

วิชาเหมืองข้อมูล Data Mining (4124305)

บทที่ 5

กฎความสัมพันธ์ (Association Rules)

2/2568

File Name : Lec05_DataMining_Association_rule_with_Python_17_12_2568



Asst. Prof. Dr. Nattapong Songneam

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

แก้ไขล่าสุด : 17.12.2568

Reviews

เทคนิคในการทำเหมืองข้อมูล

1. การวิเคราะห์ข้อมูลเชิงสถิติในงานเหมืองข้อมูล

บทที่ 4

การวิเคราะห์ข้อมูลเชิงสถิติในงานเหมืองข้อมูล

2. กฎความสัมพันธ์ (Association rule)

บทที่ 5

กฎความสัมพันธ์ (Association rule)

3. การจำแนกข้อมูล (Data classification)

บทที่ 6-7

Decision Tree, SVM , KNN , NN , Naive Bays

4. การแบ่งกลุ่มข้อมูล (Data clustering)

บทที่ 8

K-Mean, Hierarchical Clustering

5. การสร้างมโนภาพ (Visualization)

บทที่ 9

กราฟ, แผนภูมิ, แผนภาพ, หรือภาพสามมิติ

บทที่ 5

กฎความสัมพันธ์ (Association Rules)

ในบทที่ 5 กฎความสัมพันธ์ (Association Rules) เนื้อหาสามารถถูกออกแบบได้ตามความต้องการของการนำเสนอหัวข้อหลัก ๆ และรายละเอียดที่เกี่ยวข้อง ต่อไปนี้คือตัวอย่างโครงร่างเนื้อหาสำหรับบทที่ 5 เกี่ยวกับกฎความสัมพันธ์ (Association Rules)

กฎความสัมพันธ์ (Association Rules) เป็นเทคนิคทางสถิติและข้อมูลขนาดใหญ่ที่ใช้ในการค้นหาความสัมพันธ์ระหว่างตัวแปรในชุดข้อมูล โดยมักใช้ในการวิเคราะห์ข้อมูลการซื้อขายสินค้าหรือพฤติกรรมการใช้บริการลูกค้า เป้าหมายหลักของการใช้กฎความสัมพันธ์คือการค้นหากฎที่บอถึงความสัมพันธ์ระหว่างรายการสินค้าหรือตัวแปรอื่น ๆ ในชุดข้อมูล ซึ่งสามารถช่วยในการทำนายหรือตีความพฤติกรรมของลูกค้าได้

เนื้อหา (Agenda)

บทวน

บทวน

CripDM 6 ชั้น
การทำความสะดวกข้อมูล
การแปลงข้อมูล

5.1 บทนำ

- 5.1.1 ความหมายและความสำคัญของกฎความสัมพันธ์
- 5.1.2 วัตถุประสงค์ของการใช้กฎความสัมพันธ์ในการเหมืองข้อมูล
- 5.1.3 ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล

เนื้อหา (Agenda)

5.2 พื้นฐานของกฎความสัมพันธ์

- 4.2.1 กลุ่มข้อมูล (Itemset)
- 4.2.2 กฎความสัมพันธ์ (Association Rule)
- 4.2.3 ค่าสนับสนุน (Support)
- 4.2.4 ค่าเชื่อมั่น (Confidence)
- 4.2.5 ค่าเชื่อมั่นที่ต่ำสุด (Minimum Confidence)

5.3. การทำงานของกฎความสัมพันธ์

- 5.4.1 การค้นหากฎความสัมพันธ์
- 5.4.2 วิธีการเลือกกฎความสัมพันธ์
- 5.3.3 การจัดลำดับกฎความสัมพันธ์ตามความสนใจ
- 5.4.4 การประเมินและการปรับขั้นตอน
- 5.4.5 การให้ความสำคัญกับค่าสนับสนุนและค่าเชื่อมั่น
- 5.4.6 การปรับค่าเชื่อมั่นที่ต่ำสุด

เนื้อหา (Agenda)

5.4 การนำไปใช้ในธุรกิจและการวิเคราะห์ข้อมูล

5.4.1 การใช้กฎความสัมพันธ์ในธุรกิจ

5.4.2 ตัวอย่างการใช้กฎความสัมพันธ์ในการปรับแผนการตลาด

5.4.3 การนำกฎความสัมพันธ์มาปรับแต่งกลยุทธ์ธุรกิจ

5.4.4 การนำกฎความสัมพันธ์ไปใช้ในการวิเคราะห์ข้อมูล

5.4.5 การค้นพบความสัมพันธ์ที่ไม่ได้รับความสนใจ

5.4.6 การใช้กฎความสัมพันธ์ในการทำนายแนวโน้ม

5.5 ที่มาและขนาดของกฎความสัมพันธ์

5.5.1 ที่มาของกฎความสัมพันธ์

5.5.2 ทิศทางและแนวโน้มในอนาคตของการใช้กฎความสัมพันธ์

เนื้อหา (Agenda)

5.6 สรุป

5.6.1 สรุปข้อมูลหลัก

5.6.2 ความสำคัญของการใช้กฎความสัมพันธ์

5.6.3 การนำไปใช้ในการวิเคราะห์ข้อมูลและธุรกิจ

5.1

บทนำ

5.1.1 ความสำคัญของกฎความสัมพันธ์

5.1.2 วัตถุประสงค์ของการใช้กฎความสัมพันธ์ในการเหมืองข้อมูล

5.1.3 ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล

5.1

5.1.1 ความหมายของกฎความสัมพันธ์ (Association Rules)

บทที่ 5

ปัจจุบันก้าวเข้าสู่ยุคที่มีข้อมูลขนาดมหาศาลซึ่งเกิดจากการใช้งานในชีวิตประจำวันของเราเอง เช่น การซื้อสินค้าในซูเปอร์มาร์เกต หรือ การใช้งานเครือข่ายสังคม (social network) แบบต่าง ๆ อาทิเช่น เฟซบุ๊ก (Facebook) หรือ ทวิตเตอร์ (Twitter) เมื่อข้อมูลมีจำนวนมากขึ้นย่อมทำให้เกิดความต้องการนำข้อมูลเหล่านี้มาใช้เพื่อก่อให้เกิดประโยชน์มากที่สุด วิธีการหนึ่งที่นิยมใช้กันมากในปัจจุบันคือการวิเคราะห์หาความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล วิธีการนี้คือ “การขุดเหมืองข้อมูล” (data mining) หรือเรียกทับศัพท์ว่า ดาต้า ไมนิง เป็นการเรียกชื่อทับศัพท์เพื่อให้เป็นสากลและเข้าใจได้ง่ายกว่า การวิเคราะห์ข้อมูลด้วยวิธีดาต้า ไมนิงนี้มีตัวอย่างความสำเร็จให้เห็นอยู่มากมาย ในที่นี้จะขอยกมาเพียง 2 ตัวอย่าง

- ห้างวอลมาร์ต (Walmart)
- ห้างตาร์เก็ต (Target)

Association Rules เป็นหนึ่งในเทคนิคสำคัญในกระบวนการวิเคราะห์ข้อมูล (Data Mining) โดยเฉพาะอย่างยิ่งในการวิเคราะห์พฤติกรรมของลูกค้าในเชิงธุรกิจ ความสำคัญของกฎความสัมพันธ์สามารถอธิบายได้ดังนี้

5.1

5.1.1 ความหมายและความสำคัญของกฎความสัมพันธ์ (Association Rules)

บทที่ 5

กฎความสัมพันธ์ (Association rule) เป็นเทคนิคในการค้นหาค่าความสัมพันธ์ระหว่างตัวแปรต่าง ๆ ในชุดข้อมูล โดยกฎความสัมพันธ์จะแสดงความสัมพันธ์ระหว่างตัวแปรสองตัวขึ้นไป โดยระบุว่าตัวแปรเหล่านั้นเกิดขึ้นร่วมกันบ่อยครั้งแค่ไหน

ห้างวอลล์มาร์ทสามารถประยุกต์ใช้กฎความสัมพันธ์เพื่อวัตถุประสงค์ต่าง ๆ ดังนี้

- **การปรับปรุงประสบการณ์ของลูกค้า:** วอลล์มาร์ทสามารถใช้กฎความสัมพันธ์เพื่อวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า เพื่อนำไปปรับปรุงประสบการณ์ของลูกค้า เช่น นำเสนอสินค้าที่เกี่ยวข้องหรือแนะนำสินค้าอื่น ๆ ที่ลูกค้าอาจสนใจ
- **การเพิ่มยอดขาย:** วอลล์มาร์ทสามารถใช้กฎความสัมพันธ์เพื่อวิเคราะห์แนวโน้มการซื้อสินค้าของลูกค้า เพื่อนำไปพัฒนากลยุทธ์การขาย เช่น การจัดโปรโมชันสินค้าร่วมกันหรือวางสินค้าให้อยู่ในตำแหน่งที่เหมาะสม
- **ลดต้นทุน:** วอลล์มาร์ทสามารถใช้กฎความสัมพันธ์เพื่อวิเคราะห์สินค้าคงคลัง เพื่อนำไปลดต้นทุน เช่น กำหนดปริมาณการสั่งซื้อสินค้าที่เหมาะสมหรือควบคุมสินค้าคงคลังให้อยู่ในระดับที่เหมาะสม

5.1

5.1.1 ความหมายและความสำคัญของกฎความสัมพันธ์ (Association Rules)

บทที่ 5

ตัวอย่างการประยุกต์ใช้กฎความสัมพันธ์ของห้างวอลมาร์ต เช่น

- วอลมาร์ตพบว่าลูกค้ามักจะซื้อสินค้านมและขนมปังร่วมกันบ่อยครั้ง วอลมาร์ตจึงสามารถนำเสนอสินค้านมและขนมปังร่วมกันในแผนกเดียวกัน เพื่อกระตุ้นให้ลูกค้าซื้อสินค้าทั้งสองอย่างพร้อมกัน
- วอลมาร์ตพบว่าลูกค้ามักจะซื้อสินค้ากระดาษชำระและน้ำยาล้างจานร่วมกันบ่อยครั้ง วอลมาร์ตจึงสามารถจัดโปรโมชั่นสินค้ากระดาษชำระและน้ำยาล้างจานร่วมกัน เพื่อกระตุ้นให้ลูกค้าซื้อสินค้าทั้งสองอย่างพร้อมกัน
- วอลมาร์ตพบว่าสินค้าบางรายการมีแนวโน้มที่จะขายไม่หมด วอลมาร์ตจึงสามารถลดราคาสินค้าเหล่านี้เพื่อลดต้นทุนสินค้าคงคลัง

กฎความสัมพันธ์เป็นเครื่องมือที่มีประสิทธิภาพที่สามารถช่วยให้ห้างวอลมาร์ตปรับปรุงการดำเนินงานและเพิ่มผลประกอบการ

5.1

5.1.1 ความหมายและความสำคัญของกฎความสัมพันธ์ (Association Rules)

บทที่ 5

ในที่นี้จะขอยกตัวอย่างมาให้ดูสัก 2 ตัวอย่างก่อน โดยตัวอย่างแรกเป็นตัวอย่างคลาสสิก นั่นก็คือการที่ห้างวอลมาร์ต (Walmart) ได้ทำการค้นพบพฤติกรรมการซื้อสินค้าของลูกค้าที่เป็นเพศชายว่าในช่วงเย็นของวันศุกร์ลูกค้ากลุ่มนี้มักจะมาซื้อสินค้าสองอย่างควบคู่กันไป นั่นก็คือ “เบียร์และผ้าอ้อม” โดยจากการวิเคราะห์เจาะลึกลงไปก็พบเหตุผลว่าการที่สินค้าสองอย่างนี้มีการซื้อร่วมกันบ่อย ๆ เพราะว่าพ่อบ้านส่วนใหญ่มักจะซื้อเบียร์ไปดื่มในช่วงสุดสัปดาห์และเกิดคิดถึงลูกน้อยของตัวเองขึ้นมาจึงซื้อผ้าอ้อมติดไม้ติดมือไปด้วย (ถ้าเป็นบ้านเราอาจจะพบว่าพฤติกรรมแบบนี้เกิดขึ้นเพราะพ่อบ้านถูกภรรยาใช้ให้มาซื้อผ้าอ้อมให้ลูกแล้วอยากดื่มเบียร์ด้วยหรือเปล่าก็ไม่รู้) หลังจากห้างวอลมาร์ตรู้ถึงพฤติกรรมแบบนี้ทางห้างก็สามารถที่จะจัดวางสินค้าสองชนิดนี้ให้สามารถค้นหาได้ง่าย ๆ หรือมองเห็นได้ง่ายเพื่อเพิ่มโอกาสที่ลูกค้าจะได้ซื้อติดไม้ติดมือกันไปด้วย

5.1

5.1.1 ความหมายและความสำคัญของกฎความสัมพันธ์ (Association Rules)

บทที่ 5

ห้างคาร์เก็ต (Target) เป็นหนึ่งในห้างค้าปลีกที่ใหญ่ที่สุดในสหรัฐอเมริกา ห้างคาร์เก็ตใช้กฎความสัมพันธ์เพื่อวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า เพื่อนำไปปรับปรุงประสบการณ์ของลูกค้าและเพิ่มยอดขาย ตัวอย่างการประยุกต์ใช้กฎความสัมพันธ์ของห้างคาร์เก็ต เช่น

- **แนะนำสินค้าที่เกี่ยวข้อง:** ห้างคาร์เก็ตสามารถใช้กฎความสัมพันธ์เพื่อแนะนำสินค้าที่เกี่ยวข้องให้กับลูกค้า เช่น พบว่าลูกค้ามักจะซื้อสินค้านมและขนมปังร่วมกันบ่อยครั้ง ห้างคาร์เก็ตจึงสามารถแนะนำสินค้านมให้กับลูกค้าที่กำลังซื้อสินค้าขนมปัง
- **จัดโปรโมชั่นสินค้าร่วมกัน:** ห้างคาร์เก็ตสามารถใช้กฎความสัมพันธ์เพื่อจัดโปรโมชั่นสินค้าร่วมกัน เช่น พบว่าลูกค้ามักจะซื้อสินค้ากระดาษชำระและน้ำยาล้างจานร่วมกันบ่อยครั้ง ห้างคาร์เก็ตจึงสามารถจัดโปรโมชั่นสินค้ากระดาษชำระและน้ำยาล้างจานร่วมกัน เพื่อกระตุ้นให้ลูกค้าซื้อสินค้าทั้งสองอย่างพร้อมกัน
- **ส่งอีเมลส่งเสริมการขาย:** ห้างคาร์เก็ตสามารถใช้กฎความสัมพันธ์เพื่อส่งอีเมลส่งเสริมการขายให้กับลูกค้า เช่น พบว่าลูกค้ามักจะซื้อสินค้าผ้าอ้อมเด็กในช่วงไตรมาสที่ 3 ของปี ห้างคาร์เก็ตจึงสามารถส่งอีเมลส่งเสริมการขายสินค้าผ้าอ้อมเด็กให้กับลูกค้าในช่วงไตรมาสที่ 3 ของปี เพื่อกระตุ้นให้ลูกค้าซื้อสินค้า

5.1

5.1.1 ความหมายและความสำคัญของกฎความสัมพันธ์ (Association Rules)

บทที่ 5

นอกจากนี้ ห้างค้าปลีกยังใช้กฎความสัมพันธ์เพื่อวิเคราะห์แนวโน้มการซื้อสินค้าของลูกค้า เพื่อนำไปพัฒนากลยุทธ์การขาย เช่น พบว่าลูกค้ามีแนวโน้มที่จะซื้อสินค้าอิเล็กทรอนิกส์มากขึ้น ห้างค้าปลีกจึงสามารถเพิ่มพื้นที่จัดจำหน่ายสินค้าอิเล็กทรอนิกส์

กฎความสัมพันธ์เป็นเครื่องมือที่มีประสิทธิภาพที่สามารถช่วยให้ห้างค้าปลีกปรับปรุงประสบการณ์ของลูกค้าและเพิ่มยอดขาย โดยห้างค้าปลีกใช้กฎความสัมพันธ์เพื่อวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า เพื่อนำไปพัฒนากลยุทธ์การขายที่มีประสิทธิภาพ

นอกจากห้างวอล์กมาร์เก็ตและห้างค้าปลีกแล้ว ห้างค้าปลีกอื่นๆ อีกมากมายก็ใช้กฎความสัมพันธ์เพื่อปรับปรุงการดำเนินงานและเพิ่มผลประกอบการเช่นกัน

5.1

5.1.1 ความหมายและความสำคัญของกฎความสัมพันธ์ (Association Rules)

บทที่ 5

- กฎความสัมพันธ์ (อังกฤษ: Association Rules) เป็นกระบวนการหนึ่งในการทำ Data Mining ที่ได้รับความนิยมมาก โดยจะใช้ Association Rules ในการหาความสัมพันธ์ของข้อมูลสองชุดหรือมากกว่าสองชุดขึ้นไปภายในกลุ่มข้อมูลที่มีขนาดใหญ่ ในการหาความสัมพันธ์นั้นจะมีขั้นตอนวิธีการหาหลายวิธีด้วยกัน แต่ขั้นตอนวิธีที่เป็นที่รู้จักและใช้อย่างแพร่หลายคือ ขั้นตอนวิธี Apriori
- ตัวอย่างหนึ่งของ Association Rules ที่ใช้กันก็คือ Market Basket Analysis ที่ใช้ในการหาความสัมพันธ์ของสินค้าที่ลูกค้ามักจะซื้อพร้อมกัน เพื่อใช้ในการจัดรายการส่งเสริมการขาย

5.1

5.1.2 ตัวอย่างความสำคัญของกฎความสัมพันธ์

บทที่ 5

กฎความสัมพันธ์มีความสำคัญมากในการวิเคราะห์ข้อมูลและการบริหารจัดการธุรกิจด้วยหลายด้าน โดยความสำคัญของกฎความสัมพันธ์ มีดังนี้

1. **ค้นพบความสัมพันธ์ที่ซ่อนอยู่:** กฎความสัมพันธ์ช่วยในการค้นพบความสัมพันธ์ที่อาจไม่เป็นที่รู้จักหรือซ่อนอยู่ภายในข้อมูลที่มีปริมาณมาก นี้ช่วยในการเข้าใจแนวโน้มและความสัมพันธ์ที่มีอิทธิพลต่อผลลัพธ์ทางธุรกิจ
2. **การทำนายและการตัดสินใจทางธุรกิจ:** กฎความสัมพันธ์สามารถใช้ในการสร้างแบบจำลองทางสถิติหรือการเรียนรู้ของเครื่องเพื่อทำนายผลลัพธ์หรือตัดสินใจทางธุรกิจ นี้ช่วยในการวิเคราะห์แนวโน้มและการทำนายในสถานการณ์ต่าง ๆ
3. **การปรับแต่งแผนการตลาด:** กฎความสัมพันธ์สามารถใช้ในการวิเคราะห์พฤติกรรมของลูกค้าและกลุ่มเป้าหมาย, ทำให้ธุรกิจสามารถปรับแผนการตลาดในทิศทางที่เหมาะสม
4. **การตรวจจับและป้องกันปัญหา:** กฎความสัมพันธ์ช่วยในการตรวจจับแนวโน้มที่ไม่ปกติหรือข้อมูลที่ผิดปกติที่อาจชี้ว่ามีปัญหาเกิดขึ้น, ทั้งในด้านความปลอดภัย, การฉ้อโกง, หรือปัญหาอื่น ๆ

