่งเหมาะกับชุดข้อมูลที่มีขนาดใหญ่มากครับ

Cross-validation

(data)³
base | warehouse | mining

- แบ่งข้อมูลออกเป็น N ชุด เช่น $N = 5$ หรือ 10
- ข้อมูล $N-1$ ชุดสำหรับสร้างโมเดล และ ข้อมูลส่วนที่เหลือสำหรับทดสอบ วนทำจนครบ N

รอบที่ 1



training data

ID	Free	Won	Cash	Type
2	N	Y	Y	spam
3	N	N	N	normal

ข้อมูล ID 2 และ 3 ใช้สร้างโมเดล

1

facebook.com/datacube.th
www.dataminingtrend.com

สร้าง
classification model

2

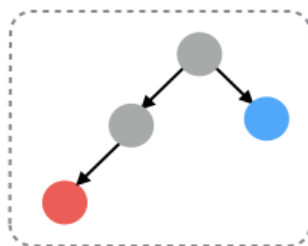


testing data

ID	Free	Won	Cash	Type
1	Y	Y	Y	spam

ข้อมูล ID 1 ใช้ทดสอบโมเดล

3



classification model



4

ID	Type	Predicted
1	spam	spam

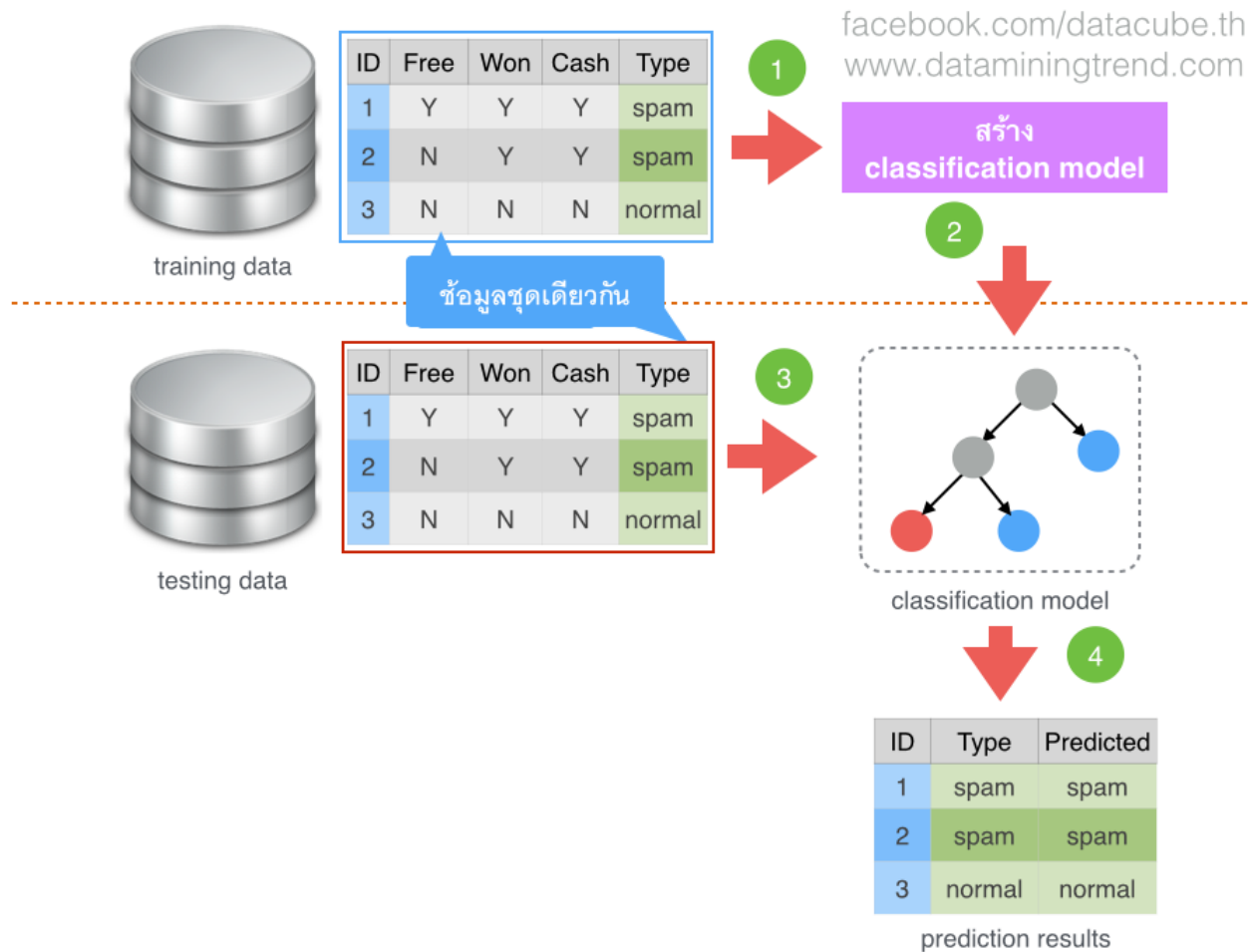
prediction results

(3) วิธี Cross-validation Test

วิธีนี้เป็นวิธีที่นิยมในการทำงานวิจัย เพื่อใช้ในการทดสอบประสิทธิภาพของโมเดลเนื่องจากผลที่ได้มีความน่าเชื่อถือ การวัด ประสิทธิภาพด้วยวิธี Cross-validation นี้จะทำการแบ่งข้อมูลออกเป็นหลายส่วน (มักจะแสดงด้วยค่า k) เช่น 5-fold cross-validation คือ ทำการแบ่งข้อมูลออกเป็น 5 ส่วน โดยที่แต่ละส่วนมีจำนวนข้อมูลเท่ากัน หรือ 10-fold cross-validation คือ การแบ่งข้อมูลออกเป็น 10 ส่วน โดยที่แต่ละส่วนมีจำนวนข้อมูลเท่ากัน หลังจากนั้นข้อมูลหนึ่งส่วนจะใช้เป็นตัวทดสอบ ประสิทธิภาพของโมเดล ทำวนไปเช่นนี้จนครบจำนวนที่แบ่งไว้ เช่น การทดสอบด้วยวิธี 5-fold cross-validation ในรูปด้านล่างครับ

