การแนะนำหนังสือด้วยเทคนิคเหมืองข้อมูลโดยใช้ RapidMiner

BOOK RECOMMENDATION WITH DATA MINING USING RAPIDMINER

วรรษา เงินดี

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2563

การแนะนำหนังสือด้วยเทคนิคเหมืองข้อมูลโดยใช้ RapidMiner

วรรษา เงินดี

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2563

BOOK RECOMMENDATION WITH DATA MINING USING RAPIDMINER

WANSA NGOENDEE

A Master's Project Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Information Technology)

Faculty of Science, Srinakharinwirot University

2020

สารนิพนธ์

เรื่อง

การแนะนำหนังสือด้วยเทคนิคเหมืองข้อมูลโดยใช้ RapidMiner

ของ

วรรษา เงินดี

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร.วีรยุทธ เจริญเรืองกิจ)

..... ประธาน

(อาจารย์ ดร.รัตน์ชัยนันท์ ธรรมสุจริต)

..... กรรมการ

(อาจารย์ ดร.โสภณ มงคลลักษมี)

ชื่อเรื่อง	การแนะนำหนังสือด้วยเทคนิคเหมืองข้อมูลโดยใช้ RapidMiner
ผู้วิจัย	วรรษา เงินดี
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2563
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. วีรยุทธ เจริญเรืองกิจ

งานวิจัยนี้นำเสนอการใช้เทคนิคเพื่อนบ้านใกล้เคียงที่สุด k-Nearest Neighbor (k-NN) และเทคนิคการแยกตัวประกอบเมทริกซ์ Matrix Factorization (MF) ในการสร้างแบบจำลองการแนะนำหนังสือด้วยทำโดยใช้ RapidMiner โดยเปรียบเทียบประสิทธิภาพระหว่าง 2 เทคนิคข้างต้น และประสิทธิภาพเทคนิคการทำ Model Combiner ของทั้ง 2 เทคนิค ซึ่งคุณลักษณะเฉพาะที่สำคัญที่นำมาใช้ในการทำเหมืองข้อมูลประกอบไปด้วย ข้อมูลผู้ใช้ (UserID) ข้อมูลหนังสือ (BookID) และข้อมูลการให้คะแนน (Rating) จากผลการวิจัยพบว่าเทคนิคที่มีประสิทธิภาพการทำนายดีที่สุดคือเทคนิค Model Combiner ด้วยค่า Area under the Curve (AUC) ที่ได้คือ 0.929, Precision at k=5 (prec@5) ที่ได้คือ 0.293, Precision at k=10 (prec@10) ที่ได้คือ 0.229, Precision at k=15 (prec@15) ที่ได้คือ 0.192, ค่า Normalized Discounted Cumulative Gain (NDCG) ที่ได้คือ 0.506 และMean Average Precision (MAP) ที่ได้คือ 0.201

คำสำคัญ : การทำเหมืองข้อมูล, ระบบแนะนำ, วิธีการกรองร่วม, เทคนิคเพื่อนบ้านใกล้เคียง, เทคนิคการแยกตัวประกอบ, โปรแกรม Rapid Miner

Title	BOOK RECOMMENDATION WITH DATA MINING USING RAPIDMINER
Author	WANSA NGOENDEE
Degree	MASTER OF SCIENCE
Academic Year	2020
Thesis Advisor	Assistant Professor Dr. Werayuth Charoenruengkit

This research presents the use of the k-Nearest Neighbor (k-NN) and the Matrix Factorization (MF) techniques for a book recommendation system using RapidMiner. The combination of the two techniques and the results were compared and the features used by the techniques were UserID and BookID, and the book ratings were given by users. The experimental results demonstrated that the combination technique achieved the best results with the area under the curve (AUC) at 0.929, precision at k=5 (prec@5) at 0.293, precision at k=10 (prec@10) at 0.229, precision at k=15 (prec@15) at 0.192, Normalized Discounted Cumulative Gain (NDCG) at 0.506 and the resulting Mean Average Precision (MAP) at 0.201.