5.1

5.1.2 ความสำคัญของกฎความสัมพันธ์

บทที่ 5

5. การจัดกลุ่มลูกค้า (Customer Segmentation): การเข้าใจและกำหนดกฎความสัมพันธ์ระหว่างลูกค้าช่วยในการแบ่งกลุ่มลูกค้าตามลักษณะที่คล้ายกัน, ซึ่งสามารถนำไปใช้ในกลยุทธ์การตลาดที่เป็นเอกลักษณ์สำหรับแต่ละกลุ่ม

6. การพัฒนาสินค้าและบริการ: การเข้าใจความต้องการและความพึงพอใจของลูกค้าทำให้ธุรกิจสามารถพัฒนาสินค้าและบริการได้ตรงกับความต้องการ

7. การปรับแต่งประสบการณ์ลูกค้า: การนำกฎความสัมพันธ์มาปรับแต่งประสบการณ์ลูกค้า, ทำให้ธุรกิจสามารถให้บริการและปรับปรุงประสบการณ์ลูกค้าให้ดียิ่งขึ้น

8. การลดความสูญเสียและเพิ่มกำไร: การนำกฎความสัมพันธ์มาใช้ในการวิเคราะห์พฤติกรรมของลูกค้าช่วยในการลดความสูญเสียและเพิ่มกำไรด้วยการปรับแผนการตลาดและการบริหารราคา

การนำกฎความสัมพันธ์มาใช้ในธุรกิจมีผลดีต่อการวิเคราะห์และการบริหารจัดการ, ช่วยให้ธุรกิจสามารถปรับตัวได้อย่างรวดเร็วตามเปลี่ยนแปลงของตลาดและความต้องการของลูกค้า

5.1

5.1.3 ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล

บทที่ 5

กฎความสัมพันธ์ (Association rule) เป็นเทคนิคหนึ่งของการวิเคราะห์ข้อมูล (Data mining) ที่ใช้เพื่อค้นหาความสัมพันธ์ระหว่างรายการข้อมูล (Itemset) สองรายการขึ้นไป โดยกฎความสัมพันธ์จะระบุรายการข้อมูลสองรายการว่ามีความเกี่ยวข้องกันหรือไม่ และมีความเกี่ยวข้องกันมากน้อยเพียงใด

การวิเคราะห์ข้อมูลเป็นกระบวนการในการค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล โดยอาศัยเทคนิคและเครื่องมือทางคณิตศาสตร์และสถิติ การวิเคราะห์ข้อมูลสามารถแบ่งออกเป็นประเภทต่างๆ ได้มากมาย ขึ้นอยู่กับวัตถุประสงค์ของการวิเคราะห์ข้อมูล กฎความสัมพันธ์เป็นเทคนิคการวิเคราะห์ข้อมูลประเภทหนึ่งที่สามารถประยุกต์ใช้ได้หลายสถานการณ์ เช่น

5.1

5.1.3 ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล

บทที่ 5

- การวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า เพื่อระบุสินค้าที่มีความสัมพันธ์กัน เช่น สินค้าประเภทขนมปังและนม มักจะถูกซื้อร่วมกัน
- การวิเคราะห์ข้อมูลทางการแพทย์ เพื่อระบุปัจจัยที่มีความสัมพันธ์กับการเกิดโรค เช่น ปัจจัยเสี่ยงต่อโรคหัวใจ
- การวิเคราะห์ข้อมูลทางการตลาด เพื่อระบุกลุ่มเป้าหมายของผลิตภัณฑ์หรือบริการ

ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล สามารถสรุปได้ดังนี้

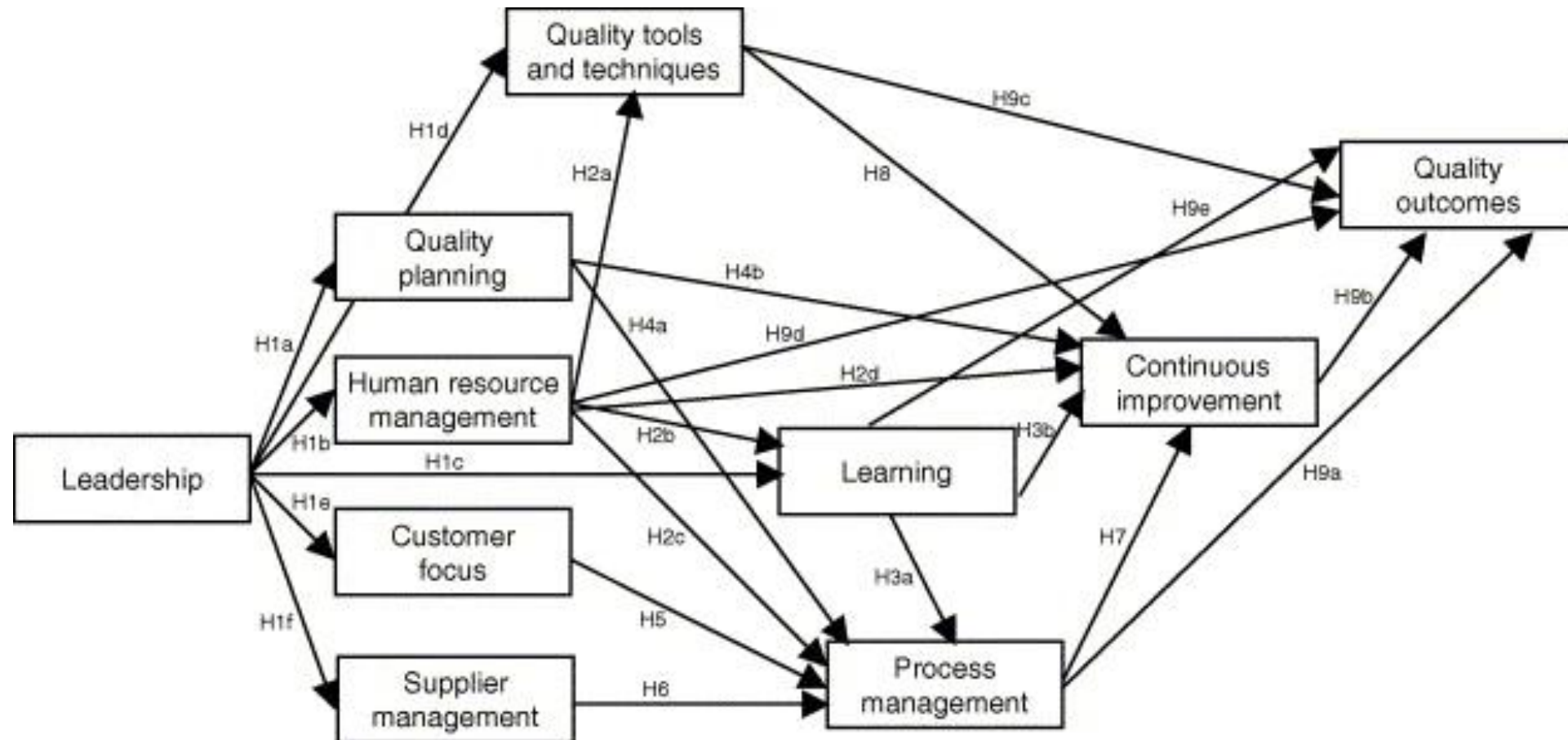
- กฎความสัมพันธ์เป็นเทคนิคการวิเคราะห์ข้อมูลประเภทหนึ่ง
- กฎความสัมพันธ์ใช้เพื่อค้นหาความสัมพันธ์ระหว่างรายการข้อมูล
- กฎความสัมพันธ์สามารถประยุกต์ใช้ได้หลายสถานการณ์

โดยสรุปแล้ว กฎความสัมพันธ์เป็นเทคนิคการวิเคราะห์ข้อมูลที่สามารถใช้ในการค้นหาความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล ซึ่งสามารถนำไปประยุกต์ใช้ได้หลากหลายสถานการณ์ เพื่อประโยชน์ในการตัดสินใจและวางแผนต่างๆ

5.1

5.1.3 ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล

บทที่ 5



ภาพที่ 5.1 ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล

5.1 5.1.3 ความเชื่อมโยงระหว่างกฎความสัมพันธ์กับการวิเคราะห์ข้อมูล

จากรูป 5.1 กฎความสัมพันธ์เป็นเทคนิคการวิเคราะห์ข้อมูลประเภทหนึ่งที่สามารถใช้ในการค้นหาความสัมพันธ์ระหว่างรายการข้อมูล รายการข้อมูลสามารถเป็นข้อมูลใดๆ ก็ได้ เช่น สินค้า ลูกค้า เหตุการณ์ หรือข้อมูลอื่นๆ กฎความสัมพันธ์สามารถประยุกต์ใช้ได้หลายสถานการณ์ เช่น การวิเคราะห์พฤติกรรมการณ์ซื้อสินค้าของลูกค้า การวิเคราะห์ข้อมูลทางการแพทย์ และการวิเคราะห์ข้อมูลทางการตลาด

- ตัวอย่างเช่น ร้านค้าปลีกสามารถใช้กฎความสัมพันธ์เพื่อวิเคราะห์ข้อมูลการซื้อสินค้าของลูกค้า เพื่อระบุสินค้าที่มีความสัมพันธ์กัน เช่น สินค้าประเภทขนมปังและนม มักจะถูกซื้อร่วมกัน ร้านค้าปลีกสามารถใช้ข้อมูลนี้เพื่อวางสินค้าให้เหมาะสมร่วมกัน เพื่อกระตุ้นให้ลูกค้าซื้อสินค้ามากขึ้น
- อีกตัวอย่างหนึ่ง โรงพยาบาลสามารถใช้กฎความสัมพันธ์เพื่อวิเคราะห์ข้อมูลทางการแพทย์ เพื่อระบุปัจจัยที่มีความสัมพันธ์กับการเกิดโรค เช่น ปัจจัยเสี่ยงต่อโรคหัวใจ โรงพยาบาลสามารถใช้ข้อมูลนี้เพื่อพัฒนาโปรแกรมป้องกันโรคหัวใจ

5.2

พื้นฐานของกฎความสัมพันธ์

5.2.1 กลุ่มข้อมูล (Itemset)

5.2.2 กฎความสัมพันธ์ (Association Rule)

5.2.3 ค่าสนับสนุน (Support)

5.2.4 ค่าเชื่อมั่น (Confidence)

5.2.5 ค่าเชื่อมั่นที่ต่ำสุด (Minimum Confidence)

5.2

5.2 พื้นฐานของกฎความสัมพันธ์

บทที่ 5

พื้นฐานของกฎความสัมพันธ์นั้นอยู่ในการค้นหาความสัมพันธ์ระหว่างกลุ่มข้อมูลหรือรายการต่าง ๆ ในชุดข้อมูลนี้ เกี่ยวข้องกับการพัฒนากฎทางสถิติที่บอกถึงลักษณะของข้อมูลที่มีความเกี่ยวข้องกัน โดยพื้นฐานที่สำคัญของกฎความสัมพันธ์ มีดังนี้

1) กลุ่มข้อมูล (Itemset): เป็นเซตของรายการที่ปรากฏพร้อมกันในชุดข้อมูล Itemset ที่มีจำนวนรายการมากกว่า 1 เรียกว่า Itemset ที่มีขนาด (size) มากกว่า 1 ยกตัวอย่างเช่น, {A, B} เป็น Itemset ขนาด 2

2) กฎความสัมพันธ์ (Association Rule): กฎที่ระบุความสัมพันธ์ระหว่าง Itemset ในรูปแบบ "ถ้ามี A แล้ว B ก็เกิดขึ้น" หรือในรูปย่อ " $A \rightarrow B$ " กฎความสัมพันธ์มีสองส่วนคือ ส่วนก่อนเงื่อนไข (Antecedent) และส่วนหลังเงื่อนไข (Consequent)

3) ค่าสนับสนุน (Support): ความถี่ที่กลุ่มข้อมูลหรือ Itemset ปรากฏในชุดข้อมูล ค่าสนับสนุนคำนวณจากจำนวนรายการที่ปรากฏ Itemset นั้น ๆ หารด้วยจำนวนรวมของ Itemset ทั้งหมด

5.2

5.2 พื้นฐานของกฎความสัมพันธ์

บทที่ 5

4) ค่าเชื่อมั่น (Confidence): ค่าเชื่อมั่นบ่งบอกถึงความน่าจะเป็นที่กฎความสัมพันธ์จะเป็นจริง, เมื่อ Itemset ในส่วนก่อนเงื่อนไข (Antecedent) ปรากฏ, แล้ว Itemset ในส่วนหลังเงื่อนไข (Consequent) ก็จะปรากฏ

5) ค่าเชื่อมั่นที่ต่ำสุด (Minimum Confidence): ค่าที่กำหนดให้กฎความสัมพันธ์ถูกยอมรับเป็นถูกต้อง กฎที่มีค่าเชื่อมั่นต่ำกว่าค่าที่กำหนดจะถือว่าไม่สนับสนุนความสัมพันธ์นั้น

การใช้ค่าสนับสนุนและค่าเชื่อมั่นที่ต่ำสุดช่วยในการควบคุมความละเอียดของกฎความสัมพันธ์ที่ถูกสร้างขึ้น, ทำให้เฉพาะกฎที่มีความสัมพันธ์ที่มีนัยยะมีความน่าสนใจถูกนำเสนอ

การใช้กฎความสัมพันธ์ในการวิเคราะห์ข้อมูลมีประโยชน์มากในการค้นพบความสัมพันธ์ที่ซ่อนอยู่, ทำนายแนวโน้ม, และปรับปรุงการตัดสินใจทางธุรกิจ

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

บทที่ 5

การทำเหมืองข้อมูล (Data Mining) เป็นกระบวนการที่ใช้เทคนิคต่าง ๆ เพื่อค้นหาความรู้ที่ซ่อนอยู่ในข้อมูล ซึ่งสามารถนำไปใช้ในการตัดสินใจหรือการวิเคราะห์ ประเภทข้อมูลสำหรับการทำเหมืองข้อมูลมีหลายประการ และขึ้นอยู่กับลักษณะของข้อมูลที่เราต้องการทำเหมือง ตัวอย่างข้อมูลที่ถูกนำมาใช้ในการทำเหมืองข้อมูล มีดังนี้

1. **ข้อมูลสถิติ (Numerical Data)** คือข้อมูลที่มีลักษณะเป็นตัวเลข เช่น อายุ, รายได้, น้ำหนัก, ความยาว เป็นต้น.
2. **ข้อมูลแบบประเภท (Categorical Data)** คือข้อมูลที่มีลักษณะเป็นประเภทหรือหมวดหมู่ เช่น เพศ, สี, ประเภทของสินค้า.
3. **ข้อมูลที่มีลำดับ (Ordinal Data)** คือข้อมูลที่มีลำดับหรือลำดับความสำคัญ เช่น ระดับการสำเร็จการศึกษา (มัธยม, ปวช., ปวส.).
4. **ข้อมูลที่มีโครงสร้าง (Structured Data)** คือข้อมูลที่ถูกจัดเก็บในรูปแบบที่มีโครงสร้าง, เช่น ข้อมูลในฐานข้อมูลที่มีตารางและคอลัมน์.
5. **ข้อมูลที่ไม่มีโครงสร้าง (Unstructured Data)** คือข้อมูลที่ไม่มีโครงสร้างที่แน่นอน, เช่น ข้อความ, รูปภาพ, ไฟล์เสียง.

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

บทที่ 5

6. ข้อมูลชุดเวลา (Time Series Data) คือข้อมูลที่มีการจัดเก็บตามช่วงเวลา, เช่น ข้อมูลการขายประจำเดือน, ข้อมูลอุณหภูมิประจำวัน.
7. ข้อมูลหลายมิติ (Multidimensional Data) คือข้อมูลที่มีหลายมิติ, เช่น ข้อมูลที่มีการวัดทั้งหลายทิศทาง (ตัวแปรหลายตัว).
8. ข้อมูลกราฟ (Graph Data) คือข้อมูลที่ถูกแสดงในรูปแบบกราฟ, เช่น ข้อมูลเครือข่ายโซเชียล, กราฟการจราจร.
9. ข้อมูลเปรียบเทียบ (Comparative Data) คือข้อมูลที่ใช้เปรียบเทียบระหว่างกลุ่ม, เช่น ผลประเมินหลายกลุ่ม.

การทำเหมืองข้อมูลมักใช้เทคนิคและวิธีการที่เหมาะสมกับประเภทข้อมูลและวัตถุประสงค์ของการทำเหมืองข้อมูลนั้น ๆ

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

บทที่ 5

ข้อมูลสถิติ (Numerical Data) คือข้อมูลที่มีลักษณะเป็นตัวเลข เช่น อายุ, รายได้, น้ำหนัก, ความยาว เป็นต้น.

ตัวอย่าง	ข้อมูลสถิติเชิงตัวเลข
1	อายุนักเรียนชั้นมัธยมศึกษาปีที่ 6
2	คะแนนสอบวิชาคณิตศาสตร์ของนักเรียนชั้นมัธยมศึกษาปีที่ 5
3	น้ำหนักของเด็กทารกแรกเกิด
4	ความยาวของไม้บรรทัด
5	อุณหภูมิของน้ำในสระว่ายน้ำ
6	จำนวนรถยนต์ที่วิ่งบนถนนสายหลัก
7	ยอดขายสินค้าของห้างสรรพสินค้า
8	ปริมาณฝนที่ตกในแต่ละวัน
9	ความเร็วลมที่พัดผ่านเสาอากาศ

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

บทที่ 5

ตัวอย่างข้อมูลแบบประเภท (Categorical Data) ได้แก่

- เพศ (ชาย/หญิง)
- สัญชาติ (ไทย/อังกฤษ/จีน/ญี่ปุ่น)
- ระดับการศึกษา (ประถมศึกษา/มัธยมศึกษา/อุดมศึกษา)
- อาชีพ (พนักงานบริษัท/ข้าราชการ/เกษตรกร)
- ประเภทสินค้า (อาหาร/เสื้อผ้า/เครื่องใช้ไฟฟ้า)
- ประเภทภาพยนตร์ (แอ็คชั่น/โรแมนติก/คอมеди)
- ประเภทเพลง (ป๊อป/ร็อก/คลาสสิก)

ตัวอย่าง	ข้อมูลแบบประเภท
1	เพศ
2	สัญชาติ
3	ระดับการศึกษา
4	อาชีพ
5	ประเภทสินค้า
6	ประเภทภาพยนตร์
7	ประเภทเพลง
8	ประเภทอาหาร
9	ประเภทสถานที่ท่องเที่ยว

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

บทที่ 5

ข้อมูลสถิติ (Numerical Data) คือข้อมูลที่มีลักษณะเป็นตัวเลข เช่น อายุ, รายได้, น้ำหนัก, ความยาว เป็นต้น.