Self-consistency Test

- ใช้ข้อมูล training ในการทดสอบประสิทธิภาพของโมเดล



ความแตกต่างระหว่าง Classification & Regression

งานส่วนใหญ่ของ Data Mining จะเป็นในการทำ Classification ซึ่งสามารถพบในชีวิตประจำวัน เช่น การพยากรณ์อากาศ , เรื่อง Speech recognition , face recognition การอัฟโหลดรูป แล้วทำนายว่าเป็นหน้าใคร เช่น ในเฟสบุ๊ก , Spam e-mail

ตัวอย่างการทำ Spam e-mail classification

- ให้ระบุว่า e-mail ใหนบ้างที่เป็น spam e-mail เช่น Keyword ที่มีคำว่า won , FREE,cash ในบทรึ้นนำความคิดมาสร้างเป็นโมเดลแล้วสร้างการทำนายข้อมูลที่มีมาใหม่ได้
- หา Keyword ที่ใช้บ่งบอกเป็น spam-email แล้วสร้างเป็นตารางประกอบด้วยคอลัมน์ FREE, Won, Cash , Type แล้วกำหนด Keyword Y,N (หน้า 356)
- สร้างเป็นโมเดล (Classification model) ได้ จากข้อมูล training data ซึ่งมีลาเบล (label) เป็นคำตอบไว้ก่อน บางที่เรียกว่าลาเบล หรือคลาส ถ้าลาเบลของเรื่องทำนายผิดตก คือ ตกหรือไม่ตก นศ.คนนี้ควรเรียนวิชาไหนดี ลาเบลคือคำตอบวิชา ดังนั้นลาเบลคือคำตอบของการทำนายนั้น ๆ ตัวใหนที่เป็นคำตอบมันจะอยู่ล่างสุดของ Decision Tree โดยคำตอบเริ่มจากใหนดบนสุดก่อนไล่มาล่างสุดก็จะเจอคำตอบ สิ่งที่ต้องคำนึงถึงคือ ถ้าคำตอบเปลี่ยนโมเดลก็จะเปลี่ยนด้วย ถ้าข้อมูลมีมากขึ้นอาจจะมีข้อมูลที่คำนวณมากขึ้นตาม
- นำข้อมูลใหม่ (unseen data) ทำนายโดยใช้โมเดล จากตัวอย่างเบอร์ 11 และ 12 เป็น Spam ทั้งคู่