Keyword : Data Mining, Recommendation, Collaborative Filtering, k-Nearest Neighbor, Matrix Factorization, RapidMiner

กิตติกรรมประกาศ

ในการจัดทำสารนิพนธ์ฉบับนี้ให้สำเร็จลุล่วง ผู้วิจัยขอกราบขอบคุณ ผศ.ดร.วีรยุทธ เจริญเรืองกิจ ซึ่งเป็นอาจารย์ที่ปรึกษาหลักที่ให้ความรู้ด้านวิชาการที่เกี่ยวข้องกับการจัดทำสารนิพนธ์ตลอดจนให้คำปรึกษาและแนะนำในการปรับปรุงแก้ไขเพื่อให้สารนิพนธ์ฉบับนี้มีความสมบูรณ์มากที่สุด และขอขอบคุณคณะอาจารย์ทุกท่านที่ให้ความรู้ในสาระวิชาต่าง ๆ ในการศึกษาระดับปริญญาโทมาตลอดการศึกษานี้

สุดท้ายนี้ขอขอบคุณบิดา มารดา และน้องสาว ที่ให้การสนับสนุนและส่งกำลังใจมาให้โดยตลอด และขอบคุณเพื่อน พี่ และน้องร่วมเรียนสาขาเทคโนโลยีสารสนเทศที่ให้ความช่วยเหลือทั้งทางตรงและทางอ้อม

วรรษา เงินดี

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันเทคโนโลยีสารสนเทศเข้ามามีบทบาทมากขึ้นในการใช้ชีวิตประจำวันล้วน สามารถเข้าถึงสิ่งต่าง ๆ ได้อย่างสะดวกและรวดเร็วผ่านระบบอินเตอร์เน็ตที่สามารถเข้าถึงได้ทุกที่ทุกเวลา ซึ่งไม่เพียงแต่ระบบซื้อขายสินค้าและบริการเท่านั้น ระบบหนังสือออนไลน์ยังเป็นที่นิยมอย่างแพร่หลาย ทั้งระบบซื้อขายหนังสือในรูปแบบหนังสืออิเล็กทรอนิกส์ (E-book) หรือระบบอ่านหนังสือออนไลน์ เป็นต้น ซึ่งข้อมูลหนังสือที่อยู่บนระบบเหล่านี้มีจำนวนมาก ผู้ใช้งานอาจไม่สามารถเข้าถึงหนังสือที่อยู่ในระบบได้ทั้งหมดผ่านการค้นหา (Search Engine) ได้เพียงอย่างเดียว จึงอาจเป็นสาเหตุให้ผู้ใช้ไม่สามารถเข้าถึงหนังสือที่มีเนื้อหา หรือหมวดหมู่หนังสือที่ต้องการได้อย่างทั่วถึง จึงได้มีแนวคิดที่จะนำระบบการแนะนำเข้ามาเป็นส่วนช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลที่อยู่บนระบบได้อย่างทั่วถึง

ระบบการแนะนำ (Recommender System) เป็นระบบที่ถูกนำมาใช้บนเว็บไซต์ต่าง ๆ อย่างแพร่หลายในปัจจุบัน เช่น Shopee, Lazada, Amazon.com เป็นต้น (นภวรรณ ดุษฎีเวทกุล, 2560) เพื่อช่วยกระตุ้นยอดขายให้กับร้านค้าออนไลน์ จากการคัดเลือกสินค้าที่คาดว่าผู้ใช้จะสนใจมาแสดงผลให้ผู้ใช้ได้เลือกแทนการแสดงผลข้อมูลทั้งหมด อีกทั้งยังสามารถแนะนำสินค้าที่เกี่ยวกับสินค้าที่ผู้ใช้สนใจจากประวัติการซื้อของผู้ใช้ท่านอื่น ๆ มานำเสนอได้อีกด้วย ซึ่งจะเพิ่มโอกาสที่ผู้ใช้จะเลือกซื้อสินค้านอกเหนือจากสินค้นที่ต้องการได้มากขึ้น โดยส่วนใหญ่ระบบแนะนำจะใช้วิธีที่แตกต่างกัน ซึ่งสามารถแบ่งประเภทของการแนะนำได้ออกเป็น 3 ประเภทหลัก ๆ ได้แก่ 1)Content-based Filtering คือการแนะนำด้วยวิธีการพิจารณาจากลักษณะข้อมูลสิ่งของหรือคุณลักษณะของสินค้าจากประวัติที่ผู้ใช้คนนั้น ๆ ชื่อชอบ หรือเคยซื้อ 2)Collaborative Filtering คือ การแนะนำด้วยวิธีการพิจารณาจากการกรองร่วมหรือจากการหาผู้ใช้ในระบบที่มีลักษณะคล้ายคลึงกับผู้ใช้ที่เป็นเป้าหมาย ความคล้ายคลึงในที่นี้เช่นมีประวัติเคยเลือกซื้อสินค้า 3)Hybrid คือการแนะนำด้วยวิธีการแบบผสมโดยนำเอาหลาย ๆ วิธีการมารวมกันเพื่อช่วยเพิ่มประสิทธิภาพของระบบแนะนำ (นภวรรณ ดุษฎีเวทกุล, 2560) ดังนั้นผู้วิจัยจึงมีความสนใจอยากศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิคการทำเหมืองข้อมูล (Data Mining) และเปรียบเทียบประสิทธิภาพของเทคนิค เพื่อให้ได้เทคนิคที่เหมาะสมในการนำมาสร้างแบบจำลองการแนะนำหนังสือ