ตัวอย่าง	ข้อมูลสถิติเชิงตัวเลข
1	อายุนักเรียนชั้นมัธยมศึกษาปีที่ 6
2	คะแนนสอบวิชาคณิตศาสตร์ของนักเรียนชั้นมัธยมศึกษาปีที่ 5
3	น้ำหนักของเด็กทารกแรกเกิด
4	ความยาวของไม้บรรทัด
5	อุณหภูมิของน้ำในสระว่ายน้ำ
6	จำนวนรถยนต์ที่วิ่งบนถนนสายหลัก
7	ยอดขายสินค้าของห้างสรรพสินค้า
8	ปริมาณฝนที่ตกในแต่ละวัน
9	ความเร็วลมที่พัดผ่านเสาอากาศ

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

กลุ่มข้อมูล (Itemset): เป็นเซตของรายการที่ปรากฏพร้อมกันในชุดข้อมูล Itemset ที่มีจำนวนรายการมากกว่า 1 เรียกว่า Itemset ที่มีขนาด (size) มากกว่า 1 ยกตัวอย่างเช่น, {A, B} เป็น Itemset ขนาด 2

รายการสินค้า

- ขนมห้าง
- นม
- ไข่
- กาแฟ
- น้ำผลไม้

ตัวอย่างกลุ่มข้อมูล(Itemset) สำหรับ
สินค้า ได้แก่:

- {ขนมห้าง, นม}
- {ไข่, นม}
- {กาแฟ, น้ำผลไม้}

1	{ขนมห้าง, นม}
2	{ไข่, นม}
3	{กาแฟ, น้ำผลไม้}
4	{ลูกคำ A, ลูกคำ B}
5	{ลูกคำ B, ลูกคำ C}
6	{ลูกคำ C, ลูกคำ D}
7	{เหตุการณ์ A, เหตุการณ์ B}
8	{เหตุการณ์ B, เหตุการณ์ C}
9	{เหตุการณ์ C, เหตุการณ์ D}

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

กลุ่มข้อมูล (Itemset): เป็นเซตของรายการที่ปรากฏพร้อมกันในชุดข้อมูล. Itemset ที่มีจำนวนรายการมากกว่า 1 เรียกว่า Itemset ที่มีขนาด (size) มากกว่า 1. ยกตัวอย่างเช่น, {A, B} เป็น Itemset ขนาด 2.

รายการลูกค้า

- ลูกค้า A
- ลูกค้า B
- ลูกค้า C
- ลูกค้า D

ตัวอย่างกลุ่มข้อมูล(Itemset) สำหรับลูกค้า ได้แก่:

- {ลูกค้า A, ลูกค้า B}
- {ลูกค้า B, ลูกค้า C}
- {ลูกค้า C, ลูกค้า D}

1	{ขนมปัง, นม}
2	{ไข่, นม}
3	{กาแฟ, น้ำผลไม้}
4	{ลูกคำ A, ลูกคำ B}
5	{ลูกคำ B, ลูกคำ C}
6	{ลูกคำ C, ลูกคำ D}
7	{เหตุการณ์ A, เหตุการณ์ B}
8	{เหตุการณ์ B, เหตุการณ์ C}
9	{เหตุการณ์ C, เหตุการณ์ D}

 ส่งออกไปยังซีดี

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

ตัวอย่างข้อมูล 5 ชุดที่ใช้สำหรับเตรียมกลุ่มข้อมูล (Itemset) เพื่อหากฎความสัมพันธ์:

1. Transaction 1: A, B, C, D
2. Transaction 2: B, D, E
3. Transaction 3: A, C, E
4. Transaction 4: B, C, D
5. Transaction 5: A, B

在这里, A, B, C, D, E คือรายการสินค้าที่ปรากฏในแต่ละธุรกรรม คุณสามารถใช้ข้อมูลนี้เพื่อทดสอบการหาความสัมพันธ์ที่เกี่ยวข้องกับ Itemset ต่าง ๆ ในชุดข้อมูลของคุณ

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

รายการสินค้าเหล่านี้เป็นรายการสินค้าทั่วไปที่พบได้บ่อยในชีวิตประจำวัน

ลำดับ	ชื่อรายการสินค้า
1	ขนมปัง
2	นม
3	ไข่
4	กาแฟ
5	น้ำผลไม้
6	ข้าว
7	ปลา
8	เนื้อสัตว์
9	ผัก
10	ผลไม้

ตัวอย่างกลุ่มข้อมูล(Itemset) สำหรับรายการสินค้า ได้แก่:

- {ขนมปัง, ไข่}
- {นม, ไข่, ปลา}
- {นม, ไข่, ผัก, ข้าว}

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

กลุ่มข้อมูล (Itemset): เป็นเซตของรายการที่ปรากฏพร้อมกันในชุดข้อมูล Itemset ที่มีจำนวนรายการมากกว่า 1 เรียกว่า Itemset ที่มีขนาด (size) มากกว่า 1 ยกตัวอย่างเช่น, {A, B} เป็น Itemset ขนาด 2

รายการเหตุการณ์

- เหตุการณ์ A
- เหตุการณ์ B
- เหตุการณ์ C
- เหตุการณ์ D

ตัวอย่างกลุ่มข้อมูลสำหรับเหตุการณ์ ได้แก่:

- {เหตุการณ์ A, เหตุการณ์ B}
- {เหตุการณ์ B, เหตุการณ์ C}
- {เหตุการณ์ C, เหตุการณ์ D}

1	{ขนมปัง, นม}
2	{ไข่, นม}
3	{กาแฟ, น้ำผลไม้}
4	{ลูกคำ A, ลูกคำ B}
5	{ลูกคำ B, ลูกคำ C}
6	{ลูกคำ C, ลูกคำ D}
7	{เหตุการณ์ A, เหตุการณ์ B}
8	{เหตุการณ์ B, เหตุการณ์ C}
9	{เหตุการณ์ C, เหตุการณ์ D}

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

โดยทั่วไปแล้ว กลุ่มข้อมูลสามารถเป็นข้อมูลใดๆ ก็ได้ ขึ้นอยู่กับวัตถุประสงค์ของการวิเคราะห์ข้อมูล ตัวอย่างเช่น รายการข้อมูลสามารถเป็นสินค้า ลูกค้า เหตุการณ์ หรือข้อมูลอื่น ๆ ก็ได้

ต่อไปนี้เป็นตัวอย่างเพิ่มเติมของกลุ่มข้อมูล:

รายการคำ

- {หมา, แมว}
- {บ้าน, รถ}
- {รัก, ชอบ}

รายการตัวเลข

- {1, 2, 3}
- {4, 5, 6}
- {7, 8, 9}

รายการสัญลักษณ์

- {A, B, C}
- {D, E, F}
- {G, H, I}

1	{ขนมปัง, นม}
2	{ไข่, นม}
3	{กาแฟ, น้ำผลไม้}
4	{ลูกค้า A, ลูกค้า B}
5	{ลูกค้า B, ลูกค้า C}
6	{ลูกค้า C, ลูกค้า D}
7	{เหตุการณ์ A, เหตุการณ์ B}
8	{เหตุการณ์ B, เหตุการณ์ C}
9	{เหตุการณ์ C, เหตุการณ์ D}
📦 ส่งออกไปยังซีดี	

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

บทที่ 5

ตารางนี้แสดงรายการข้อมูลทั้งหมดที่มีอยู่ในชุดข้อมูล รายการข้อมูลแต่ละรายการแสดงอยู่ในรายการแยกกัน โดยลำดับของรายการข้อมูลในรายการไม่ได้มีความสำคัญ

นอกจากนี้ ยังสามารถแสดงกลุ่มข้อมูลในรูปแบบตารางที่แสดงรายการข้อมูลทั้งหมดที่มีอยู่ในชุดข้อมูล และจำนวนครั้งที่รายการข้อมูลแต่ละรายการปรากฏในชุดข้อมูลได้อีกด้วย ตัวอย่างเช่น:

1	{ขนมปัง, นม}
2	{ไข่, นม}
3	{กาแฟ, น้ำผลไม้}
4	{ลูกคำ A, ลูกคำ B}
5	{ลูกคำ B, ลูกคำ C}
6	{ลูกคำ C, ลูกคำ D}
7	{เหตุการณ์ A, เหตุการณ์ B}
8	{เหตุการณ์ B, เหตุการณ์ C}
9	{เหตุการณ์ C, เหตุการณ์ D}
📦 ส่งออกไปยังซีด	

1	{ขนมปัง, นม}	2
2	{ไข่, นม}	1
3	{กาแฟ, น้ำผลไม้}	1
4	{ลูกคำ A, ลูกคำ B}	2
5	{ลูกคำ B, ลูกคำ C}	1
6	{ลูกคำ C, ลูกคำ D}	1
7	{เหตุการณ์ A, เหตุการณ์ B}	2
8	{เหตุการณ์ B, เหตุการณ์ C}	1
9	{เหตุการณ์ C, เหตุการณ์ D}	1
		📦 ส่งออกไปยังซีด

5.2

5.2.2 กลุ่มข้อมูล (Itemset):

File:

ตัวอย่างที่ 1 การแสดง items

Lec05_01_assocication_rule_Itemset.py

```
import pandas as pd
# สร้าง DataFrame จากข้อมูล
data = {
    'ลำดับ': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'ชื่อรายการสินค้า': ['ขนมปัง', 'นม', 'ไข่', 'กาแฟ', 'น้ำผลไม้',
                        'ข้าว', 'ปลา', 'เนื้อสัตว์', 'ผัก', 'ผลไม้']
}
df = pd.DataFrame(data)
# แสดง DataFrame
print(df)
```

ลำดับ	ชื่อรายการสินค้า
1	ขนมปัง
2	นม
3	ไข่
4	กาแฟ
5	
6	
7	
8	
9	
10	

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\AppServ\www\Python_DataMining> & C:/U
ลำดับ ชื่อรายการสินค้า
0      1      ขนมปัง
1      2      นม
2      3      ไข่
3      4      กาแฟ
4      5      น้ำผลไม้
5      6      ข้าว
6      7      ปลา
7      8      เนื้อสัตว์
8      9      ผัก
9     10      ผลไม้
PS C:\AppServ\www\Python_DataMining> |
```

5.2

5.2.3 กฎความสัมพันธ์ (Association Rule)

บทที่ 5

กฎความสัมพันธ์ (Association Rule)

เป็นเทคนิคหนึ่งของการวิเคราะห์ข้อมูล (Data Mining) ที่ใช้เพื่อค้นหาคำความสัมพันธ์ระหว่างรายการข้อมูล (Itemset) สองรายการขึ้นไป โดยกฎความสัมพันธ์จะระบุรายการข้อมูลสองรายการว่ามีความเกี่ยวข้องกันหรือไม่ และมีความเกี่ยวข้องกันมากน้อยเพียงใด

รายการข้อมูล (Itemset) คือชุดของข้อมูลใด ๆ ก็ได้ เช่น สินค้า ลูกค้า เหตุการณ์ หรือข้อมูลอื่นๆ กฎความสัมพันธ์จะพิจารณาว่ารายการข้อมูลสองรายการมีความเกี่ยวข้องกันหรือไม่ โดยพิจารณาจากความถี่ที่รายการข้อมูลทั้งสองรายการปรากฏร่วมกันในชุดข้อมูล

กฎความสัมพันธ์สามารถเขียนในรูปแบบต่อไปนี้:

$$A \rightarrow B$$

5.2

5.2.3 กฎความสัมพันธ์ (Association Rule)

ซึ่งหมายความว่า รายการข้อมูล A มักจะปรากฏพร้อมกับรายการข้อมูล B ในชุดข้อมูล

ตัวอย่างเช่น กฎความสัมพันธ์ต่อไปนี้ระบุว่า ลูกค้ายักษ์จะซื้อสินค้าประเภทขนมปังและนมร่วมกัน:

{ขนมปัง, นม} $\Rightarrow$ {นม}

$A \Rightarrow B$

5.2

5.2.3 กฎความสัมพันธ์ (Association Rule)

บทที่ 5

กฎความสัมพันธ์สามารถประยุกต์ใช้ได้หลายสถานการณ์ เช่น

การวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า เพื่อระบุสินค้าที่มีความสัมพันธ์กัน เช่น สินค้าประเภทขนมปังและนม มักจะถูกซื้อร่วมกัน ร้านค้าปลีกสามารถใช้ข้อมูลนี้เพื่อวางสินค้าให้เหมาะสมร่วมกัน เพื่อกระตุ้นให้ลูกค้าซื้อสินค้ามากขึ้น

การวิเคราะห์ข้อมูลทางการแพทย์ เพื่อระบุปัจจัยที่มีความสัมพันธ์กับการเกิดโรค เช่น ปัจจัยเสี่ยงต่อโรคหัวใจ โรงพยาบาลสามารถใช้ข้อมูลนี้เพื่อพัฒนาโปรแกรมป้องกันโรคหัวใจ

การวิเคราะห์ข้อมูลทางการตลาด เพื่อระบุกลุ่มเป้าหมายของผลิตภัณฑ์หรือบริการ เช่น ลูกค้าที่มีแนวโน้มที่จะซื้อผลิตภัณฑ์หรือบริการ หน่วยงานการตลาดสามารถใช้ข้อมูลนี้เพื่อกำหนดกลยุทธ์ทางการตลาดที่เหมาะสม

5.2

5.2.4 ขั้นตอนในการค้นหากฎความสัมพันธ์

บทที่ 5

ขั้นตอนในการค้นหากฎความสัมพันธ์ ประกอบไปด้วย 3 ขั้นตอนมีดังนี้

1. การหา frequent itemsets

ขั้นตอนแรกคือ การหา frequent itemsets ซึ่งเป็นรายการข้อมูลที่มีจำนวนครั้งที่ปรากฏในชุดข้อมูลมากกว่าเกณฑ์ที่กำหนด

2. การหากฎความสัมพันธ์

จาก frequent itemsets ที่ได้จากการหาในขั้นตอนแรก สามารถใช้วิธีการต่าง ๆ เพื่อหากฎความสัมพันธ์ เช่น

- วิธีการ Apriori
- วิธีการ Eclat

5.2

5.2.4 ขั้นตอนในการค้นหากฎความสัมพันธ์ (ต่อ...)

บทที่ 5

3. การคัดเลือกกฎความสัมพันธ์

กฎความสัมพันธ์ที่พบทั้งหมดอาจไม่มีความสำคัญเท่ากัน จึงต้องมีการคัดเลือกกฎความสัมพันธ์ที่มีนัยสำคัญทางสถิติ โดยพิจารณาจากตัววัดความสำคัญ เช่น

- ความถี่/ค่าสนับสนุน (Support)
- ความเชื่อมั่น (Confidence)
- ความแปลกใหม่ (Lift)

5.2

5.2.5 การหา frequent itemsets

บทที่ 5

ตัวอย่าง_02 : การหา frequent itemsets

```
# ติดตั้งไลบรารี mlxtend ถ้ายังไม่ได้ติดตั้ง
```

```
# pip install mlxtend
```

```
import pandas as pd # เพิ่มบรรทัดนี้
```

```
from mlxtend.preprocessing import TransactionEncoder
```

```
from mlxtend.frequent_patterns import apriori, association_rules
```

```
# ข้อมูลตัวอย่าง (Transaction Data)
```

```
transactions = [
```

```
    ['Milk', 'Bread', 'Eggs'],
```

```
    ['Bread', 'Eggs'],
```

```
    ['Milk', 'Bread', 'Cookies'],
```

```
    ['Milk', 'Eggs'],
```

```
    ['Bread', 'Cookies']
```

```
]
```

Part_01

ตัวอย่างที่ 02 : การหา frequent itemsets

File : Lec05_02_association_frequent_01.py

5.2

5.2.5 การหา frequent itemsets

บทที่ 5

ตัวอย่าง_01 : การหา frequent itemsets

Part_02

ตัวอย่างที่ 02 : การหา frequent itemsets

File : Lec05_02_association_frequent_01.py

แปลงข้อมูลเป็นรูปแบบที่เหมาะสมสำหรับใช้กับ Apriori Algorithm

te = TransactionEncoder()

te_ary = te.fit(transactions).transform(transactions)

df = pd.DataFrame(te_ary, columns=te.columns_) # ใช้ pd เพื่อสร้าง DataFrame

หา frequent itemsets ด้วย Apriori Algorithm

frequent_itemsets = apriori(df, min_support=0.2, use_colnames=True)

สร้าง association rules จาก frequent itemsets

rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)

แสดง frequent itemsets

print("Frequent Itemsets:")

print(frequent_itemsets)

แสดง association rules

print("\nAssociation Rules:")

print(rules)

5.2

5.2.5 การหา frequent itemsets

บทที่ 5

ตัวอย่าง_01 : การหา frequent itemsets

ผลลัพธ์

File : Lec05_02_association_frequent_01.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
4      0.4      (Bread, Cookies)
5      0.4      (Bread, Eggs)
6      0.4      (Bread, Milk)
7      0.2      (Cookies, Milk)
8      0.4      (Milk, Eggs)
9      0.2      (Bread, Cookies, Milk)
10     0.2      (Bread, Milk, Eggs)
```

Association Rules:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(Cookies)	(Bread)	0.4	0.8	0.4	1.0	1.25	0.08	inf	0.333333
1	(Cookies, Milk)	(Bread)	0.2	0.8	0.2	1.0	1.25	0.04	inf	0.250000

PS C:\AppServ\www\Python_DataMining> |

5.2

5.2.6 กฎความสัมพันธ์ (Association Rule)

บทที่ 5

ตัวอย่างของกฎความสัมพันธ์ที่มีนัยสำคัญทางสถิติ ได้แก่

- กฎความสัมพันธ์ที่มีความถี่สูง เช่น กฎความสัมพันธ์ที่ระบุว่า ลูกค้ามักจะซื้อสินค้าประเภทขนมปังและนมร่วมกัน
- กฎความสัมพันธ์ที่มีความเชื่อมั่นสูง เช่น กฎความสัมพันธ์ที่ระบุว่า ลูกค้ามักจะซื้อสินค้าประเภทขนมปังและนมร่วมกัน โดยลูกค้า 80% ที่ซื้อขนมปังมักจะซื้อนมด้วย
- กฎความสัมพันธ์ที่มีความแปลกใหม่สูง เช่น กฎความสัมพันธ์ที่ระบุว่า ลูกค้ามักจะซื้อสินค้าประเภทขนมปังและนมร่วมกัน โดยลูกค้า 20% ที่ซื้อนมมักจะซื้อขนมปังด้วย