สรุป เอา Training Data Model

Spam e-mail

ข้อมูลสำหรับ spam e-mail

ID	won	free	cash	Bank	Transfer	Lottory	type
1	y	y	y	n	y	y	Yes
2	n	n	n	n	n	y	No
					y	n	Yes

phishing website.com

From: Busuang Banking >
To: user@gmail.com > Hide

Enquiry from Busuang Service
18 January BE 2562 02:26

Warning Message : Your account is currently blocked at 17/01/2019 19:26:04

Dear customer,
Your account registered in the Busuang iPay service has been blocked because of invalid data.
To unblock your account please re-register with your valid data
Click the following link to re-register :
<https://pay.hangkokbank.com/re-register/>

Thanks ,

อีเมลหลอกลวง

Technical Support

itchul@phishing.com
2015-03-05 09:52

To: User Webmail;
Dear Web-mail User,
Your Mail quota has reached limit, You might not be able to send or get updates until you re-validate your mailbox.
To re-validate your mailbox reply to this mail and fill your.
(user-name :
(Password :
(Confirm Password
or Visit
www.it.chula.ac.th
www.it.chula.ac.th
-Technical Support

al Message -----

ve a new eFax from 478-743-8656
5 22:34

aller-ID: 478-743-8656]

ed an 1 page fax on Thu, 5 Mar 2015 16:34:49 +0100 .
ur fax attached to this email.
number for this fax is 63562625

is required to view the attached fax document.
sing the eFax service!

ความแตกต่างระหว่าง Classification & Regression (ต่อ)

Regression คำตอบจะเป็นตัวเลข

Classification คือตอบคำไม่ใช่ตัวเลข เช่น คลาส , ลาเบล Y,N

ขั้นตอนการสร้างโมเดล

- Training Data
- Testing Data

ทั้งสองขั้นตอนมาจาก Training Data รวมใหญ่ทั้งคู่หากผลการทำนายไม่เหมาะสมกับเทคนิคก็ต้องเปลี่ยนเทคนิคใหม่ เอา Test Data มาทดสอบว่าถูกต้องมากน้อยแค่ไหน Prediction

Results ถ้าผลการทำนายออกมาตรงกันก็จะงานได้และวัดประสิทธิภาพผลการทำนายว่าได้ที่เปอร์เซ็นต์

6.4 สิ่งที่ต้องคำนึงถึงในการจำแนกและการทำนายข้อมูล

ในการจำแนกและทำนายข้อมูลเรื่องหนึ่ง ๆ จะมีหลายปัจจัยที่เราต้องพิจารณาและคำนึงถึง แต่อย่างไรก็ตาม โดยทั่วไปเราจะต้องพิจารณา 2 ปัจจัยหลัก ดังนี้

- 1) การจัดเตรียมข้อมูลสำหรับการจำแนกและการทำนายข้อมูล
- 2) การเปรียบเทียบประสิทธิภาพวิธีในการจำแนกและทำนายข้อมูล

6.6 ต้นไม้ตัดสินใจ (Decision Tree)

ต้นไม้ตัดสินใจ หรือการเรียนรู้แบบต้นไม้ตัดสินใจ (อังกฤษ: decision tree learning) เป็นหนึ่งในวิธีการเรียนรู้ซึ่งใช้ในสถิติ, การเรียนรู้ของเครื่อง และการทำเหมืองข้อมูล โดยพิจารณาการสังเกตการแบ่งแยกข้อมูลโดยพิจารณาข้อมูลในการเรียนรู้ของเครื่อง (machine learning) ต้นไม้ตัดสินใจ เป็นโมเดลทางคณิตศาสตร์ที่ใช้ทำนายประเภทของข้อมูล/วัตถุ โดยพิจารณาจากลักษณะของวัตถุ ใหนดภายใน (inner node) ของต้นไม้จะแสดงตัวแปร ส่วนกิ่งจะแสดงค่าที่เป็นไปได้ของตัวแปร ส่วนใบ (leaf node) จะแสดงประเภทของวัตถุ

ต้นไม้ตัดสินใจที่บัพใบแสดงถึงข้อมูลที่เป็นข้อมูลไม่ต่อเนื่อง (discrete values) จะเรียกว่าต้นไม้ตัดสินใจแบบจำแนก (classification trees) และต้นไม้ตัดสินใจที่บัพใบเป็นข้อมูลต่อเนื่อง (continuous values) จะเรียกว่าต้นไม้ตัดสินใจแบบถดถอย (regression trees)