โดยสรุปงานวิจัยนี้มีแนวคิดเพื่อศึกษาและเปรียบเทียบเทคนิคที่ใช้ในการสร้างแบบจำลองแนะนำหนังสือ เพื่อช่วยให้สามารถนำเสนอหนังสือที่ผู้ใช้งานแต่ละคนต้องการได้ อีกทั้งเพื่อให้ผู้ใช้งานสามารถเข้าถึงหนังสือที่มีอยู่ในระบบได้มากยิ่งขึ้น

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาและเปรียบเทียบเทคนิคที่เหมาะสมที่ใช้ในการสร้างแบบจำลองการแนะนำหนังสือด้วยเทคนิคเหมืองข้อมูล (Data Mining) เพื่อให้สามารถแนะนำหนังสือได้ตรงตามความต้องการของผู้ใช้งาน

1.3 ขอบเขตของการวิจัย

1.3.1 ข้อมูลที่นำมาใช้ในการศึกษาวิจัยเป็นข้อมูลจาก Kaggle ชื่อว่า goodbooks-10k โดยเป็นข้อมูล ratings หรือ ตารางข้อมูลการให้คะแนนหนังสือ

1.3.2 ศึกษาเทคนิคเหมืองข้อมูล (Data Mining) เพื่อนำมาประยุกต์ใช้ในการสร้างแบบจำลองการแนะนำหนังสือ

1.3.3 เปรียบเทียบประสิทธิภาพของเทคนิค k-Nearest Neighbor (k-NN), Matrix Factorization และ Model Combiner (k-NN + MF)

1.3.4 ใช้ซอฟต์แวร์ Rapid Miner Studio ในการประมวลผล และสร้างแบบจำลองการแนะนำหนังสือ

1.4 วิธีดำเนินการวิจัย

1.4.1 ทบทวนวรรณกรรมที่เกี่ยวข้อง (Literature Review)

1.4.2 ศึกษาเทคนิคเหมืองข้อมูล (Data Mining) เพื่อนำมาประยุกต์ใช้ในการสร้างแบบจำลองการแนะนำหนังสือ

1.4.3 กำหนดขอบเขตการศึกษา

1.4.4 ใช้ซอฟต์แวร์ Rapid Miner Studio ในการประมวลผล และสร้างแบบจำลอง

1.4.5 เปรียบเทียบประสิทธิภาพการทำงานของแบบจำลองจากเทคนิคแต่ละประเภท

1.4.6 สรุปผล และอภิปรายผลการศึกษา

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 สามารถศึกษาการใช้งานเทคนิค k-Nearest Neighbor และ Matrix Factorization ในการสร้างแบบจำลองการแนะนำหนังสือได้

1.5.2 สามารถปรับปรุงแบบจำลองให้เหมาะสมกับชุดข้อมูลที่เลือกใช้ได้

1.5.3 สามารถศึกษาการใช้โปรแกรม Rapid Miner Studio ในการวิเคราะห์ข้อมูล ประมวลผลข้อมูล และการสร้างแบบจำลองการแนะนำที่เหมาะสมกับชุดข้อมูลที่เลือกใช้ได้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และได้นำเสนอตามหัวข้อดังนี้

1. Data Mining
2. k-Nearest Neighbors (k-NN)
3. Matrix Factorization
4. ตัววัดประสิทธิภาพแบบจำลอง
5. โปรแกรม Rapid miner studio 9.6
6. Recommendation
7. งานวิจัยที่เกี่ยวข้อง

