การทำเหมืองข้อมูล
ตอน การหาความสัมพันธ์ของข้อมูลด้วย Apriori algorithm

วิวัฒน์ ชินนาทศิริกุล[1]

"รับซาลาเปาหรือขนมปังเพิ่มไหมครับ ?" เคยนึกสงสัยบ้างไหมครับ เวลาซื้อของในร้านสะดวกซื้อ พนักงานคิดเงินมักจะแนะนำสินค้าให้ลูกค้าซื้อสินค้าเพิ่มเสมอ ผู้เขียนก็นึกสงสัย อยู่ในใจและคิดว่าทางร้านสะดวกซื้อ คงกำหนดบทหรือมีสคริปต์ให้พนักงานคิดเงินพูด เผื่อลูกค้าอยากจะซื้อสินค้าอะไรเพิ่มอีก ทำให้ทางร้านขายสินค้าได้มากขึ้น เช่น ลูกค้าซื้อกาแฟร้อน พนักงานคิดเงินก็มักจะพูดว่ารับซาลาเปาหรือขนมปังเพิ่มไหมครับ หลังจากที่ผู้เขียนได้มีโอกาสไปอบรมเรื่อง การทำเหมืองข้อมูล (data mining) จึงเข้าใจว่า ทางบริษัทของร้านค้าปลีก มีการทำเหมืองข้อมูลกับฐานข้อมูลการขาย (sale database) เพื่อศึกษาพฤติกรรมการซื้อสินค้าของลูกค้า(customer behavior) ว่าลูกค้าที่ซื้อสินค้ามีพฤติกรรมการซื้ออย่างไรบ้าง กล่าวคือ ถ้าลูกค้าซื้อสินค้าชนิดหนึ่ง แล้วลูกค้าจะซื้อสินค้าชนิดอื่น ด้วยหรือไม่อย่างไร สินค้าที่ลูกค้าซื้อมีความสัมพันธ์กันอย่างไร

เป็นที่ทราบกันดีว่า ในปัจจุบันนี้เป็นยุคของเทคโนโลยีสารสนเทศอย่างแท้จริง แทบทุกหน่วยงานหรือองค์กรในปัจจุบันจะมีการจัดเก็บข้อมูลต่างๆ ขององค์กรไว้ในฐานข้อมูล หน่วยงานหรือองค์กรใดที่สามารถสกัดหรือวิเคราะห์ข้อมูลในฐานข้อมูลเพื่อให้มาซึ่งสารสนเทศที่เป็นประโยชน์ต่อการตัดสินใจขององค์กร ย่อมได้เปรียบกว่าหน่วยงานหรือองค์กรที่มีการจัดเก็บข้อมูลไว้มากแต่ไม่ได้นำมาใช้ประโยชน์ ดังภาพที่ 1

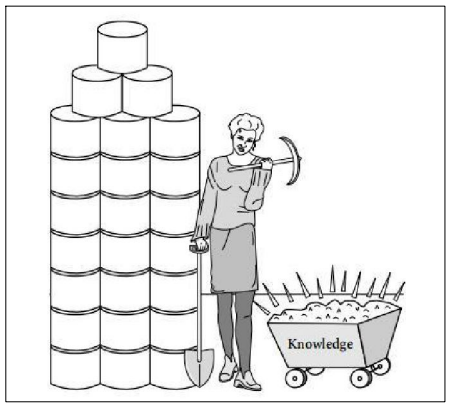
ภาพที่ 1 We are data rich , but information poor.
ที่มา Jiawei Han and Micheline KamberHan, 2006: 4

---

[1] สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏวไลยอลงกรณ์ ในพระบรมราชูปถัมภ์

การทำเหมืองข้อมูล

การทำเหมืองข้อมูล เป็นกระบวนการในการสกัด (extract) เพื่อค้นหารูปแบบ (patterns) หรือความรู้ (knowledge) จากฐานข้อมูลขนาดใหญ่ เพื่อให้ได้สารสนเทศที่น่าสนใจ ไม่คาดคิดมาก่อน และเป็นประโยชน์ มาใช้ในสนับสนุนการตัดสินใจในองค์กร