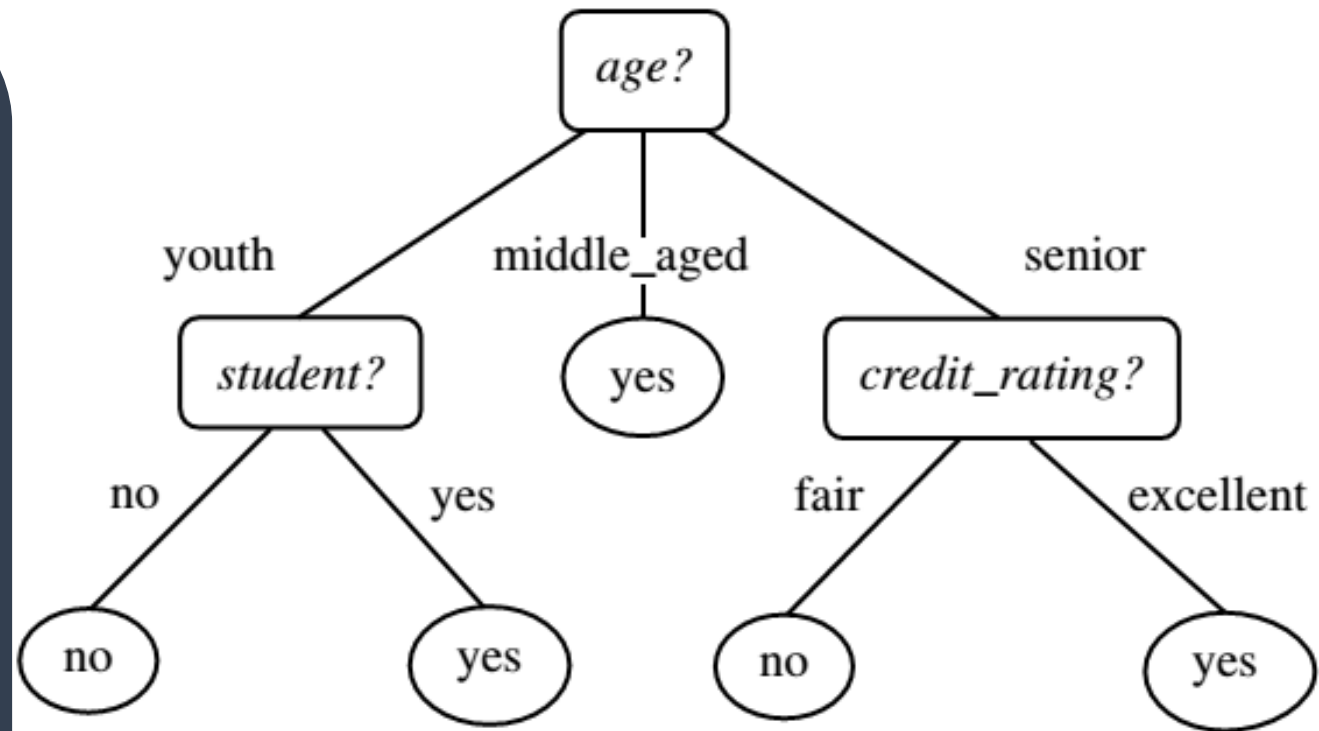
ต้นไม้การตัดสินใจในการบริหารธุรกิจ เป็นแผนผังต้นไม้ช่วยในการตัดสินใจ โดยแสดงถึงมูลค่าของทรัพยากรที่จะใช้ ความเสี่ยงในการลงทุนและผลลัพธ์ที่มีโอกาสเกิดขึ้น ต้นไม้ตัดสินใจสร้างขึ้นเพื่อช่วยการตัดสินใจเพื่อใช้ในการสร้างแผนงาน นิยมใช้มากในการบริหารความเสี่ยง (risk management) ต้นไม้ตัดสินใจเป็นส่วนหนึ่งของทฤษฎีการตัดสินใจ (decision theory) และ ทฤษฎีกราฟ ต้นไม้ตัดสินใจเป็นวิธีการพื้นฐานอย่างหนึ่งสำหรับการทำเหมืองข้อมูล