2.1 Data Mining

เทคนิคการทำเหมืองข้อมูล (Data Mining) นั้น บุญเสริม กิจศิริกุล (2546) ได้ให้คำนิยามไว้ว่าเป็นกระบวนการที่กระทำกับข้อมูลจำนวนมากเพื่อค้นหาสิ่งที่สามารถเป็นประโยชน์จากข้อมูลเหล่านั้น ด้วยการหารูปแบบหรือความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล เพื่อนำไปเป็นส่วนหนึ่งในการวิเคราะห์และประกอบการตัดสินใจในด้านต่าง ๆ เช่น ด้านการวิเคราะห์ยอดการสั่งซื้อของลูกค้า ที่สามารถช่วยจัดกลุ่มได้ว่ากลุ่มสินค้าใดที่มียอดสั่งซื้อมากน้อยเพียงใด เพื่อใช้วางแผนการผลิตและจำหน่ายสินค้าให้ตรงตามกลุ่มเป้าหมาย โดยการทำเหมืองข้อมูลสามารถแบ่งออกเป็น 2 ประเภทดังนี้

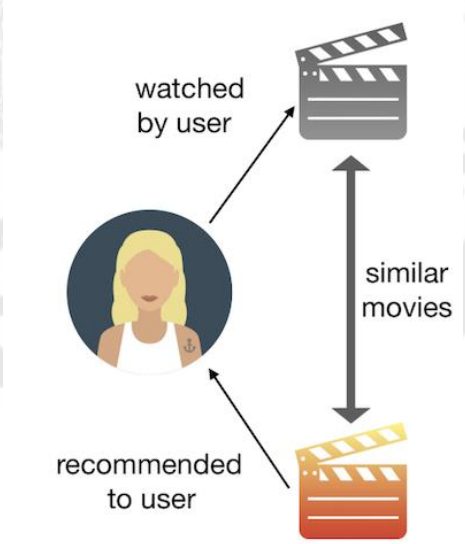
2.1.1 การสร้างแบบจำลองในการทำนาย (Predictive modeling, Supervised modeling) ด้วยกระบวนการแยกข้อมูลโดยการเรียนรู้ตามคุณสมบัติของฉลาก (Label) ซึ่งฉลากนี้คือคุณสมบัติที่จะอยู่ในทุก ๆ ข้อมูล และเป็นค่าที่ใช้ในการทำนายผลของข้อมูล

2.1.2 การสร้างแบบจำลองในการบรรยาย (Descriptive modeling, Unsupervised modeling) เป็นกระบวนการที่หาความสัมพันธ์ต่าง ๆ หรือการจัดกลุ่มของข้อมูล โดยไม่ได้มีจุดมุ่งหมายเพื่อการทำนาย

2.2 Recommendation

ระบบการแนะนำ (Recommender System) เป็นระบบที่ถูกนำมาเป็นเครื่องมือเพื่อช่วยกระตุ้นยอดขายให้กับร้านค้าออนไลน์ตามเว็บไซต์ต่าง ๆ โดยระบบจะแนะนำสินค้าที่คาดว่าผู้ใช้จะสนใจ อีกทั้งยังสามารถเพิ่มโอกาสที่ผู้ใช้จะเลือกซื้อสินค้านอกจากสินค้นที่ต้องการมากขึ้น โดยส่วนใหญ่ระบบแนะนำสินค้านั้น จะแบ่งออกเป็น 3 ประเภท ดังนี้

2.2.1 Content-based Filtering คือการแนะนำด้วยวิธีการพิจารณาจากลักษณะข้อมูลสิ่งของหรือคุณลักษณะของสินค้าจากประวัติที่ผู้ใช้คนนั้น ๆ ชื่อชอบ หรือเคยซื้อ ซึ่งส่งผลให้การแนะนำตรงตามความต้องการของผู้ใช้งาน แต่มีข้อเสียคือผู้ใช้จะได้รับการแนะนำสิ่งของที่ไม่ได้แตกต่างไปจากเดิม ทำให้ผู้ใช้ได้รับการแนะนำที่มีความหลากหลายค่อนข้างน้อย จากภาพประกอบ 1 แสดงรูปแบบการแนะนำโดยพิจารณาจากข้อมูลสิ่งของ ยกตัวอย่างเช่น ผู้ใช้ A เข้าชมหนังประเภทสืบสืบสวนสอบสวน ระบบก็จะแนะนำหนังประเภทเดียวกันซึ่งคือหนังสืบสวนสอบสวนให้กับผู้ใช้ A