กฎความสัมพันธ์สามารถนำไปประยุกต์ใช้ได้หลากหลายสถานการณ์ โดยมีประโยชน์ในการระบุความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล ซึ่งสามารถนำไปใช้ในการตัดสินใจและวางแผนต่างๆ ได้

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5

ค่าสนับสนุน (Support) เป็นค่าที่ใช้วัดความถี่ที่รายการข้อมูลสองรายการปรากฏร่วมกันในชุดข้อมูล ค่าสนับสนุนสามารถเขียนเป็นเปอร์เซ็นต์ได้ โดยคำนวณจากสูตรต่อไปนี้

$$\text{Support}(A, B) = (\text{จำนวนครั้งของ } (A, B) \text{ ปรากฏในชุดข้อมูล}) / (\text{จำนวนรายการข้อมูลทั้งหมดในชุดข้อมูล})$$

ตัวอย่างเช่น สมมติว่ามีชุดข้อมูลรายการซื้อสินค้าของลูกค้า 100 รายการ โดยมีรายการสินค้า 10 รายการ และพบรายการสินค้า {ขนมปัง, นม} ปรากฏร่วมกัน 2 ครั้ง ค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม} คือ:

$$\text{Support}(\{\text{ขนมปัง, นม}\}) = (2 / 100) * 100\% = 2\%$$

5.2

5.2.7 ค่าสนับสนุน (Support)

Itemset	Transaction ID				Support
	1	2	3	4	
{Apple, Beer}	0	0	1	0	$1/4 = 25\%$
{Apple, Cereal}	1	0	1	0	$2/4 = 50\%$
{Apple, Eggs}	0	0	1	0	$1/4 = 25\%$
{Beer, Cereal}	0	1	1	0	$2/4 = 50\%$
{Beer, Eggs}	0	1	1	1	$3/4 = 75\%$
{Cereal, Eggs}	0	1	1	0	$2/4 = 50\%$

ภาพที่ 4 แสดงค่า support ของ itemset ที่มีความยาว 2

กำหนดค่า minimum support = 50%

Itemset	Transaction ID				Support
	1	2	3	4	
{Apple, Beer}	0	0	1	0	$1/4 = 25\%$
{Apple, Cereal}	1	0	1	0	$2/4 = 50\%$
{Apple, Eggs}	0	0	1	0	$1/4 = 25\%$
{Beer, Cereal}	0	1	1	0	$2/4 = 50\%$
{Beer, Eggs}	0	1	1	1	$3/4 = 75\%$
{Cereal, Eggs}	0	1	1	0	$2/4 = 50\%$

ภาพที่ 5 แสดง frequent itemset ที่มีความยาว 2

5.2

5.2.7 ค่าสนับสนุน (Support)

- ค่าสนับสนุนสามารถใช้เพื่อวัดความสำคัญทางสถิติของกฎความสัมพันธ์ โดยกฎความสัมพันธ์ที่มีค่าสนับสนุนสูง แสดงว่ารายการข้อมูลสองรายการมีความเกี่ยวข้องกันสูง
- โดยทั่วไปแล้ว ค่าสนับสนุนที่ใช้ในการคัดเลือกกฎความสัมพันธ์จะกำหนดโดยผู้วิเคราะห์ข้อมูล โดยอาจกำหนดค่าสนับสนุนขั้นต่ำที่กฎความสัมพันธ์ต้องมี เพื่อให้มีนัยสำคัญทางสถิติ

ตัวอย่างเช่น สมมติว่ากำหนดค่าสนับสนุนขั้นต่ำไว้ที่ 10% กฎความสัมพันธ์ที่มีค่าสนับสนุนน้อยกว่า 10% จะถูกตัดทิ้ง

ค่าสนับสนุนสามารถใช้ประยุกต์ใช้ได้หลายสถานการณ์ เช่น

- การวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า เพื่อระบุสินค้าที่มีความสัมพันธ์กัน เช่น สินค้าประเภทขนมปังและนม มักจะถูกซื้อด้วยกัน ร้านค้าปลีกสามารถใช้ข้อมูลนี้เพื่อวางสินค้าให้เหมาะสมร่วมกัน เพื่อกระตุ้นให้ลูกค้าซื้อสินค้ามากขึ้น
- การวิเคราะห์ข้อมูลทางการแพทย์ เพื่อระบุปัจจัยที่มีความสัมพันธ์กับการเกิดโรค เช่น ปัจจัยเสี่ยงต่อโรคหัวใจ โรงพยาบาลสามารถใช้ข้อมูลนี้เพื่อพัฒนาโปรแกรมป้องกันโรคหัวใจ
- การวิเคราะห์ข้อมูลทางการตลาด เพื่อระบุกลุ่มเป้าหมายของผลิตภัณฑ์หรือบริการ เช่น ลูกค้าที่มีแนวโน้มที่จะซื้อผลิตภัณฑ์หรือบริการ หน่วยงานการตลาดสามารถใช้ข้อมูลนี้เพื่อกำหนดกลยุทธ์ทางการตลาดที่เหมาะสม

5.2

5.2.7 ค่าสนับสนุน (Support)

ตัวอย่างการหาค่า support ในภาษาไพธอน

ขนมปัง	นม	ไข่	กาแฟ	น้ำผลไม้
0	1	1	0	0
1	1	1	1	0
2	0	1	0	1
3	0	0	1	1
4	0	0	0	1

```
data = pd.DataFrame({
```

```
    "ขนมปัง": [1, 1, 0, 0, 0],
```

```
    "นม": [1, 1, 1, 0, 0],
```

```
    "ไข่": [0, 1, 0, 1, 0],
```

```
    "กาแฟ": [0, 0, 1, 1, 1],
```

```
    "น้ำผลไม้": [0, 0, 0, 1, 1]
```

```
})
```

ชุดข้อมูลที่กำหนดนี้เป็นตารางข้อมูล (DataFrame) ที่แสดงรายการสินค้าที่ถูกซื้อหรือไม่ถูกซื้อในแต่ละธุรกรรมหรือบรรทัดข้อมูล. ข้อมูลถูกเรียงตามชื่อสินค้าและแสดงในรูปของ 0 หรือ 1 ซึ่งแทนว่าคุณีสินค้านั้นถูกซื้อ (1) หรือไม่ถูกซื้อ (0) ในแต่ละธุรกรรม

ตัวอย่าง

แต่ละแถวแสดงธุรกรรมหนึ่ง ๆ และแต่ละคอลัมน์แสดงสินค้าที่เป็นไปได้. เช่น, ในธุรกรรมที่ 0 มีขนมปัง 1, นม 1, ไข่ 0, กาแฟ 0, น้ำผลไม้ 0 ซึ่งแปลว่ามีการซื้อขนมปังและนม แต่ไม่ซื้อไข่, กาแฟ, หรือน้ำผลไม้

ชุดข้อมูลแบบนี้มักถูกนำมาใช้ในการวิเคราะห์ Association Rules (กฎการเชื่อมโยง) โดยส่วนใหญ่เพื่อค้นหารูปแบบหรือความสัมพันธ์ที่เกิดขึ้นบ่อยในการซื้อสินค้า.

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5

```
import pandas as pd
# กำหนดชุดข้อมูล
data = pd.DataFrame({
    "ขนมปัง": [1, 1, 0, 0, 0],
    "นม": [1, 1, 1, 0, 0],
    "ไข่": [0, 1, 0, 1, 0],
    "กาแฟ": [0, 0, 1, 1, 1],
    "น้ำผลไม้": [0, 0, 0, 1, 1]
})
# คำนวณค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม}
support = data.iloc[:, :2].isin([1, 1]).sum() / data.shape[0]
# แสดงผลค่าสนับสนุน
print(support)
```

File :

ตัวอย่างที่ 5.3 การหาค่า support ในภาษาไพธอน
Lec05_03_association_rule_support01.py

ผลลัพธ์

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONITOR

PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/Local/Mi
ขนมปัง      0.4
นม          0.6
dtype: float64
PS C:\AppServ\www\Python_DataMining> |
```

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5

ตัวอย่างการหาค่า support ในภาษาไพธอน

ครั้ง

ได้ด้างต้นจะกำหนดชุดข้อมูลที่มีรายการสินค้า 5 รายการ โดยมีรายการสินค้า {ขนมปัง, นม} ปรากฏร่วมกัน 3

การคำนวณค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม} จะใช้ฟังก์ชัน `isin()` เพื่อตรวจสอบว่ารายการสินค้าทั้งสองรายการปรากฏร่วมกันในแถวใดในชุดข้อมูล

ผลลัพธ์ของได้ด้างต้นคือ:

0.6

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5

อธิบายโค้ดโปรแกรม

อธิบายโค้ดนี้ใช้ Pandas เพื่อคำนวณค่าสนับสนุน (support) ของรายการสินค้า {ขนมปัง, นม} จากชุดข้อมูลที่กำหนดใน DataFrame `data`

โดยมีขั้นตอนดังนี้

1. ``import pandas as pd``: นำเข้าไลบรารี Pandas และตั้งชื่อย่อเป็น ``pd`` เพื่อให้สามารถใช้ฟังก์ชันและคลาสของ Pandas ได้
2. กำหนดชุดข้อมูล ``data`` ในรูปแบบของ DataFrame ที่ประกอบไปด้วยคอลัมน์ต่างๆ แทนสินค้าต่างๆ และค่า 0 หรือ 1 บอกถึงการซื้อหรือไม่ซื้อสินค้านั้นๆ ในแต่ละบรรทัด
3. ``data.iloc[:, :2].isin([1, 1])``: ใช้ ``iloc`` เพื่อเลือกเฉพาะคอลัมน์ที่ 1 ถึง 2 (ไม่รวมคอลัมน์ที่ 2) จากนั้นใช้ ``isin([1, 1])`` เพื่อดูว่าค่าใน DataFrame เป็น 1, 1 หรือไม่
4. ``sum() / data.shape[0]``: นับจำนวนที่ค่าเป็น True ในแต่ละคอลัมน์แล้วหารด้วยจำนวนแถวทั้งหมดของ DataFrame เพื่อคำนวณสนับสนุน
5. ``print(support)``: แสดงผลค่าสนับสนุนที่คำนวณได้

ดังนั้น, โค้ดนี้เป็นตัวอย่างการคำนวณค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม} จากชุดข้อมูลที่กำหนดใน DataFrame `data`

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5

ตัวอย่างการหาค่า support ในภาษาไพธอน

ค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม} คือ 0.6 หรือ 60% ซึ่งแสดงว่ารายการสินค้าทั้งสองรายการมีความเกี่ยวข้องกันสูง นอกจากนี้ ยังสามารถหาค่าสนับสนุนของรายการสินค้าได้หลายวิธี เช่น การใช้ฟังก์ชัน support() จากโมดูล apyori

0.6

File :

ตัวอย่างที่ 5.4 การหาค่า support ในภาษาไพธอน
Lec05_04_association_rule_support02.py

```
from apyori import apriori
# กำหนดชุดข้อมูล
data = pd.DataFrame({
    "ขนมปัง": [1, 1, 0, 0, 0],
    "นม": [1, 1, 1, 0, 0],
    "ไข่": [0, 1, 0, 1, 0],
    "กาแฟ": [0, 0, 1, 1, 1],
    "น้ำผลไม้": [0, 0, 0, 1, 1]
})
```

หา frequent itemsets

```
frequent_itemsets = apriori(data,
min_support=0.5)
```

แสดงค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม}

```
print(frequent_itemsets[0][0][2])
```

Part01

ต่อ→Part02

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5

ตัวอย่างการหาค่า support ในภาษาไพธอน

ค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม} คือ 0.6 หรือ 60% ซึ่งแสดงว่ารายการสินค้าทั้งสองรายการมีความเกี่ยวข้องกันสูง นอกจากนี้ ยังสามารถหาค่าสนับสนุนของรายการสินค้าได้หลายวิธี เช่น การใช้ฟังก์ชัน support() จากโมดูล apyori

0.6

ผลลัพธ์ของโค้ดข้างต้นคือ:

File :

ตัวอย่างที่ 5.4 การหาค่า support ในภาษาไพธอน
Lec05_04_association_rule_support02.py

Part01

```
from apyori import apriori

# กำหนดชุดข้อมูล
data = pd.DataFrame( {"ขนมปัง": [1, 1, 0, 0, 0],      "นม": [1, 1, 1, 0, 0],
                      "ไข่": [0, 1, 0, 1, 0],      "กาแฟ": [0, 0, 1, 1, 1],      "น้ำผลไม้": [0, 0, 0, 1, 1]
                      })

# หา frequent itemsets
frequent_itemsets = apriori(data, min_support=0.5)

# แสดงค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม}
print(frequent_itemsets[0][0][2])
```

ต่อ→Part02

5.2

5.2.7 คำสนับสนุน (Support)

บทที่ 5

การติดตั้งไลบรารี apyori

ข้อผิดพลาดนี้เกิดขึ้นเนื่องจาก Python ไม่สามารถหาโมดูล `apyori` ได้ ซึ่งเป็นโมดูลที่ใช้ในการค้นหากฎความสัมพันธ์ (Association Rule) ในชุดข้อมูล

มีสองวิธีแก้ไขข้อผิดพลาดนี้:

1. ติดตั้งโมดูล `apyori`

คุณสามารถติดตั้งโมดูล `apyori` ด้วย pip:

pip install apyori

2. เอาโมดูล `apyori` จากไฟล์ `apyori.py`

โมดูล `apyori` เป็นไฟล์เดียว คุณสามารถดาวน์โหลดไฟล์จาก [\[https://github.com/ymoch/apyori\]](https://github.com/ymoch/apyori)(<https://github.com/ymoch/apyori>) และวางไว้ในไดเรกทอรีที่ Python สามารถค้นหาได้ (เช่น ไดเรกทอรีเดียวกับไฟล์สคริปต์ Python ของคุณ)

หลังจากที่คุณแก้ไขข้อผิดพลาดแล้ว คุณสามารถลองรันไฟล์สคริปต์ Python ของคุณอีกครั้ง

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/AppSer
Traceback (most recent call last):
  File "c:\AppServ\www\Python_DataMining\association_rule_support.py", line 1, in <module>
    from apyori import apriori
ModuleNotFoundError: No module named 'apyori'
PS C:\AppServ\www\Python_DataMining> █
```

5.2

5.2.7 คำสนับสนุน (Support)

บทที่ 5

การติดตั้งไลบรารี apyori

pip install apyori

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... done
Installing collected packages: apyori
  DEPRECATION: apyori is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for apyori ... done
Successfully installed apyori-1.1.2

[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: C:\Users\User\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
Requirement already satisfied: apyori in c:\users\user\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (1.1.2)

[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: C:\Users\User\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
PS C:\AppServ\www\Python_DataMining> |
```

5.2

5.2.7 คำสนับสนุน (Support)

บทที่ 5

การติดตั้งไลบรารี pandas

ติดตั้งโมดูล pandas ด้วย pip:

pip install pandas

ข้อผิดพลาดนี้เกิดจากการที่ Python ไม่สามารถหาโมดูล 'pandas' ได้ ซึ่งเป็นโมดูลที่ใช้ในการทำงานกับข้อมูลในรูปแบบ DataFrame

มีสองวิธีแก้ไขข้อผิดพลาดนี้:

****1. ติดตั้งโมดูล 'pandas'****

คุณสามารถติดตั้งโมดูล 'pandas' ด้วย pip:

...

pip install pandas

...

****2. เอาโมดูล 'pandas' จากไฟล์ 'pandas.py'****

โมดูล 'pandas' เป็นไฟล์เดียว คุณสามารถดาวน์โหลดไฟล์จาก https://github.com/pydata/pandas และวางไว้ในไดเรกทอรีที่ Python สามารถค้นหาได้ (เช่น ไดเรกทอรีเดียวกับไฟล์สคริปต์ Python ของคุณ)

หลังจากที่คุณแก้ไขข้อผิดพลาดแล้ว คุณสามารถลองรันไฟล์สคริปต์ Python ของคุณอีกครั้ง

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.
Traceback (most recent call last):
  File "c:\AppServ\www\Python_DataMining\association_rule_support.py", line 3, in <module>
    data = pd.DataFrame( {"ขนมปัง": [1, 1, 0, 0, 0], "นม": [1, 1, 1, 0, 0],
NameError: name 'pd' is not defined. Did you mean: 'id'?
PS C:\AppServ\www\Python_DataMining> |
```

5.2

5.2.7 ค่าสนับสนุน (Support)

การติดตั้งไลบรารี pandas

ติดตั้งโมดูล pandas ด้วย pip:

pip install pandas

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

```
PS C:\AppServ\www\Python_DataMining> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/AppServ/www/Python_DataMining/association_rule_support.py
Traceback (most recent call last):
  File "c:\AppServ\www\Python_DataMining\association_rule_support.py", line 3, in <module>
    data = pd.DataFrame( {"ขมมิ่ง": [1, 1, 0, 0, 0], "นม": [1, 1, 1, 0, 0],
NameError: name 'pd' is not defined. Did you mean: 'id'?
PS C:\AppServ\www\Python_DataMining> |
```

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5

การคำนวณค่าสนับสนุน (support) ของรายการสินค้า {ขนมปัง, นม} ในชุดข้อมูล ในที่นี้, เราต้องการหาว่ารายการสินค้า {ขนมปัง, นม} ปรากฏในธุรกรรมอย่างน้อย 1 ครั้ง

โดยคำนวณ support ตามสูตร:

ตัวอย่างการหาค่า support ในภาษาไพธอน

$\text{Support}(\{A,B\}) = \text{จำนวนธุรกรรมที่ปรากฏรายการสินค้า}\{A,B\} / \text{จำนวนทั้งหมดของธุรกรรม}$

$\text{Support}(\{A,B\}) = \text{จำนวนทั้งหมดของธุรกรรม} \times \text{จำนวนธุรกรรมที่ปรากฏรายการสินค้า}\{A,B\}$

เมื่อคำนวณเสร็จ, คุณจะได้ผลลัพธ์เป็น Series ที่แสดงค่า support ของแต่ละรายการสินค้า ลองรันโค้ดนี้เพื่อดูผลลัพธ์

5.2

5.2.7 ค่าสนับสนุน (Support)

บทที่ 5



ChatGPT

คุณกำลังคำนวณค่าสนับสนุน (support) ของรายการสินค้า {ขนมปัง, นม} ในชุดข้อมูลของคุณ. ในที่นี้, เราต้องการหาว่ารายการสินค้า {ขนมปัง, นม} ปรากฏในธุรกรรมอย่างน้อย 1 ครั้ง.