6.6 การจำแนกข้อมูลด้วยต้นไม้ตัดสินใจ

การจำแนกข้อมูลด้วยต้นไม้ตัดสินใจจะเป็นกระบวนการสร้างต้นไม้ขึ้นเพื่อใช้ในการตัดสินใจจากข้อมูลที่มีหมวดหมู่ข้อมูลแนบอยู่ด้วย ต้นไม้ตัดสินใจจะประกอบไปด้วย **โหนดต่างๆ (ที่ไม่ใช่โหนดใบ—nonleaf node)** ที่ซึ่งถูกใช้ในการแสดงถึงเงื่อนไขหรือแอตทริบิวต์หนึ่ง ๆ ของข้อมูล โดยที่แต่ละกิ่งก้านของโหนดหนึ่ง ๆ จะหมายถึงค่าที่เป็นไปได้จากการทดสอบกับแอตทริบิวต์นั้น ๆ และจะประกอบไปด้วย **โหนดใบ (leaf node)** ที่ซึ่งจะมีหมวดหมู่ข้อมูลจัดเก็บอยู่ โดยตัวอย่างต้นไม้ตัดสินใจจะถูกแสดงในภาพที่ 5.4 ที่จะแสดงการทำนายคุณลักษณะของลูกค้าที่จะทำการซื้อคอมพิวเตอร์จากร้านขายอุปกรณ์ไฟฟ้า โดยโหนดต่าง ๆ ที่ไม่ใช่โหนดใบจะถูกแทนด้วยสี่เหลี่ยม และโหนดใบจะถูกแทนด้วยวงรี ตามลำดับ จากภาพที่ 5.4 เราจะเห็นว่าโหนดใบจะเป็นโหนดที่บ่งบอกถึงข้อมูลหมวดหมู่ของคำตอบที่เราต้องการ อาทิ เช่น “yes” หมายถึง ลูกค้าจะซื้อคอมพิวเตอร์ และ “no” หมายถึง ลูกค้าจะไม่ซื้อคอมพิวเตอร์ โดยต้นไม้ที่ถูกสร้างขึ้นอาจเป็นต้นไม้ที่มีลักษณะเป็นไบนารีหรือไม่เป็นไบนารีก็ได้

6.6 การจำแนกข้อมูลด้วยต้นไม้ตัดสินใจ (ต่อ)

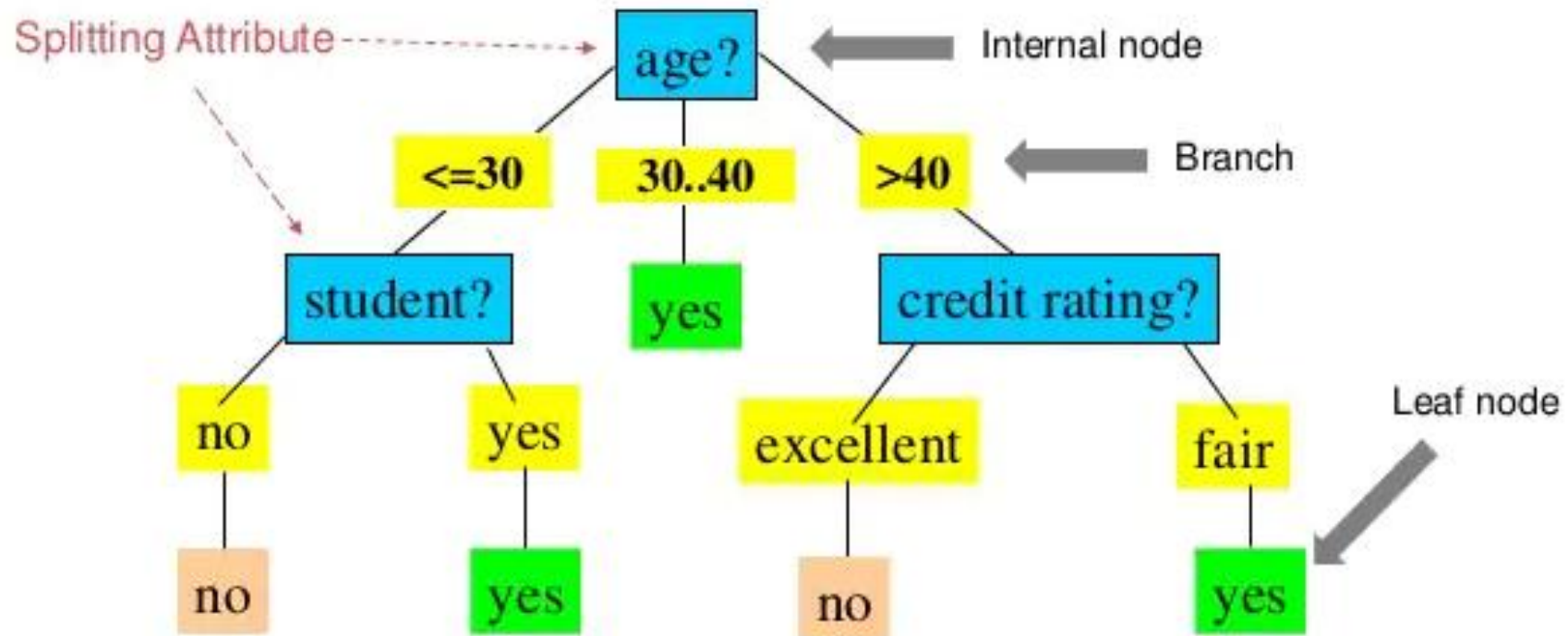
หลังจากทำการสร้างต้นไม้ตัดสินใจแล้ว เราจะสามารถใช้ต้นไม้ตัดสินใจในการจำแนกข้อมูลได้ โดยจะทดสอบการจำแนกหมวดหมู่ของข้อมูลเรคคอร์ดหนึ่ง ๆ (ที่ประกอบไปด้วยแอตทริบิวต์ต่าง ๆ แต่เราจะไม่ทราบหมวดหมู่ข้อมูลในเรคคอร์ดนั้น ๆ) ด้วยการเปรียบเทียบแอตทริบิวต์ที่อยู่ในโหนดรากกับค่าของแอตทริบิวต์ในเรคคอร์ดที่พิจารณา โดยจะทำการเปรียบเทียบจากโหนดรากไปจนถึงโหนดใบ เมื่อเราทราบถึงโหนดใบจะทำให้เราทราบถึงหมวดหมู่ข้อมูลของเรคคอร์ดที่ทำการพิจารณา