ภาพประกอบ 1 ภาพตัวอย่างการทำงานของการแนะนำหนังที่มีความคล้ายกัน

ที่มา : Emma Grimaldi (2018)

จากตาราง 1 สมมติว่าผู้ใช้ X ให้คะแนนความชื่นชอบต่อหนังเรื่องที่ 1 (Movie1) ซึ่งเป็นหนังประเภท action และ Sci-fi โดยมีนักแสดงคือ A, B และ C ระบบแนะนำประเภท Content-

based Filtering จะพิจารณาว่าในระบบมีหนังเรื่องใดบ้างที่มีคุณลักษณะที่คล้ายคลึงกันกับหนังเรื่อง Avenger มากที่สุด ซึ่งได้ว่าหนังเรื่อง 2 (Movie2) ที่เป็นหนังประเภท action sci-fi และมีนักแสดงนำครบทั้ง 3 คน รองลงมาคือหนังเรื่องที่ 4 (Movie4) ที่ที่เป็นหนังประเภท action sci-fi แต่มีนักแสดงนำแค่ A และ B เท่านั้น

ตาราง 1 ตัวอย่างการทำงานของเทคนิคการแนะนำแบบ Content-based Filtering

	Category			Actos				
Movie1	action	Sci-fi	drama	A	B	C	D	F
Movie2	✓	✓		✓	✓	✓		
Movie3	✓	✓		✓	✓	✓		
Movie4	✓	✓		✓	✓			
Movie1			✓				✓	✓

2.2.2 Collaborative Filtering คือ การแนะนำด้วยวิธีการพิจารณาจากการกรองร่วมหรือจากการหาผู้ใช้ในระบบที่มีลักษณะคล้ายคลึงกับผู้ใช้ที่เป็นเป้าหมาย เช่นมีประวัติการเลือกซื้อสินค้า โดยจะนิยมใช้กับระบบที่มีการให้คะแนนความชื่นชอบ (Rating) ต่อสินค้า (Item) ในการหาความคล้ายคลึงนั้นแบ่งออกเป็น 2 ประเภท คือ

1. User-based หรือการหาความคล้ายคลึงระหว่างผู้ใช้ คือการหาว่ามีผู้ใช้คนใดในระบบบ้างที่มีความคล้ายกันกับผู้ใช้เป้าหมาย ยกตัวอย่างจากตาราง 2 ผู้ใช้ A และ ผู้ใช้ B มีการให้คะแนนหนังเรื่องที่ 1 (Movie1) และหนังเรื่องที่ 2 (Movie2) ไปในทิศทางเดียวกันคือให้คะแนนความชื่นชอบต่อหนังทั้งสองเรื่องสูงเช่นเดียวกัน และยังให้คะแนนหนังเรื่องที่ 4 (Movie4) น้อยเหมือนกันด้วย ระบบก็จะมองว่าผู้ใช้ A และ B นั้นมีความคล้ายคลึงกัน

2. Item-based หรือหาความคล้ายคลึงระหว่างสินค้า คือการหาว่าสินค้าใดในระบบบ้างที่มีความคล้ายคลึงกันกับสินค้าที่เป็นเป้าหมาย ยกตัวอย่างจากตาราง 2 หนังเรื่องที่ 1 (Movie1) และหนังเรื่องที่ 2 (Movie2) เป็นหนังที่ได้รับการให้คะแนนจากผู้ใช้ A และ B ด้วยคะแนนความชื่นชอบที่สูงเหมือนกัน ระบบก็จะมองว่าหนังเรื่องที่ 1 (Movie1) และหนังเรื่องที่ 2 (Movie2) มีความคลายคลึงกัน

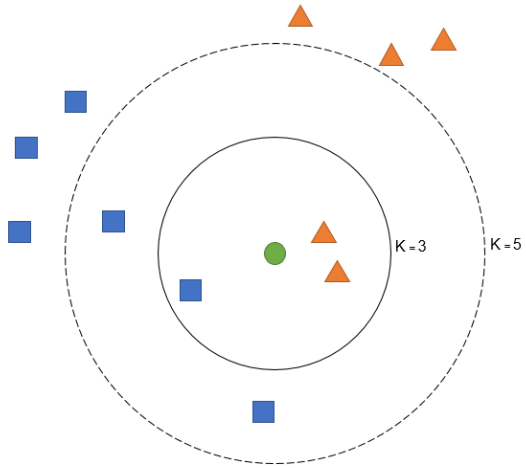
ตาราง 2 ตัวอย่างการทำงานของเทคนิคการแนะนำแบบ Collaborative Filtering