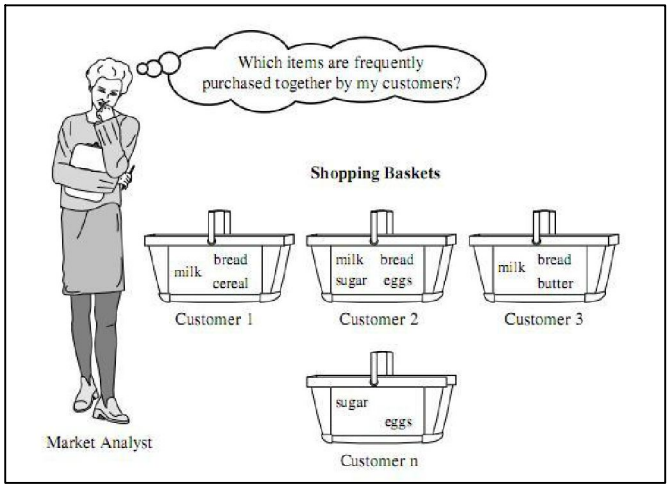


ภาพที่ 2 Data mining–searching for knowledge(interesting patterns) in your data
ที่มา Jiawei Han and Micheline KamberHan, 2006: 5

ตัวอย่างการนำเหมืองข้อมูลมาประยุกต์ใช้งาน เช่น ในทางการแพทย์ ใช้สารสนเทศที่ได้จากการทำเหมืองข้อมูลจากฐานข้อมูลประวัติคนไข้ เพื่อนำมาวินิจฉัยผู้ป่วยหรือทำนายโรคของผู้ป่วย ในทางการธนาคาร ใช้สารสนเทศที่ได้จากการทำเหมืองข้อมูลจากฐานข้อมูลสินเชื่อมาวิเคราะห์ในการปล่อยสินเชื่อให้ลูกค้ารายใหม่ ว่าทางธนาคารควรปล่อยสินเชื่อหรือไม่ ในทางอาชญวิทยาใช้สารสนเทศที่ได้จากการทำเหมืองข้อมูลจากฐานข้อมูลลายนิ้วมือหรือใบหน้ามาวิเคราะห์เพื่อหาเจ้าของลายนิ้วมือหรือเจ้าของใบหน้า

การหาความสัมพันธ์ (Association)

การทำเหมืองข้อมูล เพื่อหาความสัมพันธ์ของข้อมูล มักใช้ในธุรกิจการค้าปลีก (retailing business) เช่น ร้านค้าสะดวกซื้อ หรือซุปเปอร์มาเก็ต เป็นการวิเคราะห์ตะกร้าตลาด (market basket analysis) เพื่อศึกษาพฤติกรรมการซื้อสินค้าของลูกค้า และหาความสัมพันธ์ของสินค้าที่ลูกค้าซื้อ ดังภาพที่ 3 เพื่อนำผลลัพธ์ที่ได้จากความสัมพันธ์ มาใช้ในการจัดวางสินค้าบนชั้น เพื่อให้ลูกค้าสามารถหยิบซื้อสินค้าที่ซื้อด้วยกันได้สะดวก หรือนำผลลัพธ์ที่ได้มาใช้ในการส่งเสริมการขายสินค้า หรือจัดทำแคตาล็อกสินค้า



ภาพที่ 3 Market basket analysis
ที่มา Jiawei Han and Micheline KamberHan, 2006: 228

ความสัมพันธ์ของสินค้าที่ลูกค้าซื้อ จะแสดงในรูปของกฎความสัมพันธ์ (Association rule) ดังนี้

A -> B [support, confidence] โดยที่ A, B แทนรายการสินค้า

เช่น milk -> eggs [support=25%, confident=33.34%] หมายความว่า 25% ของทรานแซคชั่นทั้งหมด ลูกค้าจะซื้อนม (milk) และซื้อไข่ (eggs) พร้อมกัน และ 33.34% ของลูกค้าที่ซื้อนมแล้วจะซื้อไข่ด้วย