ภาพที่ 6.4 ตัวอย่างต้นไม้ตัดสินใจสำหรับการจำแนกคุณลักษณะของลูกค้าที่ทำการซื้อคอมพิวเตอร์

6.6 ต้นไม้ตัดสินใจ (Decision Tree)

ส่วนประกอบของผลลัพธ์ของการเรียนรู้ต้นไม้ตัดสินใจ:



ภาพที่ 6.5 ส่วนประกอบของต้นไม้ตัดสินใจ

ตัวอย่าง

การประยุกต์ใช้ต้นไม้ตัดสินใจ

- 1. การพิจารณาความจงรักภักดี ต่อสินค้า เช่น ลูกค้า iphone มักจะกลับไปใช้บริการหรือซื้อสินค้าของ iPhone (royal) แต่ถ้าลูกค้าเปลี่ยนไปซื้อสินค้ายี่ห้ออื่น เช่น Samsung (churn)**
- 2. กรณีลูกค้าเครือข่าย Ais , DTAC**

- ปัจจัยใด ที่ลูกค้าจะสนใจเลือกซื้อสินค้าชนิดนี้มากที่สุด
- จำนวนคนที่สนใจ และไม่สนใจเป็นเท่าไร (36: 64)

	A	B	C	D	E	F	G	H	I
1	zip code	name	age	lifestyle	family status	car	sports	earnings	label
2	50168	VnSEFOuL	62.0	cozily	married	practical	athletics	102526.0	no response
3	66479	8Tv0hcce	34.0	active	married	expensive	soccer	33006.0	no response
4	16592	Zny9ysbk	69.0	healthy	single	expensive	badminton	118760.0	response
5	50068	HV3xCamM	57.0	cozily	married	practical	soccer	131429.0	response
6	35988	sfIRsQ6v	66.0	cozily	single	practical	badminton	96003.0	response
7	60039	w9voDHj0	21.0	healthy	single	expensive	badminton	83376.0	no response
8	69662	VUKT58V5	22.0	healthy	married	practical	badminton	49826.0	no response
9	40432	HzMvC0c1	59.0	cozily	single	practical	soccer	41166.0	response
10	55994	B9MsYPOR	63.0	active	single	practical	badminton	134340.0	response
11	37460	mWUY9Cjs	31.0	active	married	practical	athletics	108695.0	no response
12	36782	Lavsf4Ms	60.0	cozily	single	practical	badminton	49723.0	response
13	12447	kNAihUVY	38.0	healthy	single	expensive	soccer	105801.0	response
14	56504	FHL6d3WH	36.0	active	married	expensive	athletics	79703.0	no response
15	84343	taO86r6P	30.0	cozily	single	expensive	soccer	68693.0	no response
16	66428	qQSz8WWj	64.0	healthy	single	expensive	badminton	111755.0	no response
17	18272	waFEJp2V	37.0	cozily	married	practical	badminton	36430.0	no response
18	24254	CGEljPBt	61.0	cozily	married	practical	soccer	129420.0	response
19	29470	xQPaw1nF	48.0	cozily	single	expensive	badminton	78714.0	no response
20	178165	1W7J7r0s	17.0	active	married	expensive	soccer	87219.0	no response
21	65878	VzsFKacF	51.0	cozily	married	expensive	badminton	123879.0	no response
22	15047	39OUyKyk	41.0	cozily	married	practical	athletics	137658.0	no response
23	87888	9uXUHkgm	30.0	cozily	single	practical	athletics	89243.0	no response
24	18734	GxQC3KvX	51.0	active	single	expensive	soccer	22977.0	no response