โดยคำนวณ support ตามสูตร:

$$\text{Support}(\{A, B\}) = \frac{\text{จำนวนธุรกรรมที่ปรากฏรายการสินค้า}\{A, B\}}{\text{จำนวนทั้งหมดของธุรกรรม}}$$

เมื่อคำนวณเสร็จ, คุณจะได้อผลลัพธ์เป็น Series ที่แสดงค่า support ของแต่ละรายการสินค้า. ลองรันโค้ดนี้เพื่อดูผลลัพธ์:

```
python
Copy code

import pandas as pd

# กำหนดชุดข้อมูล
data = pd.DataFrame({
    "ขนมปัง": [1, 1, 0, 0, 0],
    "นม": [1, 1, 1, 0, 0],
    "ไข่": [0, 1, 0, 1, 0],
    "กาแฟ": [0, 0, 1, 1, 1],
    "น้ำผลไม้": [0, 0, 0, 1, 1]
})

# คำนวณค่าสนับสนุนของรายการสินค้า {ขนมปัง, นม}
support = data.iloc[:, :2].isin([1, 1]).sum() / data.shape[0]

# แสดงผลค่าสนับสนุน
print(support)
```

ผลลัพธ์ที่คุณจะได้จะแสดงค่า support ของรายการสินค้า {ขนมปัง, นม} ในรูปแบบ Series.

5.2

5.2.8 ค่าเชื่อมั่น (Confidence)

บทที่ 5

- **ค่าเชื่อมั่น (Confidence):** ค่าเชื่อมั่นบ่งบอกถึงความน่าจะเป็นที่กฎความสัมพันธ์จะเป็นจริง, เมื่อ Itemset ในส่วนก่อนเงื่อนไข (Antecedent) ปรากฏ, แล้ว Itemset ในส่วนหลังเงื่อนไข (Consequent) ก็จะปรากฏ

ค่าเชื่อมั่น (Confidence) ในการทำ Association Rule Mining เป็นตัววัดความน่าเชื่อถือของกฎความสัมพันธ์ที่เราสร้างขึ้น เช่น $X \rightarrow Y$ (ถ้าซื้อ XXX แล้วมักจะซื้อ YYY ด้วย) โดยค่าเชื่อมั่นคำนวณจากสูตร:

5.2

5.2.8 ค่าเชื่อมั่น (Confidence)

บทที่ 5

ค่าเชื่อมั่น (Confidence) ในการทำ Association Rule Mining เป็นตัววัดความน่าเชื่อถือของกฎความสัมพันธ์ที่เราสร้างขึ้น เช่น $X \rightarrow Y$ (ถ้าซื้อ X แล้วมักจะซื้อ Y ด้วย) โดยค่าเชื่อมั่นคำนวณจากสูตร:

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

อธิบายสูตร

1. $\text{Support}(X \cup Y)$:

- ความถี่หรือสัดส่วนของลูกค้าที่ซื้อทั้ง X และ Y พร้อมกัน (หรือชุดสินค้า $\{X, Y\}$)
- คำนวณจาก:

$$\text{Support}(X \cup Y) = \frac{\text{จำนวนแถวที่ซื้อทั้ง } X \text{ และ } Y}{\text{จำนวนแถวทั้งหมด}}$$

2. $\text{Support}(X)$:

- ความถี่หรือสัดส่วนของลูกค้าที่ซื้อ X
- คำนวณจาก:

$$\text{Support}(X) = \frac{\text{จำนวนแถวที่ซื้อ } X}{\text{จำนวนแถวทั้งหมด}}$$

3. ผลลัพธ์ (Confidence):

- ค่าเชื่อมั่นบอกว่าในบรรดาลูกค้าทั้งหมดที่ซื้อ X มีกี่เปอร์เซ็นต์ที่ซื้อ Y ด้วย
- ค่า $\text{Confidence}(X \rightarrow Y)$ มีค่าอยู่ระหว่าง 0 ถึง 1 หรืออาจแสดงเป็นเปอร์เซ็นต์ (0% ถึง 100%)

5.2

5.2.8 ค่าเชื่อมั่น (Confidence)

ตัวอย่างการคำนวณ กำหนดชุดข้อมูลดังตาราง

กฎ: ขนมปัง→นม

คำนวณ ({Support}({ขนมปัง} ∪ {นม})

- ลูกค้าที่ซื้อทั้ง "ขนมปัง" และ "นม" คือ ลูกค้า 1 และ 2 (2 คน)
- จำนวนลูกค้าทั้งหมด = 5
- $\text{Support}(\text{ขนมปัง} \cup \text{นม}) = \frac{2}{5} = 0.4$ $\{ \text{Support} \} (\{ \text{ขนมปัง} \} + \{ \text{นม} \}) = \{ 2 \} \{ 5 \} = 0.4$ $\text{Support}(\text{ขนมปัง} \cup \text{นม}) = \frac{2}{5} = 0.4$

คำนวณ ({Support}({ขนมปัง})

- ลูกค้าที่ซื้อ "ขนมปัง" คือ ลูกค้า 1, 2, และ 5 (3 คน)
- $\text{Support}(\text{ขนมปัง}) = \frac{3}{5} = 0.6$ $\{ \text{Support} \} (\{ \text{ขนมปัง} \}) = \{ 3 \} \{ 5 \} = 0.6$
- $\text{Support}(\text{ขนมปัง}) = \frac{3}{5} = 0.6$

ลูกค้า	ขนมปัง	นม	ไข่
1	1	1	0
2	1	1	1
3	0	1	0
4	0	0	1
5	1	0	0

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

ใน Association Rule Mining คือค่ากำหนดขั้นต่ำที่กฎความสัมพันธ์ต้องมีค่า Confidence เท่ากับหรือมากกว่า เพื่อให้กฎนั้นถูกพิจารณาว่ามีความน่าเชื่อถือและเหมาะสมที่จะใช้งาน

เหตุผลของ Minimum Confidence

- การกำหนด Minimum Confidence เป็นตัวช่วยกรองกฎที่ไม่แข็งแรงพอ (Low Confidence Rules) ออก
- กฎที่มีค่า Confidence ต่ำอาจบ่งชี้ว่าความสัมพันธ์ระหว่างสินค้าหรือเหตุการณ์ไม่มีความน่าเชื่อถือในการใช้งานจริง

ตัวอย่าง กำหนด Minimum Confidence = 0.6 (60%)

หากเรามีกฎ $A \rightarrow BA$ ที่:

- Confidence = 0.5 (50%)

กฎนี้จะ ไม่ผ่านเกณฑ์ Minimum Confidence และถูกคัดออก

การใช้งาน Minimum Confidence

1. ในกระบวนการสร้างกฎ (Rule Generation):

1. ใช้ Minimum Confidence เพื่อคัดเลือกเฉพาะกฎที่มีค่า Confidence สูงพอ

2. ร่วมกับค่า Support:

1. ค่า Minimum Confidence มักใช้ร่วมกับ Minimum Support เพื่อให้ได้กฎที่มีทั้งความน่าเชื่อถือและความสำคัญทางสถิติ

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

บทที่ 5

ตัวอย่างใน Python (Apriori Algorithm)

```
from mlxtend.frequent_patterns import apriori, association_rules
import pandas as pd
# สร้าง DataFrame ตัวอย่าง
data = pd.DataFrame({
    "ขนมปัง": [1, 1, 0, 0, 0],
    "นม": [1, 1, 1, 0, 0],
    "ไข่": [0, 1, 0, 1, 0],
    "กาแฟ": [0, 0, 1, 1, 1],
    "น้ำผลไม้": [0, 0, 0, 1, 1]
})
# คำนวณ Frequent Itemsets
frequent_itemsets = apriori(data, min_support=0.2, use_colnames=True)
# สร้าง Association Rules และกำหนด Minimum Confidence
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.6)
# แสดงผลกฎที่ผ่านเกณฑ์
print(rules)
```

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน
Lec05_05_association_rules_minimum_confident.py

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

บทที่ 5

ตัวอย่างใน Python (Apriori Algorithm)

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน

Lec05_05_association_rules_minimum_confident.py

```
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> & C:/Users/.../ec05_association_rules_minimum_confident.py
C:\Users\User\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\site-packages\mlxtend\frequent_patterns\fpcommon.py:110: DeprecationWarning: DataFrames with non-bool types result in worse computational performance and their support might be discontinued in the future. Please use a DataFrame with bool type
  warnings.warn(

```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(นม)	(ขนมปัง)	0.6	0.4	0.4	0.666667	1.666667	0.16	1.8	1.000000
1	(ขนมปัง)	(นม)	0.4	0.6	0.4	1.000000	1.666667	0.16	inf	0.666667
2	(น้ำผลไม้)	(กาแฟ)	0.4	0.6	0.4	1.000000	1.666667	0.16	inf	0.666667
3	(กาแฟ)	(น้ำผลไม้)	0.6	0.4	0.4	0.666667	1.666667	0.16	1.8	1.000000
4	(นม, ไข่)	(ขนมปัง)	0.2	0.4	0.2	1.000000	2.500000	0.12	inf	0.750000
5	(ขนมปัง, ไข่)	(นม)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf	0.500000
6	(น้ำผลไม้, ไข่)	(กาแฟ)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf	0.500000
7	(ไข่, กาแฟ)	(น้ำผลไม้)	0.2	0.4	0.2	1.000000	2.500000	0.12	inf	0.750000

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

Deprecation Warning:

มีคำเตือนว่า DataFrame ของคุณไม่ได้ใช้ประเภท Boolean ซึ่งอาจทำให้ประสิทธิภาพลดลงในอนาคต ควรแปลงข้อมูลเป็น Boolean โดยใช้ `.astype(bool)` เช่น:

```
data = data.astype(bool)
```

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

แก้ไขโค้ดใหม่

```
from mlxtend.frequent_patterns import apriori, association_rules
import pandas as pd
# สร้าง DataFrame ตัวอย่าง
data = pd.DataFrame({
    "ขนมปัง": [1, 1, 0, 0, 0],
    "นม": [1, 1, 1, 0, 0],
    "ไข่": [0, 1, 0, 1, 0],
    "กาแฟ": [0, 0, 1, 1, 1],
    "น้ำผลไม้": [0, 0, 0, 1, 1]
})
data = data.astype(bool)
# คำนวณ Frequent Itemsets
frequent_itemsets = apriori(data, min_support=0.2, use_colnames=True)
# สร้าง Association Rules และกำหนด Minimum Confidence
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.6)
# แสดงผลกฎที่ผ่านเกณฑ์
print(rules)
```

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน
Lec05_05_association_rules_minimum_confident.py

Edited

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

บทที่ 5

แก้ไขล่าสุด

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน

Lec05_05_association_rules_minimum_confident.py

Edited

```

PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/AppServ/ec05_association_rules_minimum_confident.py

```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(ขนมปัง)	(นม)	0.4	0.6	0.4	1.000000	1.666667	0.16	inf	0.666667
1	(นม)	(ขนมปัง)	0.6	0.4	0.4	0.666667	1.666667	0.16	1.8	1.000000
2	(กาแฟ)	(น้ำผลไม้)	0.6	0.4	0.4	0.666667	1.666667	0.16	1.8	1.000000
3	(น้ำผลไม้)	(กาแฟ)	0.4	0.6	0.4	1.000000	1.666667	0.16	inf	0.666667
4	(ขนมปัง, ไข่)	(นม)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf	0.500000
5	(ไข่, นม)	(ขนมปัง)	0.2	0.4	0.2	1.000000	2.500000	0.12	inf	0.750000
6	(กาแฟ, ไข่)	(น้ำผลไม้)	0.2	0.4	0.2	1.000000	2.500000	0.12	inf	0.750000
7	(ไข่, น้ำผลไม้)	(กาแฟ)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf	0.500000

```

PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects>

```

ผลลัพธ์ที่แสดงเป็น Association Rules ที่ได้จากการใช้ Apriori Algorithm เพื่อค้นหาความสัมพันธ์ระหว่างรายการในข้อมูล โดยพิจารณาจากค่าต่าง ๆ เช่น Confidence, Lift, และอื่น ๆ เราจะวิเคราะห์ค่าต่าง ๆ ที่ปรากฏในตารางนี้:

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

ตัวอย่างใน Python (Apriori Algorithm)

Edited

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน
Lec05_05_association_rules_minimum_confident.py

1. คำอธิบายค่าต่าง ๆ ในผลลัพธ์

- antecedents: กลุ่มรายการที่อยู่ในด้านซ้ายของกฎ (เงื่อนไข)
- consequents: กลุ่มรายการที่อยู่ในด้านขวาของกฎ (ผลลัพธ์)
- antecedent support: ความถี่ของ antecedents ในข้อมูล
- consequent support: ความถี่ของ consequents ในข้อมูล
- support: ความถี่ที่ antecedents และ consequents ปรากฏร่วมกันในข้อมูล
- confidence: ความน่าเชื่อถือของกฎ (โอกาสที่ consequent จะเกิดขึ้นเมื่อ antecedent เกิดขึ้น)
- lift: ค่าที่แสดงถึงความแข็งแกร่งของความสัมพันธ์ระหว่าง antecedents และ consequents (ค่ามากกว่า 1 บ่งบอกว่ามีความสัมพันธ์เชิงบวก)
- leverage: ความแตกต่างระหว่างความถี่ที่เกิดขึ้นจริงกับความถี่ที่คาดหวัง
- conviction: โอกาสที่ antecedents จะเกิดขึ้นโดยไม่มี consequent
- zhangs_metric: ตัวชี้วัดความสัมพันธ์ระหว่าง antecedents และ consequents โดยไม่ขึ้นอยู่กับค่า support

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

ตัวอย่างใน Python (Apriori Algorithm)

2. การวิเคราะห์ผลลัพธ์ที่ได้

กฎตัวอย่าง

กฎที่ 1:

- Antecedents = (ขนมปัง)
- consequents = (นม)
- Confidence: 1.000 หมายความว่า หากลูกค้าซื้อ "ขนมปัง" โอกาสที่พวกเขาจะซื้อ "นม" คือ 100%
- Lift: 1.666 บ่งบอกว่าการซื้อ "ขนมปัง" มีผลในเชิงบวกกับการซื้อ "นม" โดยเพิ่มความน่าจะเป็นในการซื้อ "นม" 1.666 เท่าเมื่อเทียบกับการสุ่มซื้อ
- Conviction: INF (Infinity) แสดงว่าเมื่อ antecedent (ขนมปัง) เกิดขึ้น การไม่เกิด consequent (นม) เป็นไปไม่ได้เลย

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน

Lec05_05_association_rules_minimum_confident.py

Edited

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

บทที่ 5

ตัวอย่างใน Python (Apriori Algorithm)

Edited

กฎที่ 6:

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน
Lec05_05_association_rules_minimum_confident.py

- antecedents=(ไข่, นม)
- consequents = (ขนมปัง)
- Confidence: 1.000 หมายความว่า หากลูกค้าซื้อ "ไข่" และ "นม" พร้อมกัน โอกาสที่พวกเขาจะซื้อ "ขนมปัง" คือ 100%
- Lift: 2.500 การซื้อ "ไข่" และ "นม" พร้อมกันเพิ่มโอกาสในการซื้อ "ขนมปัง" มากขึ้นถึง 2.5 เท่าเมื่อเทียบกับการสุ่มซื้อ

3. การตีความผลรวม

กฎที่ Confidence = 1.0:

กฎเหล่านี้มีความน่าเชื่อถือ 100% ซึ่งเหมาะสมที่จะใช้งานจริง แต่ควรตรวจสอบว่า support มีค่ามากพอหรือไม่ (เพราะกฎที่ support ต่ำอาจไม่มีนัยสำคัญทางสถิติ)

ค่า Lift ที่สูงกว่า 1:

ทุกกฎในผลลัพธ์มีค่า Lift > 1 แสดงว่าความสัมพันธ์ระหว่าง antecedents และ consequents มีแนวโน้มเป็นบวกและเป็นกฎที่มีประโยชน์

Zhang's Metric:

ค่าที่สูงบ่งบอกว่ากฎเหล่านี้มีความสัมพันธ์ที่ชัดเจนระหว่าง antecedents และ consequents

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

ตัวอย่างใน Python (Apriori Algorithm)

4. ข้อสังเกต

• Deprecation Warning:

มีคำเตือนว่า DataFrame ของคุณไม่ได้ใช้ประเภท Boolean ซึ่งอาจทำให้ประสิทธิภาพลดลงในอนาคต ควรแปลงข้อมูลเป็น Boolean โดยใช้ `.astype(bool)` เช่น:

python

Copy code

```
data = data.astype(bool)
```

• Support ของบางกฎค่อนข้างต่ำ (0.2) อาจต้องพิจารณาว่ากฎเหล่านี้มีความสำคัญพอหรือไม่ในบริบทการใช้งาน

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน

Lec05_05_association_rules_minimum_confident.py

Edited

5.2

5.2.9 ค่าเชื่อมั่น (Confidence)

ตัวอย่างใน Python (Apriori Algorithm)

5. คำแนะนำ

- ปรับปรุงข้อมูลให้เป็น Boolean เพื่อลดค่าเตือนและเพิ่มประสิทธิภาพ
- วิเคราะห์กฎที่มี Support และ Confidence สูงร่วมกับค่า Lift และตัวชี้วัดอื่น ๆ เพื่อเลือกกฎที่มีประโยชน์ที่สุด

*** หากต้องการขยายการวิเคราะห์เพิ่มเติม เช่น การกรองกฎโดย Minimum Confidence หรือ Minimum Lift สามารถปรับพารามิเตอร์ใน `association_rules` ได้

File :

ตัวอย่างที่ 5.5 การหาค่า confidence ในภาษาไพธอน
Lec05_05_association_rules_minimum_confident.py

Edited

5.3

5.3.1 การคัดเลือกกฎความสัมพันธ์

บทที่ 5

ตัวอย่าง การคัดเลือกกฎความสัมพันธ์ ในภาษาไพรอน

การคัดเลือกกฎความสัมพันธ์ (Association Rules) เป็นกระบวนการที่สำคัญในการวิเคราะห์ข้อมูลเพื่อค้นหาความสัมพันธ์หรือรูปแบบที่เกิดขึ้นบ่อยในข้อมูล. กระบวนการนี้เน้นการค้นหารายการ (itemset) ที่มีความสัมพันธ์กันมากและสร้างกฎความสัมพันธ์ที่มีความน่าสนใจดังนี้ โดยขั้นตอนการคัดเลือกกฎความสัมพันธ์ มีดังนี้

1. การหา Itemsets ที่มีความถี่ (Frequent Itemsets): ในขั้นตอนแรก, ให้ค้นหา itemsets ที่มีความถี่มากกว่าหรือเท่ากับค่า support threshold ที่กำหนด. Itemset ที่มีความถี่นี้เรียกว่า Frequent Itemsets
2. การสร้างกฎความสัมพันธ์ (Association Rules): จาก Frequent Itemsets ที่ได้จากขั้นตอนที่ 1, สร้างกฎความสัมพันธ์ที่มีความน่าสนใจ. การสร้างกฎนี้จะเกี่ยวข้องกับการกำหนดค่า confidence threshold เพื่อคัดเลือกกฎที่มีความน่าสนใจ
3. การคัดเลือกกฎความสัมพันธ์: ในขั้นตอนนี้, คัดเลือกกฎความสัมพันธ์ที่ตรงตามเกณฑ์ที่กำหนด, เช่นค่า confidence มากกว่าหรือเท่ากับค่าที่กำหนด