กฎความสัมพันธ์ที่สนใจหรือกฎความสัมพันธ์ที่แข็งแกร่ง (strong association rules) คือ กฎความสัมพันธ์ที่มีค่าสนับสนุน (support) และค่าความเชื่อมั่น (confidence) ผ่านเกณฑ์ขั้นต่ำ (minimum threshold) ที่ผู้วิเคราะห์ข้อมูลกำหนดขึ้นมา

แนวคิดพื้นฐานเกี่ยวกับกฎความสัมพันธ์

ถ้ากำหนดให้ D เป็นเซตของทรานแซคชั่น และแต่ละทรานแซคชั่น ประกอบด้วยเซตของรายการ (items)

กำหนดให้ I แทนเซตของรายการ โดยที่ $I = \{ i_1, i_2, ..., i_m \}$

กฎความสัมพันธ์ จะเขียนในรูป $A \rightarrow B$ โดยที่ $A \subset I, B \subset I$ และ $A \cap B = \varnothing$ และนิยามคำศัพท์ ดังนี้

itemsets เป็นเซตของรายการหรือรายการสินค้า

k-itemsets เป็น itemsets ที่ประกอบด้วย k รายการ

frequent itemsets เป็น itemsets ที่มีค่าสนับสนุน (supports) มากกว่าหรือเท่ากับค่าสนับสนุนที่กำหนด (minimum support threshold)

กฎความสัมพันธ์ที่สนใจคือ กฎความสัมพันธ์ ที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับ-สนุนที่กำหนด และค่าความเชื่อมั่น มีค่ามากกว่าหรือเท่ากับค่าความเชื่อมั่นที่กำหนด (minimum confidence threshold)

ค่าสนับสนุน หาจากสูตร

$$\text{support}(A \rightarrow B) = P(A \cup B) = \frac{\text{จำนวนทรานแซคชั่นที่ปรากฏรายการทั้ง A และ B}}{\text{จำนวนทรานแซคชั่นทั้งหมด}}$$

ค่าความเชื่อมั่น หาจากสูตร

$$\text{confidence}(A \rightarrow B) = P(B \mid A) = P(A \cup B) / P(A)$$

$$= \frac{\text{จำนวนทรานแซคชั่นที่ปรากฏรายการทั้ง A และ B}}{\text{จำนวนทรานแซคชั่นที่ปรากฏรายการ A}}$$

ขั้นตอนการหากฎความสัมพันธ์

มี 2 ขั้นตอน ได้แก่

1. หา frequent itemsets ทั้งหมด
2. หา กฎความสัมพันธ์ จาก frequent itemsets ที่ได้จากข้อที่ 1 ที่ผ่านเกณฑ์ของค่าสนับสนุนและค่าความเชื่อมั่นที่กำหนด

การหา frequent itemsets ด้วย Apriori algorithm

Apriori เป็นอัลกอริทึมพื้นฐาน ที่นิยมนำมาใช้ในการหา frequent itemsets อัลกอริทึมนี้นำเสนอโดย R. Agrawal and R. Srikant ใน ปี ค.ศ. 1994 ชื่อของอัลกอริทึมอิงมาจากพื้นฐานความจริงที่ว่า อัลกอริทึมจะใช้ความรู้ก่อนหน้านี้ (prior knowledge) ของ frequent itemsets มาใช้หรือนำ frequent itemsets ที่ได้ก่อนหน้านี้ ใช้หา frequent itemsets ในระดับถัดไป

Apriori algorithm จะใช้ k-itemsets เพื่อหา (k+1)-itemsets เริ่มต้นการทำงานจะหา 1-itemsets โดยการอ่านค่า (scan) จากฐานข้อมูลเพื่อนับจำนวนหรือความถี่ของแต่ละ items (ข้อมูลหรือรายการสินค้า) เพื่อหา items ที่ผ่านเกณฑ์ของ minimum support ผลลัพธ์ที่ได้ สมมติให้เป็นเซตของ L_1 จากนั้นนำเซตของ L_1 (เซตของ 1-items ที่ผ่านเกณฑ์ minimum support) ไปหาเซตของ L_2 (เซตของ 2-items ที่ผ่านเกณฑ์ minimum support) และนำเซตของ L_2 ไปหาเซตของ L_3 (เซตของ 3-items ที่ผ่านเกณฑ์ minimum support) และทำเช่นนี้เรื่อยไป จนกระทั่งไม่สามารถหา frequent k-itemsets ได้