ภาพที่ 5.10 ตัวอย่างข้อมูล จำนวนคนที่สนใจและไม่สนใจ

หลักการ หรือสิ่งที่ต้องพิจารณาเพิ่มเติม

- ข้อมูล กับวิธีการเหมาะสมหรือไม่ จะต้องการเปรียบเทียบกับวิธีจำแนกข้อมูลแบบอื่นอย่างน้อย 3 วิธี (เลือกวิธีที่มีประสิทธิภาพสูง)
- ปริมาณของข้อมูลที่ใช้ในการสอนหรือการทดสอบ มากหรือน้อยเพียงใด
- การเลือกข้อมูลที่เหมาะสม เช่น zip code

- ให้นักศึกษา ใช้วิธีแบ่งข้อมูล split test 70: 30 แล้ว ทำ process และ run ใหม่
- ปัจจัยใด ที่ลูกค้าจะสนใจเลือกซื้อสินค้าชนิดนี้มากที่สุด
- จำนวนคนที่สนใจ และไม่สนใจเป็นเท่าไร (36: 64)

	A	B	C	D	E	F	G	H	I
1	zip code	name	age	lifestyle	family status	car	sports	earnings	label
2	50168	VnSEFOuL	62.0	cozily	married	practical	athletics	102526.0	no response
3	66479	8Tv0hcce	34.0	active	married	expensive	soccer	33006.0	no response
4	16592	Zny9ysbk	69.0	healthy	single	expensive	badminton	118760.0	response
5	50068	HV3xCamM	57.0	cozily	married	practical	soccer	131429.0	response
6	35988	sflRsQ6v	66.0	cozily	single	practical	badminton	96003.0	response
7	60039	w9voDHj0	21.0	healthy	single	expensive	badminton	83376.0	no response
8	69662	VUkT58V5	22.0	healthy	married	practical	badminton	49826.0	no response
9	40432	HzMvC0c1	59.0	cozily	single	practical	soccer	41166.0	response
10	55994	B9MsYPOR	63.0	active	single	practical	badminton	134340.0	response
11	37460	mWUY9Cjs	31.0	active	married	practical	athletics	108695.0	no response
12	36782	Lavsf4Ms	60.0	cozily	single	practical	badminton	49723.0	response
13	12447	kNAihUVY	38.0	healthy	single	expensive	soccer	105801.0	response
14	56504	FHL6d3WH	36.0	active	married	expensive	athletics	79703.0	no response
15	84343	taO86r6P	30.0	cozily	single	expensive	soccer	68693.0	no response
16	66428	qQSz8WWj	64.0	healthy	single	expensive	badminton	111755.0	no response
17	18272	waFEJp2V	37.0	cozily	married	practical	badminton	36430.0	no response
18	24254	CGEljPBt	61.0	cozily	married	practical	soccer	129420.0	response
19	29470	xQPaw1nF	48.0	cozily	single	expensive	badminton	78714.0	no response
20	78136	TW0vcILS	15.0	cozily	married	expensive	soccer	87219.0	no response
21	65878	VzsFKacF	51.0	cozily	married	expensive	badminton	123879.0	no response
22	15047	39OUyKYK	41.0	cozily	married	practical	athletics	137658.0	no response
23	87888	9uXUHkgm	30.0	cozily	single	practical	athletics	89243.0	no response
24	18734	GXQC3KVx	51.0	active	single	expensive	soccer	22977.0	no response

ภาพที่ 5.15 การแบ่งข้อมูล (Split Data)

ตัวอย่าง การสร้างต้นไม้ตัดสินใจ (Decision Tree) ด้วยโปรแกรม Rapid Miner Studio

โหนดที่เป็นตัว ๆ ไป X , โหนดที่เป็นลิฟโหนดไม่มีอะไรมาต่อ Y โหนดบนสุด Root มีเส้นที่เชื่อมเรียก กิ่ง หรือ Branch , Internal Node ตัวอย่างการตัดสินใจไปเรื่อย ๆ