5.3

5.3.1 การคัดเลือกกฎความสัมพันธ์

บทที่ 5

ตัวอย่าง การคัดเลือกกฎความสัมพันธ์ ในภาษาไพรอน

การคัดเลือกกฎความสัมพันธ์มีได้หลายวิธี แต่โดยทั่วไปแล้วมีหลักการดังนี้:

- ความถี่ (Support): คัดเลือกกฎที่มีความถี่ขั้นต่ำที่กำหนด
- ความน่าสนใจ (Interest): คัดเลือกกฎที่มีความน่าสนใจมาก, โดยคำนวณจากสูตรที่มีการเชื่อมโยงระหว่างความถี่ของ Itemsets และความถี่ของ Itemsets ที่ส่งผลให้กฎ
- ความคุ้มค่า (Lift): คัดเลือกกฎที่มีความคุ้มค่ามาก, โดยคำนวณจากสูตรที่มีการเชื่อมโยงระหว่างความถี่ของ Itemsets และความถี่ของ Itemsets ที่ส่งผลให้กฎ และความถี่ของ Itemsets ต้นฉบับ

การคัดเลือกกฎความสัมพันธ์จะช่วยลดขนาดของชุดกฎและช่วยให้เหลือกฎที่มีความน่าสนใจและสามารถนำไปใช้ประโยชน์ได้ในการตัดสินใจหรือวิเคราะห์ข้อมูล

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 1 : ห้างวอลมาร์ต (Walmart)

ในที่นี้จะขอยกตัวอย่างมาให้ดู 2 ตัวอย่างก่อน โดยตัวอย่างแรกเป็นตัวอย่างคลาสสิก นั่นก็คือการที่ห้างวอลมาร์ต (Walmart) ได้ทำการค้นพบพฤติกรรมการซื้อสินค้าของลูกค้าที่เป็นเพศชายว่า ในช่วงเย็นของวันศุกร์ลูกค้ากลุ่มนี้มักจะมาซื้อสินค้าสองอย่างควบคู่กันไป นั่นก็คือ “เบียร์และผ้าอ้อม” โดยจากการวิเคราะห์เจาะลึกลงไปก็พบเหตุผลว่าการที่สินค้าสองอย่างนี้มีการซื้อร่วมกันบ่อย ๆ เพราะว่า พ่อบ้านส่วนใหญ่มักจะซื้อเบียร์ไปดื่มในช่วงสุดสัปดาห์และเกิดคิดถึงลูกน้อยของตัวเองขึ้นมาจึงซื้อผ้าอ้อมติดไม้ติดมือไปด้วย (ถ้าเป็นบ้านเราอาจจะพบว่าพฤติกรรมแบบนี้เกิดขึ้นเพราะพ่อบ้านถูกภรรยาใช้ให้มาซื้อผ้าอ้อมให้ลูกแล้วอยากดื่มเบียร์ด้วยหรือเปล่าก็ไม่รู้) หลังจากที่ห้างวอลมาร์ตรู้ถึงพฤติกรรมแบบนี้ทางห้างก็สามารถที่จะจัดวางสินค้าสองชนิดนี้ให้สามารถค้นหาได้ง่ายๆ หรือมองเห็นได้ง่ายเพื่อเพิ่มโอกาสที่ลูกค้าจะได้ซื้อติดไม้ติดมือกันไปด้วย

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 1 : ห้างวอลมาร์ต (Walmart)

1. การจัดวางสินค้าในร้านค้า

ในกรณีของ Walmart ที่ค้นพบว่า “เบียร์และผ้าอ้อม” มักถูกซื้อคู่กันในช่วงเย็นของวันศุกร์ ส่งผลให้ร้านสามารถปรับกลยุทธ์ เช่น:

- วางสินค้าสองชนิดนี้ใกล้กันเพื่อความสะดวกในการเลือกซื้อ
- สร้างโปรโมชั่นหรือส่วนลดสำหรับสินค้าที่เกี่ยวข้องเพื่อกระตุ้นยอดขาย

2. การแนะนำสินค้าในระบบ E-commerce

ตัวอย่าง: Amazon หรือ Shopee ใช้ Association Rule ในการแนะนำสินค้า เช่น:

- หากลูกค้าซื้อ “โทรทัศน์มือถือ” ระบบอาจแนะนำ “เคสมือถือ” หรือ “ฟิล์มกันรอย” เพราะพบว่าลูกค้าส่วนใหญ่มักซื้อสินค้ากลุ่มนี้ร่วมกัน
- การแสดงคำแนะนำ “ลูกค้าที่ซื้อสินค้านี้มักจะซื้อสิ่งนี้ด้วย” เป็นผลลัพธ์ของการวิเคราะห์กฎความสัมพันธ์

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 1 : ห้างวอลมาร์ต (Walmart)

ตัวอย่างข้อมูล

ข้อมูลตัวอย่างมี 10 รายการ (ลูกค้า 10 คน) โดย 1 หมายถึงซื้อ และ 0 หมายถึงไม่ซื้อ:

ลูกค้า	เบียร์	ผ้าอ้อม	นม	ขนมปัง	ขนมขบเคี้ยว
1	1	1	0	1	0
2	0	1	1	0	1
3	1	0	1	1	0
4	1	1	0	1	1
5	0	1	1	0	1
6	1	0	1	1	1
7	1	1	0	0	0
8	0	1	1	1	1
9	0	0	1	1	1
10	1	1	0	1	1

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

ตัวอย่างโค้ดภาษา Python ที่ใช้ในการวิเคราะห์ A

File :

ตัวอย่างที่ 5.6 กรณีศึกษาห้าง Walwart
Lec05_06_ association_rule_analysis.py

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
# สร้าง DataFrame ตัวอย่าง (ข้อมูล 10 รายการ)

data = {
    'เบียร์': [1, 0, 1, 1, 0, 1, 1, 0, 0, 1],
    'ผ้าอ้อม': [1, 1, 0, 1, 1, 0, 1, 1, 0, 1],
    'นม': [0, 1, 1, 0, 1, 1, 0, 1, 1, 0],
    'ขนมปัง': [1, 0, 1, 1, 0, 1, 0, 1, 1, 1],
    'ขนมขบเคี้ยว': [0, 1, 0, 1, 1, 1, 0, 1, 1, 1]
}
```

Part01

```
df = pd.DataFrame(data)
# แปลงข้อมูลให้เป็นประเภท bool

df = df.astype('bool')

# สร้าง Frequent Itemsets ด้วย Apriori

frequent_itemsets = apriori(df, min_support=0.3,
                             use_colnames=True)
# สร้าง Association Rules

rules = association_rules(frequent_itemsets,
                          metric="confidence", min_threshold=0.7)
# แสดงผลลัพธ์

print("Frequent Itemsets:")
print(frequent_itemsets)

print("\nAssociation Rules:")
print(rules)
```

Part02

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 1 : ห้างวอลมาร์ต (Walmart)

Frequent Itemsets:

	support	itemsets
0	0.6	(เบียร์)
1	0.7	(ผ้าอ้อม)
2	0.6	(นม)
3	0.7	(ขนมปัง)
4	0.7	(ขนมขบเคี้ยว)
5	0.4	(ผ้าอ้อม, เบียร์)
6	0.5	(เบียร์, ขนมปัง)
7	0.3	(เบียร์, ขนมขบเคี้ยว)
8	0.3	(ผ้าอ้อม, นม)

Frequent Itemsets:

9	0.4	(ผ้าอ้อม, ขนมปัง)
10	0.5	(ผ้าอ้อม, ขนมขบเคี้ยว)
11	0.4	(ขนมปัง, นม)
12	0.5	(ขนมขบเคี้ยว, นม)
13	0.5	(ขนมขบเคี้ยว, ขนมปัง)
14	0.3	(ผ้าอ้อม, เบียร์, ขนมปัง)
15	0.3	(เบียร์, ขนมปัง, ขนมขบเคี้ยว)
16	0.3	(ผ้าอ้อม, ขนมขบเคี้ยว, นม)
17	0.3	(ผ้าอ้อม, ขนมขบเคี้ยว, ขนมปัง)
18	0.3	(ขนมปัง, นม, ขนมขบเคี้ยว)

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 1 : ห้างวอลมาร์ต (Walmart)

Association Rules:

	antecedents		consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	
zhangs_metric											
0	(เบียร์)	(ขนมปัง)		0.6	0.7	0.5	0.833333	1.190476	0.08	1.80	0.400000
1	(ขนมปัง)	(เบียร์)		0.7	0.6	0.5	0.714286	1.190476	0.08	1.40	0.533333
2	(ผ้าอ้อม)	(ขนมขบเคี้ยว)		0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667
3	(ขนมขบเคี้ยว)	(ผ้าอ้อม)		0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667
4	(ขนมขบเคี้ยว)	(นม)		0.7	0.6	0.5	0.714286	1.190476	0.08	1.40	0.533333
5	(นม)	(ขนมขบเคี้ยว)		0.6	0.7	0.5	0.833333	1.190476	0.08	1.80	0.400000
6	(ขนมขบเคี้ยว)	(ขนมปัง)		0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667
7	(ขนมปัง)	(ขนมขบเคี้ยว)		0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 1 : ห้างวอลมาร์ต (Walmart)

Association Rules:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	
zhangs_metric										
8	(ผ้าอ้อม, เบียร์)	(ขนมปัง)	0.4	0.7	0.3	0.750000	1.071429	0.02	1.20	0.111111
9	(ผ้าอ้อม, ขนมปัง)	(เบียร์)	0.4	0.6	0.3	0.750000	1.250000	0.06	1.60	0.333333
10	(เบียร์, ขนมขบเคี้ยว)	(ขนมปัง)	0.3	0.7	0.3	1.000000	1.428571	0.09	inf	0.428571
11	(ผ้าอ้อม, นม)	(ขนมขบเคี้ยว)	0.3	0.7	0.3	1.000000	1.428571	0.09	inf	0.428571
12	(ผ้าอ้อม, ขนมปัง)	(ขนมขบเคี้ยว)	0.4	0.7	0.3	0.750000	1.071429	0.02	1.20	0.111111
13	(ขนมปัง, นม)	(ขนมขบเคี้ยว)	0.4	0.7	0.3	0.750000	1.071429	0.02	1.20	0.111111

5.4

5.4.1 ตัวอย่างการประยุกต์ใช้งาน Association rule

กรณีศึกษาที่ 1 : ห้างวอลมาร์ต (Walmart)

```
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining> python ec05_association_rule_analysis.py
Frequent Itemsets:
support      itemsets
0      0.6      (เบียร์)
1      0.7      (ผ้าอ้อม)
2      0.6      (นม)
3      0.7      (ขนมปัง)
4      0.7      (ขนมปัง เคี้ยว)
5      0.4      (ผ้าอ้อม, เบียร์)
6      0.5      (เบียร์, ขนมปัง)
7      0.3      (เบียร์, ขนมปัง เคี้ยว)
8      0.3      (ผ้าอ้อม, นม)
9      0.4      (ผ้าอ้อม, ขนมปัง)
10     0.5      (ผ้าอ้อม, ขนมปัง เคี้ยว)
11     0.4      (ขนมปัง, นม)
12     0.5      (ขนมปัง เคี้ยว, นม)
13     0.5      (ขนมปัง เคี้ยว, ขนมปัง)
14     0.3      (ผ้าอ้อม, เบียร์, ขนมปัง)
15     0.3      (เบียร์, ขนมปัง, ขนมปัง เคี้ยว)
16     0.3      (ผ้าอ้อม, ขนมปัง เคี้ยว, นม)
17     0.3      (ผ้าอ้อม, ขนมปัง เคี้ยว, ขนมปัง)
18     0.3      (ขนมปัง, นม, ขนมปัง เคี้ยว)
```

Association Rules:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(เบียร์)	(ขนมปัง)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.80	0.400000
1	(ขนมปัง)	(เบียร์)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.40	0.533333
2	(ผ้าอ้อม)	(ขนมปัง เคี้ยว)	0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667
3	(ขนมปัง เคี้ยว)	(ผ้าอ้อม)	0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667
4	(ขนมปัง เคี้ยว)	(นม)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.40	0.533333
5	(นม)	(ขนมปัง เคี้ยว)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.80	0.400000
6	(ขนมปัง เคี้ยว)	(ขนมปัง)	0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667
7	(ขนมปัง)	(ขนมปัง เคี้ยว)	0.7	0.7	0.5	0.714286	1.020408	0.01	1.05	0.066667
8	(ผ้าอ้อม, เบียร์)	(ขนมปัง)	0.4	0.7	0.3	0.750000	1.071429	0.02	1.20	0.111111
9	(ผ้าอ้อม, ขนมปัง)	(เบียร์)	0.4	0.6	0.3	0.750000	1.250000	0.06	1.60	0.333333
10	(เบียร์, ขนมปัง เคี้ยว)	(ขนมปัง)	0.3	0.7	0.3	1.000000	1.428571	0.09	inf	0.428571
11	(ผ้าอ้อม, นม)	(ขนมปัง เคี้ยว)	0.3	0.7	0.3	1.000000	1.428571	0.09	inf	0.428571
12	(ผ้าอ้อม, ขนมปัง)	(ขนมปัง เคี้ยว)	0.4	0.7	0.3	0.750000	1.071429	0.02	1.20	0.111111
13	(ขนมปัง, นม)	(ขนมปัง เคี้ยว)	0.4	0.7	0.3	0.750000	1.071429	0.02	1.20	0.111111

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

ผลลัพธ์

การปรับใช้

- ตัวอย่างนี้เหมาะสำหรับการเข้าใจพฤติกรรมของลูกค้าในร้านค้า เช่น การจัดโปรโมชั่นสินค้า หรือการวางสินค้าที่มักจะถูกซื้อพร้อมกันไว้ใกล้กัน
- สามารถเพิ่มข้อมูลจริงได้จากการเก็บข้อมูลในระบบขายหน้าร้าน (POS)

5.4

5.4.2 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 2 : ห้างทาร์เก็ต (Target)

ตัวอย่างที่ 2 ก็ยังคงมาจากห้างสรรพสินค้าเหมือนกัน นั่นคือห้างทาร์เก็ต (Target) ห้างทาร์เก็ตนี้เป็นห้างที่เกิดขึ้นมาทีหลังทำให้การจะแข่งขันกับห้างวอลมาร์ตที่มีอยู่ก่อนแล้วก็คงไม่ใช่เรื่องง่าย ดังนั้นทางห้างจึงพยายามหาวิธีที่จะดึงดูดให้ลูกค้ามาซื้อสินค้ากับทางห้างให้มากขึ้นและรักษฐานลูกค้าที่มีอยู่ให้เชื่อใจและอยากกลับมาซื้อสินค้าที่ห้างของตนเองให้ได้มากที่สุด จากการวิจัยทางการตลาดของห้างทาร์เก็ตพบว่า เมื่อครอบครัวมีสมาชิกใหม่เกิดขึ้นคนในครอบครัวก็จะเริ่มมีการจับจ่ายใช้สอยมากขึ้นเพื่อรองรับการขยายขนาดของครอบครัว ดังนั้นเมื่อทราบเช่นนี้แล้วทางห้างทาร์เก็ตจึงได้ทำการวิเคราะห์พฤติกรรมของลูกค้าผู้หญิงที่มาซื้อสินค้าและพบว่าเมื่อลูกค้าเหล่านี้เริ่มตั้งครรภ์ ลูกค้าจะมีพฤติกรรมการซื้อสินค้าที่เปลี่ยนไป เช่น เริ่มมีการซื้อวิตามินบำรุงมากขึ้น เปลี่ยนไปกินอาหารที่มีประโยชน์ หรือแม้กระทั่งซื้อตู้เสื้อผ้าเพิ่ม จากรูปแบบพฤติกรรมลักษณะนี้ทำให้ทางห้างสามารถส่งโปรโมชั่นเกี่ยวกับการตั้งครรภ์หรือสินค้าสำหรับเด็กให้กับลูกค้ากลุ่มนี้ได้ นอกจากนี้ห้างทาร์เก็ตยังมีความมั่นใจว่าถ้าลูกค้าเชื่อใจที่จะซื้อสินค้าให้กับบุตรที่เกิดขึ้นใหม่แล้วลูกค้าเหล่านี้ก็จะเชื่อใจซื้อสินค้าชนิดอื่น ๆ ของทางห้างไปอีกเรื่อย ๆ (รายละเอียดเพิ่มเติมดูได้จากหัวข้อ “เอกสารอ้างอิง” ช่วงท้ายของบทนี้ครับ)

5.4

5.4.2 ตัวอย่างการประยุกต์ใช้งาน Association rule

กรณีศึกษาที่ 2 : ห้างทาร์เก็ต (Target)

จากตัวอย่างกรณีศึกษาของห้าง Target ที่ได้วิเคราะห์พฤติกรรมการณ์การซื้อสินค้าของลูกค้าผู้หนึ่ง โดยเฉพาะในช่วงที่ลูกค้าตั้งครรภ์ การนำ Association Rule มาประยุกต์ใช้งานสามารถช่วยให้ห้างเข้าใจพฤติกรรมการณ์การซื้อสินค้าของลูกค้าและส่งโปรโมชั่นที่เหมาะสมได้ ซึ่งสามารถเขียนโค้ดตัวอย่างใน Python เพื่อจำลองกระบวนการนี้ได้ดังนี้

5.4

5.4.2 ตัวอย่าง

File :

บทที่ 5

ตัวอย่างที่ 5.7 การหาค่า confidence ในภาษาไพธอน

Lec05_07_association_rule_analysis_02.py

กรณีศึกษาที่ 2 : ห้างการค้าเกิด (Target)

Part01

```
import pandas as pd

from mlxtend.frequent_patterns import apriori, association_rules
# สร้างข้อมูลจำลอง

data = {
    'Prenatal_Vitamins': [1, 0, 1, 1, 0, 0, 1, 1, 0, 1],
    'Healthy_Foods': [1, 1, 0, 1, 1, 1, 0, 1, 0, 1],
    'Baby_Clothes': [0, 1, 1, 1, 1, 0, 1, 1, 1, 0],
    'Wardrobe': [1, 0, 1, 1, 0, 1, 0, 0, 1, 1]
}

# สร้าง DataFrame

df = pd.DataFrame(data)
# แปลงข้อมูลเป็นประเภท Boolean

df = df.astype('bool')
```

Part02

```
# ใช้ Apriori เพื่อหา Frequent Itemsets

frequent_itemsets = apriori(df, min_support=0.3,
                             use_colnames=True)
# สร้าง Association Rules

rules = association_rules(frequent_itemsets,
                          metric="confidence", min_threshold=0.7)
# แสดงผลลัพธ์

print("Frequent Itemsets:")
print(frequent_itemsets)

print("\nAssociation Rules:")
print(rules)
```

5.4

5.4.2 ตัวอย่าง

File :

ตัวอย่างที่ 5.7 การหาค่า confidence ในภาษาไพธอน

Lec05_07_association_rule_analysis_02.py

กรณีศึกษาที่ 2 : ห้างค้าเกิด (Target)