ในการหาแต่ละเซตของ L_k จำเป็นต้องอ่านค่าจากฐานข้อมูลซึ่งทำให้เสียเวลาในการทำงาน เพื่อเป็นการปรับปรุงประสิทธิภาพในการทำงาน ในการหา frequent itemsets จะใช้ Apriori property ช่วยในการทำงานเพื่อลดจำนวนครั้งในการค้นหาข้อมูล

Apriori property กล่าวว่า ทุกสับเซตที่ไม่ใช่เซตว่างของ frequent itemsets จะต้องเป็น frequent เช่น ถ้า {A, B} เป็น frequent itemsets แล้ว ทั้ง {A} และ {B} ต้องเป็น frequent itemsets แต่ถ้า {A} หรือ {B} ไม่เป็น frequent itemsets แล้ว {A, B} จะไม่เป็น frequent itemsets

ตัวอย่าง การนำ Apriori property ไปใช้ในอัลกอริทึม คือการใช้ L_{k-1} (frequent (k-1)-itemsets) เพื่อหา L_k (frequent k-itemsets) เมื่อ k>=2 ซึ่งมีกระบวนการทำงาน 2 ขั้นตอน ได้แก่ การ join และการ prune

1. ขั้นตอนการ join ในการหา L_k เซตของ candidate k-itemsets หรือ C_k จะถูกสร้างขึ้นโดยนำ L_{k-1} มา join กัน สมาชิกในเซตของ L_{k-1} จะนำมา join กันได้ ถ้าสมาชิก k-2 ตัวแรก มีค่าเหมือนกัน หมายความว่า ถ้า k=3 เซตของ C_3 จะสร้างโดยนำ L_{3-1} หรือ L_2 มา join กัน สมาชิกใน L_2 ที่นำมา join กันได้ ต้องมีสมาชิก 3-2 ตัวแรก หรือ 1 รายการแรกเหมือนกัน หรือ ถ้า k=4 เซตของ C_4 จะสร้างโดย L_3 มา join กัน สมาชิกใน L_3 ที่นำมา join กันได้ ต้องมีสมาชิก 2 รายการแรกเหมือนกัน

2. ขั้นตอนการ prune การอ่านค่าจากฐานข้อมูลเพื่อหาความถี่ของสมาชิกแต่ละตัวใน C_k เพื่อหาว่าเป็น frequent หรือไม่นั้น ถ้า C_k มีขนาดใหญ่จะทำให้เสียเวลาในการอ่านข้อมูลจากฐานข้อมูลเพื่อนับความถี่ของสมาชิกแต่ละตัวของ C_k เพื่อเป็นการลดขนาดของ C_k ก่อนที่จะอ่านค่าในฐานข้อมูล จะนำ Apriori property มาใช้ โดยพิจารณาว่า ถ้า (k-1)-subset ใดของ C_k ที่ไม่ได้อยู่ใน L_{k-1} แล้ว k-itemsets ที่มี (k-1)-subset จะไม่เป็น frequent ด้วย สามารถตัดสมาชิกดังกล่าวออกจาก C_k ได้ เช่น ถ้า {A, B} ไม่เป็น frequent แล้ว {A, B, C} จะไม่เป็น frequent ด้วยเพราะ {A, B} เป็นสับเซตของ {A, B, C} และ {A, B} ไม่เป็น frequent การกระทำดังกล่าวช่วยลดขนาดของ C_k และทำให้การทำงานรวดเร็วขึ้น

ตัวอย่าง จากข้อมูลที่กำหนดดังตารางที่ 1 จะหากฎความสัมพันธ์ เมื่อกำหนดค่า minimum support = 33.34% และค่า minimum confidence =80%