	Movie1	Movie2	Movie3	Movie4	Movie5
A	5	5		1	
B	5	4		1	
C		3	5	4	
D	4			2	4
E	3		4	5	1

2.2.3 Hybrid คือการแนะนำด้วยวิธีการแบบผสมโดยนำวิธี Content-based และ Collaborative Filtering มารวมกันเพื่อช่วยเพิ่มประสิทธิภาพของระบบแนะนำ โดยเป็นการนำข้อดีของแต่ละวิธีมาประยุกต์ใช้เพื่อเติมเต็มส่วนที่ขาดหายหรือบกพร่องของแต่ละวิธี

2.3 k-Nearest Neighbors (k-NN)

k-Nearest Neighbors หรือวิธีการเพื่อนบ้านใกล้ที่สุดจะใช้ในการจัดกลุ่มข้อมูลโดยจัดข้อมูลที่อยู่ใกล้กันเป็นกลุ่มเดียวกัน จากการตรวจสอบข้อมูลจำนวน k ที่อยู่ใกล้เคียงกันมากที่สุด ทั้งนี้วิธีการเพื่อนบ้านใกล้เคียงที่สุดนั้นเป็นอัลกอลิทึมที่นิยมใช้ในการสร้างเครื่องมือการแนะนำมากในปัจจุบัน เนื่องจากเป็นอัลกอริทึมที่ใช้งานที่ง่ายและให้ประสิทธิภาพดี นอกจากนี้ยังมีแนวทางในการแก้ปัญหาที่คล้ายกันกับ Collaborative filtering เนื่องจากใช้วิธีการจัดกลุ่มผู้ใช้ที่มีความคล้ายคลึงกัน (Similar user) หรือใกล้เคียงกันเข้าด้วยกัน ถึงแม้ว่าวิธีการเพื่อนบ้านใกล้เคียงที่สุดจะเป็นที่นิยมในการสร้างระบบแนะนำ ขณะเดียวกันก็มีความท้าทายอยู่หลายประการ หนึ่งในความท้าทายที่สุดของการสร้างแบบจำลองด้วยวิธีการเพื่อนบ้านใกล้ที่สุดคือการเลือกค่า k ซึ่งหากค่า k น้อยเกินไปก็จะมีความไวต่อสัญญาณรบกวนมาก ในทางกลับกันหากค่า k มีค่ามากเกินไปในการหาเพื่อนบ้านใกล้เคียงก็จะรวมคะแนนของคลาสอื่นเข้ามาด้วย (Torstensson, 2019)



ภาพประกอบ 2 ตัวอย่างการจัดกลุ่มข้อมูลของขั้นตอนวิธีการเพื่อนบ้านใกล้ที่สุด

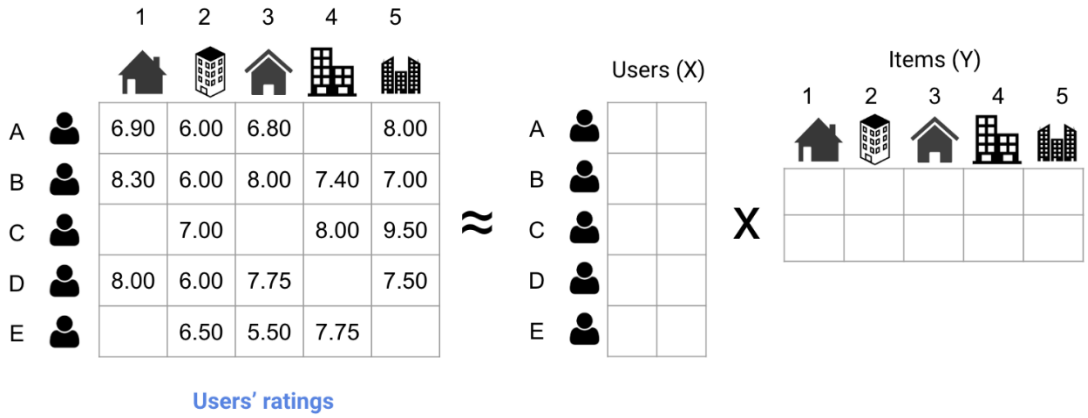
จากภาพประกอบ 2 แสดงตัวอย่างการจัดกลุ่มข้อมูล โดยกำหนดให้จุดที่พิจารณาคือวงกลมสีเขียว ควรจัดกลุ่มให้จุดที่พิจารณาไปอยู่ใน กลุ่มที่หนึ่งคือสี่เหลี่ยมสีน้ำเงิน หรือ กลุ่มที่สองคือสามเหลี่ยมสีแดง