```
PS C:\AppServ\www\siam2dev_net\E_Learning\DataMining\DM_Projects> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/AppServ/www/siam2dev_net/E_Learning/DataMining/DM_Projects/Lec05_07_association_rule_analysis_02.py
```

Frequent Itemsets:

support	itemsets
0 0.6	(Prenatal_Vitamins)
1 0.7	(Healthy_Foods)
2 0.7	(Baby_Clothes)
3 0.6	(Wardrobe)
4 0.4	(Prenatal_Vitamins, Healthy_Foods)
5 0.4	(Baby_Clothes, Prenatal_Vitamins)
6 0.4	(Wardrobe, Prenatal_Vitamins)
7 0.4	(Baby_Clothes, Healthy_Foods)
8 0.4	(Wardrobe, Healthy_Foods)
9 0.3	(Baby_Clothes, Wardrobe)
10 0.3	(Wardrobe, Prenatal_Vitamins, Healthy_Foods)

Association Rules:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(Wardrobe, Prenatal_Vitamins)	(Healthy_Foods)	0.4	0.7	0.3	0.75	1.071429	0.02	1.2	0.111111
1	(Wardrobe, Healthy_Foods)	(Prenatal_Vitamins)	0.4	0.6	0.3	0.75	1.250000	0.06	1.6	0.333333
2	(Prenatal_Vitamins, Healthy_Foods)	(Wardrobe)	0.4	0.6	0.3	0.75	1.250000	0.06	1.6	0.333333

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

กฎที่ควรเลือก

กฎที่ 2 (Wardrobe, Healthy_Foods) → (Prenatal_Vitamins)

กฎที่ 3 (Prenatal_Vitamins, Healthy_Foods) → (Wardrobe)

เนื่องจากทั้งสองกฎมีค่า Lift, Leverage, และ Zhang's Metric สูงกว่า ทำให้มีความสัมพันธ์เชิงบวกที่แข็งแกร่งกว่ากฎที่ 1

5.4

5.4.2 ตัวอย่างการประยุกต์ใช้งาน Association rule

กรณีศึกษาที่ 2 : ห้างค้าเกิด (Target)

File :

ตัวอย่างที่ 5.7 การหาค่า confidence ในภาษาไพธอน
Lec05_07_association_rule_analysis_02.py

ตัวชี้วัดที่สำคัญ

Confidence: แสดงโอกาสที่เหตุการณ์ใน Consequent จะเกิดเมื่อ Antecedent เกิด

ทุกกฎมี Confidence = 0.75 (เหมือนกัน)

Lift: แสดงระดับความสัมพันธ์เชิงบวก (ค่ามากกว่า 1 คือมีความสัมพันธ์)

กฎที่ 2 และกฎที่ 3 มีค่า Lift = 1.25 ซึ่งสูงกว่ากฎที่ 1 (Lift = 1.071)

Leverage: แสดงการเพิ่มโอกาสของการเกิด Consequent เมื่อ Antecedent เกิด

กฎที่ 2 และกฎที่ 3 มีค่า Leverage = 0.06 ซึ่งสูงกว่ากฎที่ 1 (Leverage = 0.02)

Zhang's Metric: แสดงระดับความสัมพันธ์ระหว่าง Antecedents และ Consequents

กฎที่ 2 และกฎที่ 3 มีค่า Zhang's Metric = 0.333 ซึ่งสูงกว่ากฎที่ 1 (Zhang's Metric = 0.111)

5.4

5.4.2 ตัวอย่างการประยุกต์ใช้งาน Association rule

บทที่ 5

กรณีศึกษาที่ 2 : ห้างค้าเกิด (Target)

File :

ตัวอย่างที่ 5.7 การหาค่า confidence ในภาษาไพธอน
Lec05_07_association_rule_analysis_02.py

อธิบายได้ด:

1. สร้างข้อมูลจำลอง:

- ข้อมูลจำลองประกอบด้วยสินค้า เช่น วิตามินบำรุงครรภ์ (Prenatal_Vitamins), อาหารเพื่อสุขภาพ (Healthy_Foods), เสื้อผ้าเด็ก (Baby_Clothes), และตู้เสื้อผ้า (Wardrobe)
- ใช้ 1 หมายถึงซื้อสินค้า และ 0 หมายถึงไม่ได้ซื้อสินค้า

2. ใช้ Apriori Algorithm:

- ฟังก์ชัน apriori ใช้ในการค้นหารูปแบบการซื้อสินค้าที่เกิดขึ้นบ่อย (Frequent Itemsets) โดยกำหนดค่า min_support เป็น 0.3 (ซื้อร่วมกัน 30% ของรายการทั้งหมด)

3. สร้าง Association Rules:

- ใช้ฟังก์ชัน association_rules เพื่อสร้างกฎความสัมพันธ์โดยตั้งค่า confidence ไว้ที่ 0.7 (ความมั่นใจ 70%)

4. ผลลัพธ์:

- แสดงรายการสินค้าที่มีการซื้อร่วมกันบ่อย (Frequent Itemsets)
- แสดงกฎความสัมพันธ์ เช่น หากลูกค้าซื้อ "วิตามินบำรุงครรภ์" แล้วมีโอกาสสูงที่จะซื้อ "อาหารเพื่อสุขภาพ" ด้วย

5.5

แบบฝึกหัด (Assignment)

บทที่ 5

ตัวอย่างธุรกิจ

• ระบบร้านขายโทรศัพท์

กำหนดข้อมูลนี้ การซื้อสินค้าจากร้านขายโทรศัพท์มือถือ โดยในแต่ละรายการจะแสดงสินค้าที่ลูกค้าซื้อในแต่ละครั้ง เพื่อใช้สำหรับการวิเคราะห์ Association Rules

Transaction ID	Smartphone	Phone_Case	Screen_Protector	Charger	Earphones	Power_Bank	Smartwatch	Tablet
1	1	1	1	1	0	1	0	0
2	1	1	0	1	1	0	1	0
3	1	0	1	1	0	1	0	0
4	0	1	1	0	1	0	1	0
5	1	0	1	1	0	0	1	0
6	1	1	1	0	1	1	0	1
7	0	1	0	1	0	1	1	0
8	1	1	1	1	1	0	1	0
9	1	1	1	1	0	1	1	0
10	0	1	1	0	1	0	0	1



5.5

แบบฝึกหัด (Assignment)

บทที่ 5

ตัวอย่างธุรกิจ

• ระบบร้านขายโทรศัพท์

โจทย์แบบฝึกหัด

การเตรียมข้อมูล:

1. ให้นักเรียนสร้าง DataFrame จากข้อมูลตัวอย่างที่ให้ไว้ หรือสร้างข้อมูลใหม่ที่มี Transaction ID และสินค้าอย่างน้อย 10 รายการ
2. การวิเคราะห์ Itemsets: ใช้ Apriori Algorithm เพื่อค้นหา Frequent Itemsets ที่มี support ไม่น้อยกว่า 30%

คำถาม:

- สินค้าใดบ้างที่มีจะถูกซื้อพร้อมกันบ่อยที่สุด?
3. การสร้างกฎความสัมพันธ์ (Association Rules):
 - ใช้ Frequent Itemsets ที่ได้มาเพื่อสร้าง Association Rules โดยกำหนด:
 - Confidence ไม่น้อยกว่า 70%
 - Lift มากกว่า 1

คำถาม:

- กฎความสัมพันธ์ใดที่น่าสนใจที่สุด?
- กฎใดที่สามารถนำไปใช้ในเชิงกลยุทธ์ได้?



5.5

แบบฝึกหัด (Assignment)

บทที่ 5

ตัวอย่างธุรกิจ

• ระบบร้านขายโทรศัพท์

4. การวิเคราะห์ผลลัพธ์:

อธิบายผลลัพธ์ที่ได้จากกฎความสัมพันธ์

คำถาม:

- คุณพบรูปแบบพฤติกรรมใดบ้างจากข้อมูล?
- คุณจะแนะนำให้ร้านขายโทรศัพท์ทำโปรโมชันอย่างไร เพื่อเพิ่มยอดขายจากข้อมูลที่ได้?

5. การปรับเปลี่ยนเงื่อนไข:

ทดลองปรับเปลี่ยนค่าขั้นต่ำของ support และ confidence แล้วเปรียบเทียบผลลัพธ์ที่ได้

คำถาม:

- การปรับค่าดังกล่าวส่งผลอย่างไรต่อจำนวน Frequent Itemsets และ Association Rules?



5.5

แบบฝึกหัด (Assignment)

บทที่ 5

ตัวอย่างธุรกิจ

• ระบบร้านขายโทรศัพท์

ตัวอย่างคำตอบที่คาดหวัง

1. Frequent Itemsets:

- {Smartphone, Phone_Case} (support = 0.5)
- {Smartphone, Screen_Protector} (support = 0.4)

2. Association Rules:

- หากลูกค้าซื้อ Smartphone แล้วมีโอกาส 75% ที่จะซื้อ Phone_Case (Confidence = 75%, Lift = 1.5)

3. ข้อเสนอแนะในเชิงกลยุทธ์:

- เสนอโปรโมชั่นซื้อ Smartphone แถม Phone_Case หรือส่วนลดพิเศษสำหรับการซื้อร่วมกัน
- สร้างแคมเปญส่งเสริมการขายสินค้าที่ลูกค้าซื้อบ่อย เช่น Smartwatch หรือ Tablet



5.5 แบบฝึกหัด (Assignment)

1. กำหนดธุรกิจ ทำความเข้าใจธุรกิจ ตั้งคำถามที่จะต้องการวิเคราะห์ก่อน
2. เตรียมข้อมูล ใส่ไฟล์ Excel 100 รายการ
3. ทำ Select data / Data Cleaning / Data Transformation
4. ใช้โปรแกรม Rapid Miner Studio สร้าง Model
5. แปลผล สรุปผล

ทำเป็นรายงานส่งใส่ ms-word ทุกหัวข้อ ส่งมาที่อีเมล siam2dev@Hotmail.com

1. ระบุชื่อสมาชิก รหัส รุ่น/หมู่
2. ระบุชื่อวิชา
3. ส่งก่อน 19.00 น.

6. การนำแบบจำลองที่ได้ไปใช้งาน (Deployment)

- วางแผนการตลาด ถ้ารู้ว่าสินค้าใด สั่งซื้อในปริมาณที่มาก ก็ผลิตสินค้านั้นให้เพียงพอกับความต้องการ
- จัดกิจกรรมส่งเสริมการตลาด ก็อาจจะมีการจัด สินค้าคู่กัน หรือให้ส่วนลด เมื่อซื้อสินค้าคู่กัน จัดโปรโมชั่นในเว็บไซต์ให้ตรงกับความต้องการของลูกค้า

บทที่ 3 กฎความสัมพันธ์ (Association Rules)

- กฎความสัมพันธ์ (Association Rules) เป็นกระบวนการหนึ่งในการทำ Data Mining ที่ได้รับความนิยมมาก โดยจะใช้ Association Rules ในการหาความสัมพันธ์ของข้อมูลสองชุดหรือมากกว่าสองชุดขึ้นไปภายในกลุ่มข้อมูลที่มีขนาดใหญ่ ในการหาความสัมพันธ์นั้นจะมีขั้นตอนวิธีการหาหลายวิธีด้วยกัน แต่ขั้นตอนวิธีที่เป็นที่รู้จักและใช้อย่างแพร่หลายคือ ขั้นตอนวิธี Apriori
- ตัวอย่างหนึ่งของ Association Rules ที่ใช้กันก็คือ Market Basket Analysis ที่ใช้ในการหาความสัมพันธ์ของสินค้าที่ลูกค้ามักจะซื้อพร้อมกัน เพื่อใช้ในการจัดรายการส่งเสริมการขาย

บทที่ 3 กฎความสัมพันธ์ (Association Rules)

- เป็นเทคนิคหนึ่งของ Data Mining คือการค้นหากฎความสัมพันธ์ของข้อมูลจากข้อมูลขนาดใหญ่ที่มีอยู่เพื่อนำไปหารูปแบบที่เกิดขึ้นบ่อยๆ (frequent pattern) และใช้ในการวิเคราะห์ความสัมพันธ์หรือทำนายปรากฏการณ์ต่างๆ
- ฐานข้อมูลที่ใช้ในการทำเหมืองความสัมพันธ์ (Association Mining) มักเป็นฐานข้อมูลประเภท Transaction Database
- ผลลัพธ์ที่ได้เป็นกฎความสัมพันธ์ (Association Rule) สามารถเขียนได้ในรูปเซตของรายการที่เป็นเหตุ ไปสู่เซตของรายการที่เป็นผล ซึ่งมีรากฐานมาจากการวิเคราะห์ตะกร้าตลาด (Market Basket Analysis) เช่น ลูกค้าที่ซื้อผ้าอ้อมส่วนใหญ่จะซื้อเบียร์ด้วย
- ข้อมูลที่นำมาใช้จะอยู่ในรูปแบบ Nominal หรือ Ordinal เท่านั้น

การนำเทคนิคไปประยุกต์ใช้กับงานจริง

- ระบบแนะนำหนังสือให้กับลูกค้าแบบอัตโนมัติของ Amazon หมายถึงข้อมูลการสั่งซื้อทั้งหมด จะถูกนำมาประมวลผลเพื่อหาความสัมพันธ์ของข้อมูล เช่น ลูกค้าที่ซื้อหนังสือเล่มหนึ่ง ๆ มักจะซื้อหนังสือเล่มใดพร้อมกันด้วยเสมอ ความสัมพันธ์ที่ได้จากกระบวนการนี้สามารถนำไปใช้คาดเดาได้ว่าควรแนะนำหนังสือเล่มใดเพิ่มเติมให้กับลูกค้า

บทที่ 3 กฎความสัมพันธ์ (Association Rules)

- ขั้นตอนการหากฎความสัมพันธ์ (Association Rules)

กฎความสัมพันธ์ (Association Rules)

- การวิเคราะห์ตะกร้าตลาด เป็นรูปแบบที่ใช้เพื่อหากลุ่มสิ่งของที่น่าจะปรากฏร่วมกันใน transaction หนึ่ง ๆ ซึ่งมักเป็น transaction ณ จุดขาย ผลลัพธ์ที่ได้สามารถแสดงได้ด้วยกฎ ซึ่งบอกความเป็นไปได้ของการซื้อสินค้าต่าง ๆ ร่วมกัน
- การวิเคราะห์ตะกร้าตลาด มีบทบาทสำคัญต่ออุตสาหกรรมการค้าปลีก (retail industry) ซึ่งใช้สารสนเทศ ศึกษาพฤติกรรมของลูกค้า
 - จัดพื้นที่ร้านค้า
 - จัดวางสินค้าร่วมกันเพื่อส่งเสริมการขาย
 - การวางแผนการส่งเสริมการขายและตั้งราคาผลิตภัณฑ์

อัลกอริทึม Apriori

- อัลกอริทึม Apriori ถือเป็นอัลกอริทึมที่นิยมใช้ในการหากฎความสัมพันธ์ของข้อมูล แต่ถ้าฐานข้อมูลมีการเพิ่มข้อมูลเข้ามา หรือ เกิดมีการเปลี่ยนแปลงข้อมูล อัลกอริทึม Apriori จะต้องนำข้อมูลทั้งหมดมารวมกันก่อน แล้วจึงจะสามารถนำข้อมูลทั้งหมดไปค้นหา กฎความสัมพันธ์ใหม่ทั้งหมด โดยไม่สามารถนำกฎความสัมพันธ์ที่หาได้จากกลุ่มข้อมูลเก่าก่อนหน้ามาใช้ให้เกิดประโยชน์ได้ ทำให้เสียเวลาในการทำงานเพื่อค้นหากฎความสัมพันธ์ใหม่ทั้งหมด

ขั้นตอนการหากฎความสัมพันธ์ (Association Rules)

ในการหากฎความสัมพันธ์นี้มีอยู่หลายวิธี แต่สำหรับบทความนี้จะแสดงการหากฎความสัมพันธ์ด้วยวิธี **Apriori** ซึ่งจะมีอยู่ 2 ขั้นตอนใหญ่ ๆ คือ

1. **การหา frequent itemset** เป็นการหารูปแบบของข้อมูลที่เกิดขึ้นร่วมกันบ่อยๆ ในฐานข้อมูล หรือ มากกว่าค่า minimum support ที่ผู้ใช้กำหนด ในขั้นตอนนี้จะแบ่งได้อีกเป็น 2 ขั้นตอนย่อย คือ
 1. การสร้างรูปแบบของ itemset (join) จะใช้รูปแบบของ itemset ที่มีค่ามากกว่า minimum support มาทำการสร้างรูปแบบของ itemset ที่มีขนาดยาวมากขึ้นทีละหนึ่งขั้นไปเรื่อยๆ
 2. การนับค่า support (count) หลังจากการสร้างรูปแบบของ itemset ได้แล้ว ขั้นถัดมาจะทำการคำนวณค่า support ที่เกิดขึ้น โดยที่ support คือจำนวนเปอร์เซ็นต์ที่พบ itemset ในฐานข้อมูล
2. **การสร้าง association rule** หลังจากที่ได้หา frequent itemset ได้แล้วจะนำรูปแบบที่หาได้มาสร้างเป็นกฎความสัมพันธ์โดย เช่น Apple => Cereal หมายความว่าเมื่อลูกค้าซื้อ Apple แล้วลูกค้าจะซื้อ Cereal ร่วมไปด้วย

ในบทความนี้จะใช้ตัวอย่างข้อมูลที่แสดงในภาพที่ 1 ซึ่งเรียกว่า transaction database ในฐานข้อมูลนี้จะประกอบด้วย 4 transaction และมีสินค้าที่ลูกค้าซื้อทั้งหมด 5 ประเภทคือ Apple, Beer, Cereal, Diapers และ Eggs

Transaction ID	Items
1	Apple, Cereal, Diapers
2	Beer, Cereal, Eggs
3	Apple, Beer, Cereal, Eggs
4	Beer, Eggs

ภาพที่ 1 แสดง transaction database ที่จะใช้หากฎความสัมพันธ์

ในบทความนี้จะแสดงตัวอย่างการหากฎความสัมพันธ์โดยกำหนดค่า minimum support ไว้ที่ 50% นั่นคือสินค้าที่มีการซื้อมากกว่าหรือเท่ากับ 50% จะถือว่าเป็น frequent itemset ขั้นตอนการทำงานมีดังต่อไปนี้

1. คำนวณค่า support ของแต่ละ item ดังแสดงในภาพที่ 2 แต่ละแถวคือข้อมูลสินค้าแต่ละชนิดและแต่ละคอลัมน์มีความหมายดังนี้

- คอลัมน์ที่ 1 คือชื่อสินค้าหรือ item หรือ itemset
- คอลัมน์ที่ 2-5 คือหมายเลขของ transaction ถ้ามีการซื้อสินค้าใน transaction ใดที่คอลัมน์ของ transaction นั้นจะมีหมายเลข 1 แต่ถ้าไม่มีจะเป็นเลข 0