ตารางที่ 1 แสดงตัวอย่างรายการสินค้าที่ลูกค้าซื้อ

ลำดับการซื้อ	รายการสินค้าที่ซื้อ
1	กาแฟร้อน, ซาลาเปา, ขนมปัง
2	กาแฟร้อน, ซาลาเปา
3	กาแฟร้อน, น้ำดื่ม, ผ้าเย็น
4	ผ้าเย็น, น้ำดื่ม
5	ผ้าเย็น, ขนมปัง
6	กาแฟร้อน, น้ำดื่ม, ขนมปัง, ซาลาเปา

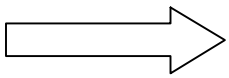
ขั้นตอนการทำงาน

1. หา frequent 1-itemsets (L_1) จากข้อมูลที่กำหนด โดยพิจารณาจาก candidate 1-itemsets (C_1) โดย อ่านค่าจากข้อมูลเพื่อนับความถี่ของ 1-temsets ใน C_1 จากนั้นนำค่า support count ในแต่ละ 1-itemsets ของ C_1 ไปเทียบกับค่า minimum support ที่กำหนด 1-itemsets ที่ผ่านเกณฑ์ เรียกว่า frequent 1-itemsets (L_1) ดังตารางที่ 2

ตารางที่ 2 แสดง Candidate 1-itemsets (C_1) และ Frequent 1-itemsets (L_1)

C_1

itemsets	Support count
{กาแฟร้อน}	4
{ซาลาเปา}	3
{ขนมปัง}	3
{น้ำดื่ม}	3
{ผ้าเย็น}	3



เปรียบเทียบค่า support count กับ minimum support count

L_1

itemsets	Support count
{กาแฟร้อน}	4
{ซาลาเปา}	3
{ขนมปัง}	3
{น้ำดื่ม}	3
{ผ้าเย็น}	3

ข้อมูลในตาราง C_1 แสดง 1-itemsets และความถี่ที่ปรากฏในทรานแซคชั่น เช่น กาแฟร้อน ปรากฏอยู่ 4 ทรานแซคชั่นคือ ทรานแซคชั่นที่ 1, 2, 3 และ 6 ซาลาเปา ปรากฏอยู่ 3 ทรานแซคชั่นคือ ทรานแซคชั่นที่ 1, 2 และ 6 จากนั้นนำความถี่ของแต่ละ 1-itemsets ในตาราง C_1 ไปเปรียบเทียบกับค่า minimum support ที่กำหนดว่าผ่านเกณฑ์หรือไม่ เพื่อสร้าง L_1 จากตัวอย่างกำหนดค่า minimum support = 33.34% คิดเป็น $\frac{33.34}{100}\times 6 = 2$ ทรานแซคชั่น ดังนั้น 1- itemsets ทุกรายการ ในตาราง C_1 เป็น frequent 1-itemsets เนื่องจากมีค่าสนับสนุนมากกว่าหรือเท่ากับ 2

2. หา frequent 2-itemsets (L_2) โดยพิจารณาจาก candidate 2-itemsets (C_2) ซึ่ง C_2 จะสร้างจาก itemsets ที่ได้ใน L_1 ในตารางที่ 2 มา join กัน ในขั้นตอนนี้ยังไม่มีการ prune เพราะทุกๆ itemsets ใน L_1 ที่นำมา join เป็น frequent ทุกตัว เมื่อได้ candidate 2-itemsets หรือ C_2 แล้วจะอ่านข้อมูลเพื่อนับความถี่ของ 2-temsets ใน C_2 ผลลัพธ์ที่ได้แสดงดังตาราง C_2 จากนั้นนำค่า support count ในแต่ละ 2-itemsets ของ C_2 ไปเทียบกับค่า minimum support ที่กำหนด 2-itemsets ที่ผ่านเกณฑ์ เรียกว่า frequent 2-itemsets (L_2) ดังตารางที่ 3

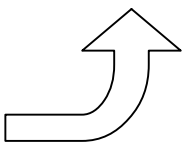
ตารางที่ 3 แสดง Candidate 2-itemsets (C_2) และ Frequent 2-itemsets (L_2)