ข้อดี คือสามารถแทนความหมายได้ง่าย

Decision Tree เป็นเทคนิคที่จำแนกการหาจาก Training Data ที่เรามี Attribute ทั่วไปที่มีเช่น มี Free, won ,Cash จาก Label ที่เรามี มันจะต้องมีการแบ่งแยกคำตอบได้อย่างชัดเจนที่สุด ด้วยการคำนวณค่า Entropy (เ็นโทรปี) และ Information Gain (IG)

สูตร Entropy (c1) = $-p(c1) \log p(c1)$

IG (parent, child) = Entropy (parent) - $[p(c1) \times \text{Entropy}(c1) + p(c2) \times \text{Entropy}(c2) + \dots]$

ลักษณะของค่า Entropy ถ้าค่าเหมือนกันหมดจะมีค่าเป็น ศูนย์ มาจากสูตร ความน่าจะเป็น 1 แอททริบิวต์ไหนที่จะมาเป็น Root Node คือเอาแอททริบิวต์ที่สามารถแบ่งแยกคำตอบได้อย่างชัดเจนก่อน

ตัวอย่าง การสร้างต้นไม้ตัดสินใจ (Decision Tree) ด้วยโปรแกรม Rapid Miner Studio

ตัวอย่างข้อมูล Weather เก็บสภาพภูมิอากาศจำนวน 14 วันเพื่อพิจารณาว่ามีการแข่งขันกีฬาได้หรือไม่ (ถามว่ามี Yes ที่ตัว ตอบ = 9 แทนด้วยรูปสีกฟ้า No = 5 แทนด้วยสีส้ม) ผลของวงกลมยังไม่ได้แบ่งข้อมูลออกมาออกมาพอกตามวน เข้าสู่ Entropy = 0.97 มีค่าใกล้ 1 แล้วทำการกรองข้อมูลจาก 14 ตัวที่มีผลเป็น Sunny

ไหนไหนแบ่งแยกคำตอบได้อย่างชัดเจนเป็น root node ก่อน ในตัวอย่างเลือก outlook เพราะมีคำตอบ outlook = overcast ที่เป็นสีฟ้าล้วนให้คำตอบในทางเดียวได้ชัดเจน

โมเดลเปลี่ยนตามดาต้าที่มี แต่ถ้ามีข้อมูลใหม่มันต้องเรียนรู้ได้มากขึ้น

ถ้าข้อมูลเป็นตัวเลข

- เรียงลำดับข้อมูลที่เป็นตัวเลขจากน้อยไปมาก
- แบ่งข้อมูลออกเป็น 2 ส่วนโดยหาจุดกึ่งกลางระหว่างค่าตัวเลข 2 ค่า
- คำนวณค่า information Gain จากข้อมูล 2 ส่วนที่แบ่งได้
- เลือกจุดกึ่งกลางที่ให้ค่า Information Gain สูงที่สุดมาใช้งานต่อ

แบบฝึกหัด

1. จงออกแบบ Decision Tree ของร้านขายขนมหวาน โดยเก็บข้อมูลลูกค้า พร้อมสินค้าที่ซื้อ (Data set)
2. จงออกแบบ Decision Tree การเข้าเช็คอิน - เช็คเอาท์ตามสถานที่ต่าง ๆ (Data set)

ตัวอย่าง

id	name	age	gender	order
1	Somchai	26	male	cake
2	Thanawat	18	male	Green tea
3	Sirirak	21	female	cake
1	Somchai	26	male	coffe
1	Somchai	26	male	coffe

อ้างอิง

- Data mining–การหาเหมืองข้อมูล โดย ดร. โทเมศ อัมพวัน
- www.itsci.mju.ac.th
- http://dataminingtrend.com/2014/wp-content/uploads/2014/02/RM7_chapter1.pdf
- <https://th.wikipedia.org/wiki/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%80%E0%B8%A3%E0%B8%B5%E0%B8%A2%E0%B8%99%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%A1%E0%B8%B5%E0%B8%9C%E0%B8%B9%E0%B9%89%E0%B8%AA%E0%B8%AD%E0%B8%99>
- <https://datawow.co.th/blogs/what-is-machine-learning>
- <https://www.depa.or.th/th/article-view/article11-2563>

วิดีโอ

https://www.youtube.com/watch?v=9xWrourDrc_o