- คอลัมน์ที่ 6 คือ ค่า support

Items	Transaction ID				Support
	1	2	3	4	
Apple	1	0	1	0	$2/4 = 50\%$
Beer	0	1	1	1	$3/4 = 75\%$
Cereal	1	1	1	0	$3/4 = 75\%$
Diapers	1	0	0	0	$1/4 = 25\%$
Eggs	0	1	1	1	$3/4 = 75\%$

ภาพที่ 2 แสดงการคำนวณค่า support ของสินค้าแต่ละชนิด

จากค่า support ที่คำนวณได้ในภาพที่ 2 จะเห็นว่า Diapers มีค่า support ต่ำกว่าค่า minimum support ($25\% < 50\%$) ดังนั้น Diapers จะถูกตัดออกและไม่นำไปพิจารณาสร้างเป็น itemset ที่มีความยาว 2 ต่อไป ดังภาพที่ 3 และเรียก Apple, Beer, Cereal และ Eggs ว่า frequent itemset

Items	Transaction ID				Support
	1	2	3	4	
Apple	1	0	1	0	2/4 = 50%
Beer	0	1	1	1	3/4 = 75%
Cereal	1	1	1	0	3/4 = 75%
Diapers	1	0	0	0	1/4 = 25%
Eggs	0	1	1	1	3/4 = 75%

ภาพที่ 3 Diapers ถูกตัดทิ้งเนื่องจากมีค่า support ต่ำกว่าค่า minimum support ที่กำหนดไว้

2. นำ frequent itemset ที่ได้จากภาพที่ 3 มาสร้างเป็น itemset ที่มีความยาว 2 โดยใช้การ join ได้เป็น {Apple, Beer}, {Apple, Cereal}, {Apple, Eggs}, {Beer, Cereal}, {Cereal, Eggs} และเนื่องจากเป็นเซต ลำดับของข้อมูลไม่มีผล นั่นคือ {Apple, Beer} = {Beer, Apple}

ส่วนการคำนวณค่า support ก็นำข้อมูล transaction ในภาพที่ 3 มาทำ intersect กัน (หรือใช้ bit operation AND ก็ได้) เช่น การหา support ของ {Apple, Beer} จะเกิดจาก

•transaction(Apple) = {1,0,1,0}

•transaction(Beer) = {0,1,1,1}

•transaction(Apple, Beer) = {0,0,1,0}

ค่า support ของ itemsets ที่มีความยาว 2 ทั้งหมดจะแสดงในภาพที่ 4 และ itemset ที่มีค่า support น้อยกว่า minimum support จะถูกตัดทิ้งไปทำให้เหลือ frequent itemset ที่มีความยาว 2 เพียงแค่ {Apple, Cereal}, {Beer, Cereal}, {Beer, Eggs}, {Cereal, Eggs} ดังแสดงในภาพที่ 5

Itemset	Transaction ID				Support
	1	2	3	4	
{Apple, Beer}	0	0	1	0	$1/4 = 25\%$
{Apple, Cereal}	1	0	1	0	$2/4 = 50\%$
{Apple, Eggs}	0	0	1	0	$1/4 = 25\%$
{Beer, Cereal}	0	1	1	0	$2/4 = 50\%$
{Beer, Eggs}	0	1	1	1	$3/4 = 75\%$
{Cereal, Eggs}	0	1	1	0	$2/4 = 50\%$

ภาพที่ 4 แสดงค่า support ของ itemset ที่มีความยาว 2

Itemset	Transaction ID				Support
	1	2	3	4	
{Apple, Beer}	0	0	1	0	$1/4 = 25\%$
{Apple, Cereal}	1	0	1	0	$2/4 = 50\%$
{Apple, Eggs}	0	0	1	0	$1/4 = 25\%$
{Beer, Cereal}	0	1	1	0	$2/4 = 50\%$
{Beer, Eggs}	0	1	1	1	$3/4 = 75\%$
{Cereal, Eggs}	0	1	1	0	$2/4 = 50\%$

ภาพที่ 5 แสดง frequent itemset ที่มีความยาว 2

3. ถัดมาทำการสร้าง itemset ที่มีความยาวเท่ากับ 3 โดยมีเงื่อนไขว่าข้อมูล item แรกจะต้องมีค่าเหมือนกันจึงสามารถทำการ join กันได้ เช่น {Beer, Cereal, Eggs} เกิดจากการ join {Beer, Cereal} และ {Beer, Eggs} แต่ไม่สามารถ join {Apple, Cereal} และ {Beer, Cereal} ได้ ดังนั้นในขั้นตอนนี้จะมีเพียงแต่ itemset เดียวคือ {Beer, Cereal, Eggs} และเป็น frequent itemset ดังแสดงในภาพที่ 6

Items	Transaction ID				Support
	1	2	3	4	
{Beer, Cereal, Eggs}	0	1	1	0	2/4 = 50%

ภาพที่ 6 แสดง itemset ที่มีความยาวเท่ากับ 3

จากภาพที่ 6 เราไม่สามารถสร้าง itemset ที่มีความยาวมากขึ้นกว่านี้ได้อีก ขั้นตอนการหา frequent itemset จึงหยุด และได้ frequent itemsets ทั้งหมดดังแสดงในภาพที่ 7

การหา frequent itemset ทั้งหมด

Frequent itemset	Support	Size
{Apple}	$2/4 = 50\%$	1
{Beer}	$3/4 = 75\%$	1
{Cereal}	$3/4 = 75\%$	1
{Eggs}	$3/4 = 75\%$	1
{Apple, Cereal}	$2/4 = 50\%$	2
{Beer, Cereal}	$2/4 = 50\%$	2
{Beer, Eggs}	$3/4 = 75\%$	2
{Cereal, Eggs}	$2/4 = 50\%$	2
{Beer, Cereal, Eggs}	$2/4 = 50\%$	3

ภาพที่ xxx แสดง frequent itemset ทั้งหมดที่หาได้

4. ขั้นตอนถัดมาคือการสร้างกฎความสัมพันธ์จาก frequent itemset ที่หาได้ โดยจะพิจารณา frequent itemset ที่มีความยาวมากกว่า 2 item ขึ้นไปมาสร้างเป็นกฎความสัมพันธ์ เช่น {Apple, Cereal} จะสร้างกฎความสัมพันธ์ได้เป็น Apple $\Rightarrow$ Cereal เป็นต้น โดยกฎความสัมพันธ์ทั่วไปจะแสดงเป็น

LHS $\Rightarrow$ RHS

โดยที่ LHS (Left Hand Side) แสดงรูปแบบของ itemset ด้านซ้ายของกฎความสัมพันธ์ และ RHS (Right Hand Side) แสดงรูปแบบของ itemset ด้านขวาของกฎความสัมพันธ์
ในการพิจารณาว่ากฎความสัมพันธ์ที่สร้างได้ดีหรือไม่จำเป็นจะต้องมีตัววัดประสิทธิภาพของกฎ ในบทความนี้จะขอแนะนำ 2 ตัววัด คือ confidence และ lift

•ค่า confidence แสดงความเชื่อมั่นของกฎความสัมพันธ์ที่เมื่อรูปแบบ LHS เกิดขึ้นแล้วรูปแบบ RHS จะเกิดขึ้นด้วยเป็นจำนวนที่เปอร์เซ็นต์ การคำนวณค่า confidence หาได้จาก

$$\text{confidence}(\text{LHS} \Rightarrow \text{RHS}) = \frac{\text{support}(\text{LHS}, \text{RHS})}{\text{support}(\text{LHS})}$$

โดยที่ support (LHS, RHS) คือ ค่า support ที่รูปแบบ LHS และ RHS เกิดขึ้นพร้อมๆ กัน ตัวอย่างการหา confidence ของกฎ Apple $\Rightarrow$ Cereal แสดงได้ดังนี้

$$\begin{aligned}\text{confidence}(\text{Apple} \Rightarrow \text{Cereal}) &= \frac{\text{support}(\text{Apple}, \text{Cereal})}{\text{support}(\text{Apple})} \\ &= \frac{2/4}{2/4} \\ &= 100\%\end{aligned}$$

ค่า lift คือค่าที่บ่งบอกว่าการเกิดรูปแบบ LHS และ RHS มีความสัมพันธ์กันแค่ไหน โดยถ้าค่า lift เป็น 1 แสดงว่ารูปแบบ LHS และ RHS ไม่ขึ้นต่อกัน (independent) ค่า lift คำนวณได้จาก

$$\text{lift}(\text{LHS} \Rightarrow \text{RHS}) = \frac{\text{support}(\text{LHS}, \text{RHS})}{\text{support}(\text{LHS}) \times \text{support}(\text{RHS})}$$

ตัวอย่างของการหาค่า lift ของกฎ Apple => Cereal แสดงได้ดังนี้

$$\begin{aligned}\text{lift}(\text{Apple} \Rightarrow \text{Cereal}) &= \frac{\text{support}(\text{Apple}, \text{Cereal})}{\text{support}(\text{Apple}) \times \text{support}(\text{Cereal})} \\ &= \frac{2/4}{2/4 \times 3/4} \\ &= 1.33\end{aligned}$$

ค่า confidence และ lift ของกฎความสัมพันธ์ทั้งหมดที่สร้างได้แสดงในภาพที่ 8 โดยทำการเรียงตามค่า confidence และ lift จากมากไปหาน้อย

Rule No.	Frequent itemset	Confidence	Lift
1	Apple $\Rightarrow$ Cereal	100%	1.33
2	Beer $\Rightarrow$ Eggs	100%	1.33
3	Eggs $\Rightarrow$ Beer	100%	1.33
4	Beer, Cereal $\Rightarrow$ Eggs	100%	1.33
5	Cereal, Eggs $\Rightarrow$ Beer	100%	1.33
6	Cereal $\Rightarrow$ Apple	67%	1.33
7	Beer $\Rightarrow$ Cereal, Eggs	67%	1.33
8	Eggs $\Rightarrow$ Beer, Cereal	67%	1.33
9	Beer $\Rightarrow$ Cereal	67%	0.89
10	Cereal $\Rightarrow$ Beer	67%	0.89
11	Cereal $\Rightarrow$ Eggs	67%	0.89
12	Eggs $\Rightarrow$ Cereal	67%	0.89
13	Cereal $\Rightarrow$ Beer, Eggs	67%	0.89
14	Beer, Eggs $\Rightarrow$ Cereal	67%	0.89

การหาความสัมพันธ์ หรือเรียกว่า กฎความสัมพันธ์

เริ่มจากการ Market Basket Analysis จากการมองเห็นว่าคนซื้อสินค้า 1 ชิ้น แล้วน่าจะเลือกซื้อสินค้าใดต่อร่วม จากข้อมูลที่ได้จากการซื้อสินค้าแต่ละครั้ง จะพบว่าการซื้อครั้งที่ 1 ซื้ออะไรบ้าง แล้ว Record ไว้ จากนั้นทำการบันทึกข้อมูลคนถัด ๆ ไปเรื่อย ๆ จำนวนแถวข้อมูลการซื้อก็จะเพิ่มมากยิ่งขึ้นเรื่อย ๆ เป็น Transaction ซึ่งข้อมูลที่ได้ก็ไม่สะดวกในการนำไปวิเคราะห์ โดยทางที่ดีก็นำข้อมูลมาทำการแปลงข้อมูลเป็นแถว โดย Group by ตามเวลาที่ซื้อสินค้า

Support

นับจำนวนครั้งการซื้อสินค้าแต่ละชนิดคิดเป็น % ของการซื้อสินค้า หรือการพบสินค้า ทำการนับในการซื้อสินค้า (จำนวนครั้ง)

Frequent itemset

จำนวนสินค้าที่มีการพบบ่อย รูปแบบของการซื้อสินค้า (items) ที่มีค่า support $\geq$ minimum support การมี Frequent กับ infrequent เพื่อเป็นการกรองข้อมูลออกไป

สินค้าใน Supermarket มีจำนวน 3 ชนิด โอกาสที่เป็นไปได้ทั้งหมด (หรือ Subset ทั้งหมด) ถ้าตัวไหนไม่ผ่านเงื่อนไขที่กำหนดให้ตัดทิ้งเลย

การหากฎความสัมพันธ์ (association rules) มี 2 อัลกอริทึมใหญ่ ๆ

- Apriori : แอบเพียวรี (Agrawal and Srikant, 1994)

สร้างรูปแบบสินค้าที่มีการเพิ่มจำนวนทีละ 1 ถ้าผ่านเงื่อนไขก็จะทำการ Join เพิ่มไปเรื่อย ๆ แบบนี้มันจะช้า เพราะการหาความสัมพันธ์แต่ละครั้ง ก็ต้องไปดึงค่าจาก Database ทุกครั้ง หลักการทำงาน superset ของรูปแบบที่เป็น frequent Apriori นับจำนวนครั้งในการซื้อที่น้อยกว่า 50% ตัดผ้าอ้อมออก จับคู่ให้เป็น set ขึ้น เมื่อเป็น set ข้อมูลที่อยู่ข้างในไม่จำเป็นต้องมีลำดับ 1 ตัวหน้าที่ทำการ JOIN กันต้องเหมือนกัน เช่น {Beer, Cereal, Eggs} เพื่อให้ได้มา Frequent Itemset สินค้าที่มีการซื้อมากกว่าหรือเท่ากับ 50% จะได้ขนาด size ออกมาด้วย

- FP-Growth : เอฟพีไทล

ขั้นตอนที่ 1 ทำโดยการสร้าง Compact data Structure เรียก FP-Tree ขึ้นมา โดยการอ่านข้อมูลจากฐานข้อมูลเพียงแค่ 2 ครั้งเท่านั้น แล้วหาค่า Support ของรูปแบบการซื้อสินค้า (Items) ที่มีความยาว 1 แล้วเรียงลำดับตาม Support ที่คำนวณได้ แล้วมาทำการสร้าง FP-Tree จากการอ่านข้อมูลในฐานข้อมูลเพียงหนึ่งรอบ เช่น เจอ Apple 1 ครั้ง มันจะมีลิงค์ต่อไปเกิดเป็น 1 เส้นทาง ที่วิ่งไปโดยดูว่าตัวไหนที่มันสามารถซ้อนกันได้ก็ซ้อนไป ข้อดีคือถ้ามี Database 10 ล้านแถว เราสามารถทำทีละส่วนได้ แล้ว Free memory ทิ้งไปได้ แล้วเรียกส่วนที่เหลือมาทำงานต่อได้ ค่อยๆ อ่านค่ามาทำงาน Operator แบบนี้จะเรียก Stream Database

ขั้นตอนที่ 2 สร้างรูปแบบของสินค้าที่ชื้อบ่อย ๆ (Frequent items) จาก FP-Tree โดยกำหนด min_support = 50%

เปรียบเทียบผลของ Apriori & FP-Growth จะพบว่า FP-Growth จะเร็วกว่าแต่ผลลัพธ์เหมือนกัน

Association Rules : สร้างกฎความสัมพันธ์

สร้างจากสินค้าที่ลูกค้าซื้อบ่อย ๆ รูปแบบของกฎ คือ LHS , RHS

LHS คือ Left Hand Side สินค้าที่ซื้อพร้อมกันบ่อย ๆ ด้านซ้ายของกฎ เช่น Apple Cereal

RHS คือ Right Hand Side สินค้าที่ซื้อพร้อมกันบ่อย ๆ ด้านขวาของกฎ เช่น Cereal Apple

ตัววัดประสิทธิภาพของกฎความสัมพันธ์

- **Confidence** แสดงความเชื่อมั่นของกฎความสัมพันธ์ที่เมื่อรูปแบบ LHS เกิดขึ้นแล้วรูปแบบ RHS จะเกิดขึ้น % ความน่าจะเป็น 1 ความน่าจะเป็นจะเกิดขึ้นด้วยทุกครั้ง ถ้า Confidence กฎนั้นสูงน่าสนใจกว่า
- **Lift** ถ้า Lift เกิน 1 แสดงว่าโอกาสจะเกิดขึ้นพร้อมกันมากกว่าแยก ถ้าน้อยกว่า 1 แสดงว่าตัวล่างมากกว่าตัวบน แสดงว่าซื้อแยกมากกว่า

Applications

ใช้ในการเพิ่มยอดขายโดยการแนะนำสินค้าที่ลูกค้ามักจะซื้อพร้อมกันบ่อย ๆ (cross-selling) ใช้ในการจัดสินค้าในร้าน เช่น วางสินค้าที่ลูกค้ามักจะซื้อพร้อม ๆ กันไว้ใกล้ๆ กัน หรือวางสินค้าไว้ใน Catalog ต่าง ๆ

Support

Transaction ID	Items
1	Apple, Cereal, Diapers
2	Beer, Cereal, Eggs
3	Apple, Beer, Cereal, Eggs
4	Beer, Eggs

Items	Transaction ID				Support
	1	2	3	4	
Apple	1	0	1	0	$2/4 = 50\%$
Beer	0	1	1	1	$3/4 = 75\%$
Cereal	1	1	1	0	$3/4 = 75\%$
Diapers	1	0	0	0	$1/4 = 25\%$
Eggs	0	1	1	1	$3/4 = 75\%$

การแปลงข้อมูล Data Transformation

Transaction ID	Items
1	Apple, Cereal, Diapers
2	Beer, Cereal, Eggs
3	Apple, Beer, Cereal, Eggs
4	Beer, Eggs

Tran_id	apple	cereal	diapers	eggs	beer
1	Y	Y	Y	?	
2	?	Y	?	Y	
3	Y	Y	?	Y	Y
4	?	?	?	Y	Y

Support

Itemset	Transaction ID				Support
	1	2	3	4	
{ Apple , Beer }	0	0	1	0	$1/4 = 25\%$
{ Apple , Cereal }	1	0	1	0	$2/4 = 50\%$
{ Apple , Eggs }	0	0	1	0	$1/4 = 25\%$
{ Beer , Cereal }	0	1	1	0	$2/4 = 50\%$
{ Beer , Eggs }	0	1	1	1	$3/4 = 75\%$
{ Cereal , Eggs }	0	1	1	0	$2/4 = 50\%$

Support > 50%

Items	Transaction ID				Support
	1	2	3	4	
Apple	1	0	1	0	$2/4 = 50\%$
Beer	0	1	1	1	$3/4 = 75\%$
Cereal	1	1	1	0	$3/4 = 75\%$
Diapers	1	0	0	0	$1/4 = 25\%$
Eggs	0	1	1	1	$3/4 = 75\%$

อ้างอิง

- <http://dataminingtrend.com/2014/association-rules/>
- <http://www.autosoft.in.th/data-science/forecasting-with-the-microsoft-time-series-data-mining-algorithm/>