C_2

itemsets	Support count
{กาแฟ, ซาลาเปา}	3
{กาแฟ, ขนมปัง}	2
{กาแฟ, น้ำดื่ม}	2
{กาแฟ, ผ้าเย็น}	1
{ซาลาเปา, ขนมปัง}	2
{ซาลาเปา, น้ำดื่ม}	1
{ซาลาเปา, ผ้าเย็น}	0
{ขนมปัง, น้ำดื่ม}	1
{ขนมปัง, ผ้าเย็น}	1
{น้ำดื่ม, ผ้าเย็น}	2

L_2

itemsets	Support count
{กาแฟ, ซาลาเปา}	3
{กาแฟ, ขนมปัง}	2
{กาแฟ, น้ำดื่ม}	2
{ซาลาเปา, ขนมปัง}	2
{น้ำดื่ม, ผ้าเย็น}	2



เปรียบเทียบค่า support count กับ minimum support count

3. หา frequent 3-itemsets (L_3) โดยพิจารณาจาก candidate 3-itemsets (C_3) ซึ่ง C_3 จะสร้างจาก itemsets ที่ได้ใน L_2 ในตารางที่ 3 มา join กัน 2-itemsets ที่จะนำมา Join กัน จะพิจารณาจาก k-1 รายการแรกที่มีค่าหมือนกัน ในที่นี้นำ frequent 2-itemsets มา Join ดังนั้น k=2, k-1=1 นั่นคือ จะนำ frequent 2-itemsets ที่มี 1 รายการแรกเหมือนกันมา join กัน ได้แก่

{กาแฟ, ซาลาเปา} Join กับ {กาแฟ, ขนมปัง} ได้ {กาแฟ, ซาลาเปา, ขนมปัง}

{กาแฟ, ซาลาเปา} Join กับ {กาแฟ, น้ำดื่ม} ได้ {กาแฟ, ซาลาเปา, น้ำดื่ม}

{กาแฟ, ขนมปัง} Join กับ {กาแฟ, น้ำดื่ม} ได้ {กาแฟ, ขนมปัง, น้ำดื่ม}

จากนั้นทำการ prune โดยพิจารณา k-1 subset ของผลลัพธ์ที่ได้ว่าอยู่ใน frequent 2-itemsets หรือไม่ ถ้าเซตใดมีสับเซตที่ไม่อยู่ ใน frequent 2-itemsets เซตนั้นจะถูกตัดทิ้ง

สับเซตที่มีสมาชิก 2 ตัวของ {กาแฟ, ซาลาเปา, ขนมปัง} คือ {กาแฟ, ซาลาเปา} {กาแฟ, ขนมปัง} และ {ซาลาเปา, ขนมปัง} ซึ่งทุกสับเซตอยู่ใน frequent 2-itemsets ในตาราง L_2 ดังนั้น {กาแฟ, ซาลาเปา, ขนมปัง} เป็น candidate 3-itemsets

ส่วน {กาแฟ, ซาลาเปา, น้ำดื่ม} ไม่เป็น candidate 3-itemsets เพราะ {ซาลาเปา, น้ำดื่ม} ไม่ใช่ frequent 2-itemsets

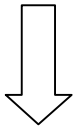
และ {กาแฟ, ขนมปัง, น้ำดื่ม} ไม่เป็น candidate 3-itemsets เพราะ{ขนมปัง, น้ำดื่ม} ไม่ใช่ frequent 2-itemsets

จากนั้นทำการอ่านค่าจากฐานข้อมูล เพื่อนับจำนวนรายการของ candidate 3-itemsets ผลลัพธ์แสดงดังตารางที่ 4

ตารางที่ 4 แสดง Candidate 3-itemsets (C_3) และ Frequent 3-itemsets (L_3)

C_3

itemsets	Support count
{กาแฟ, ซาลาเปา, ขนมปัง}	2



เปรียบเทียบค่า support count กับ minimum support count

L_3

itemsets	Support count
{กาแฟ, ซาลาเปา , ขนมปัง}	2

จากขั้นตอนดังกล่าว จะได้ frequent itemsets ทั้งหมด คือ {กาแฟ} {ซาลาเปา} {ขนมปัง} {น้ำดื่ม} {ผ้าเย็น} {กาแฟ, ซาลาเปา} {กาแฟ, ขนมปัง} {กาแฟ, น้ำดื่ม} {ซาลาเปา, ขนม-ปัง} {น้ำดื่ม, ผ้าเย็น} และ {กาแฟ, ซาลาเปา, ขนมปัง}