ถ้า k = 3 แล้ว วงกลมสีเขียวจะอยู่ในกลุ่มที่สอง เพราะมี สี่เหลี่ยม 1 รูป และ สามเหลี่ยม 2 รูป อยู่ในวงกลมวงใน

ถ้า k = 5 แล้ว วงกลมสีเขียวจะอยู่ในกลุ่มที่หนึ่ง เพราะมี สี่เหลี่ยม 3 รูป และ สามเหลี่ยม 2 รูป อยู่ในวงกลมวงนอก

2.4 **เทคนิค** Matrix Factorization

Matrix Factorization คือการพยายามแยกเมทริกซ์ให้มาเป็นเมทริกซ์ย่อย ๆ ลงไป โดยที่ผลคูณของเมกทริกซ์ย่อย ๆ นั้น จะได้กลับมาเป็นเมทริกซ์ตั้งต้น ยกตัวอย่างเช่น การให้คะแนนของผู้ใช้แต่ละคน และสินค้าแต่ละชิ้น โดยจะทำการแยกเมทริกซ์นี้ออกเป็น 2 เมทริกซ์ คือ เมทริกซ์ผู้ใช้ (X) และเมทริกซ์สินค้า (Y) โดยเมื่อนำเมทริกซ์ทั้ง X และ Y นี้มาคูณกัน ก็จะได้กลับมาเป็นเมทริกซ์การให้คะแนน ตั้งต้น (Torstensson, 2019)



ภาพประกอบ 3 ตัวอย่าง Matrix User rating ที่แยกเป็น Matrix User และ Matrix Items

ที่มา: Sirinart Tangruamsub (2562)

จากภาพประกอบ 3 เพื่อให้เราสามารถหาข้อมูลการให้คะแนนในแต่ละช่องว่างของ Matrix User rating ได้ เราจำเป็นต้องทราบค่าน้ำหนักของ Matrix User และค่าน้ำหนักของ Matrix Items ที่เมื่อนำข้อมูลมาคูณกันแล้วจะได้ผลลัพธ์ใกล้เคียงกับข้อมูลใน Matrix User rating โดยเราจะใช้วิธีการเรียนรู้ต่าง ๆ เช่นการทำ Singular Value Decomposition (SVD), Principal Component Analysis (PCA) หรือ Gradient Descent

2.5 ตัววัดประสิทธิภาพแบบจำลอง

ตัววัดประสิทธิภาพโมเดลเป็นวิธีการต่าง ๆ ที่ใช้ในการประเมินคุณภาพของการจัดกลุ่ม รวมไปถึง การประเมินประสิทธิภาพของแบบจำลอง โดยการนำผลลัพธ์ที่ได้จากแบบจำลองมาสร้างเป็น Confusion Matrix จากภาพประกอบ 4 แสดงตาราง Confusion Matrix ซึ่งมีรายละเอียดดังนี้

True Positive (TP) คือจำนวนข้อมูลที่สนใจมีผลเป็นบวก และมีผลจากการทำนายเป็นบวก ตัวอย่างเช่นจำนวนครั้งที่ข้อมูลจริงผู้ป่วยเป็นไข้เลือดออก และแบบจำลองทำนายว่าผู้ป่วยเป็นไข้เลือดออก

False Positive (FP) คือจำนวนข้อมูลที่สนใจมีผลเป็นลบ และมีผลจากการทำนายเป็นบวก ตัวอย่างเช่นจำนวนครั้งที่ข้อมูลจริงผู้ป่วยไม่ได้เป็นไข้เลือดออก และแบบจำลองทำนายว่าผู้ป่วยเป็นไข้เลือดออก

False Negative (FN) คือจำนวนข้อมูลที่สนใจมีผลเป็นบวก และมีผลจากการทำนายเป็นลบ ตัวอย่างเช่นจำนวนครั้งที่ข้อมูลจริงผู้ป่วยเป็นไข้เลือดออก และแบบจำลองทำนายว่าผู้ป่วยไม่เป็นไข้เลือดออก