ผู้วิเคราะห์ข้อมูล ต้องพิจารณาว่าต้องการสร้างความสัมพันธ์ของสินค้ากี่รายการ ถ้าพิจารณาความสัมพันธ์ของสินค้า 2 รายการ เช่น {กาแฟ, ซาลาเปา} สามารถสร้างกฎความสัมพันธ์จาก {กาแฟ, ซาลาเปา} ดังนี้

1. กาแฟ -> ซาลาเปา [50%, 75%]
2. ซาลาเปา -> กาแฟ [50%, 100%]

ค่าเปอร์เซนต์ที่อยู่หลังกฎความสัมพันธ์คือ ค่าสนับสนุน และค่าความเชื่อมั่น ซึ่งคำนวณได้จากค่า support $(A \rightarrow B) = P(A \cup B)$ และ confidence $(A \rightarrow B) = P(B \mid A) = P(A \cup B) / P(A)$ เนื่องจากกำหนด minimum confidence = 80% ดังนั้นกฎที่รับได้คือ ซาลาเปา -> กาแฟ (50%,100%) ซึ่งหมายความว่า มีลูกค้าซื้อซาลาเปาและกาแฟพร้อมกันอยู่ 50% ของทรานแซคชั่นทั้งหมด และถ้าลูกค้าซื้อซาลาเปาแล้วจะซื้อกาแฟด้วยมีอยู่ 100%

กรณีพิจารณาความสัมพันธ์ของสินค้า 3 รายการ เช่น {กาแฟ, ซาลาเปา, ขนมปัง} สามารถสร้างกฎความสัมพันธ์ได้ดังนี้

1. กาแฟ -> ซาลาปา ^ ขนมปัง [33.34%, 50%]
2. ซาลาเปา -> กาแฟ ^ ขนมปัง [33.34%, 66.68%]
3. ขนมปัง -> กาแฟ ^ ซาลาเปา [33.34%, 66.68%]
4. กาแฟ ^ ซาลาเปา -> ขนมปัง [33.34%, 66.68%]
5. กาแฟ ^ ขนมปัง -> ซาลาเปา [33.34%, 100%]
6. ซาลาเปา ^ ขนมปัง -> กาแฟ [33.34%, 100%]

กฎที่รับได้คือ กฎข้อ 5 และ 6 คือ กาแฟ ^ ขนมปัง -> ซาลาเปา [33.34%,100%] ซึ่งหมายความว่า มีลูกค้าที่ซื้อกาแฟ ขนมปัง และซาลาเปา พร้อมกันอยู่ 33.34% ของทรานแซคชั่นทั้งหมด และถ้าลูกค้าซื้อกาแฟและขนมปังแล้วจะซื้อซาลาเปาด้วยมีอยู่ 100% และ ซาลาเปา ^ ขนมปัง -> กาแฟ [33.34%, 100%] ซึ่งหมายความว่า มีลูกค้าที่ซื้อ ซาลาเปา ขนมปัง และกาแฟ พร้อมกันอยู่ 33.34% ของทรานแซคชั่นทั้งหมด และถ้าลูกค้าซื้อซาลาเปาและขนมปังแล้วจะซื้อกาแฟด้วยมีอยู่ 100%

จากบทความดังกล่าว ท่านผู้อ่านคงพอจะเข้าใจบ้างแล้วนะครับว่าทำไมพนักงานคิดเงินในร้านสะดวกซื้อต้องพูดว่า **“รับซาลาเปาหรือขนมปังเพิ่มไหมครับ ?”**

เอกสารอ้างอิง

Daniel T. Larose. (2005). **Discovering knowledge in data : an introduction to data mining**. Canada: John Wiley & Sons.

Han J, Kamber M. (2006.). **Data Mining: Concepts and Techniques**. (2nd ed). San Francisco: Morgan Kaufmann Publishers.