True Negative (TN) คือจำนวนข้อมูลที่สนใจมีผลเป็นลบ และมีผลจากการทำนายเป็นลบ ตัวอย่างเช่นจำนวนครั้งที่ข้อมูลจริงผู้ป่วยไม่ได้เป็นไข้เลือดออก และแบบจำลองทำนายว่าผู้ป่วยไม่เป็นไข้เลือดออก

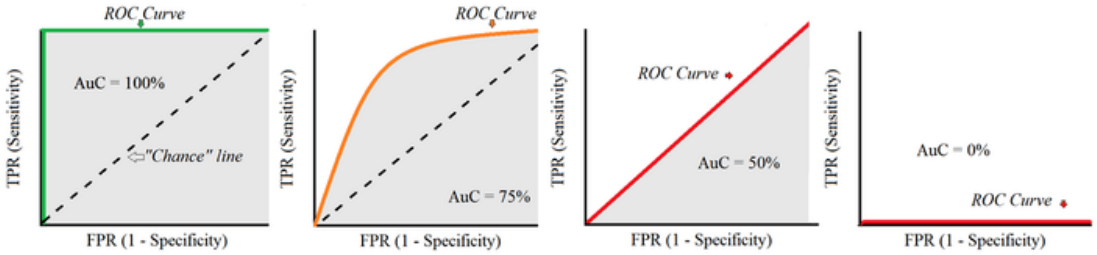
		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

ภาพประกอบ 4 แสดงตาราง Confusion Matrix

ตัววัดประสิทธิภาพโมเดลเป็นวิธีการต่าง ๆ ที่ใช้ในการประเมินคุณภาพของการจัดกลุ่ม รวมไปถึง การประเมินประสิทธิภาพของแบบจำลองภายในและภายนอก ในงานศึกษาวิจัยครั้งนี้ได้ใช้เครื่องมือวัดประสิทธิ์ภาพจากโปรแกรม Rapid Miner ซึ่งเลือกใช้เทคนิค ดังนี้

2.5.1 Area under the Curve (AUC) คือ การวัดประสิทธิภาพโดยการหาค่าพื้นที่ใต้โค้ง ซึ่งเป็นการหาพื้นที่ใต้โค้งหรือกราฟความสัมพันธ์ระหว่าง True positive rate (Sensitivity: ความไว) กับ False positive rate (1-Specificity: ความจำเพาะ) โดยที่ True positive rate คืออัตราส่วนของจำนวนข้อมูลที่สนใจมีผลเป็นบวกและผลการทำนายเป็นบวก (TP) ต่อจำนวนข้อมูลที่สนใจมีผลเป็นบวกทั้งหมด (TP+FN) และ False positive rate คืออัตราส่วนของจำนวนข้อมูลมีผลเป็นลบและผลการทำนายเป็นลบ (TN) ต่อจำนวนข้อมูลที่สนใจมีผลเป็นลบทั้งหมด (FP+TN)

ซึ่งเป็นการพิจารณาความไวและความจำเพาะไปพร้อม ๆ กัน โดยที่หากค่าความไวมีค่ามาก โอกาสที่จะได้ผล FN จะน้อยลง และหากค่าความจำเพาะมีค่ามาก โอกาสที่จะได้ผล FP จะน้อยลงเช่นกัน (เบญจพร เอี่ยมประโคน & ณัตติฤดี เจริญรักษ์, 2018)



ภาพประกอบ 5 แสดงกราฟความสัมพันธ์ระหว่าง TPR (Sensitivity) กับ FPR (1-Specificity)

ที่มา: Glen (2019)

ในการหา AUC เพื่อวัดประสิทธิภาพของแบบจำลองว่าสามารถทำนายค่าออกมาได้มีประสิทธิภาพเพียงใด จากภาพประกอบ 5 หากค่า AUC ได้ผลลัพธ์ออกมาคือ 100% คือแบบจำลองสามารถทำนายผลลัพธ์ออกมาได้ถูกต้องทั้งหมด แต่หากผลลัพธ์ของ AUC คือ 50% คือแบบจำลองสามารถทำนายได้ผลลัพธ์ที่ถูก ผิด เท่า ๆ กัน และหากค่า AUC ต่ำกว่า 50% แสดงให้เห็นว่าแบบจำลองทำนายได้ผลลัพธ์ที่ผิดเป็นส่วนใหญ่

2.5.2 Precision คือ การวัดประสิทธิภาพโดยการหาค่าความแม่นยำของแบบจำลอง คำนวณจากอัตราส่วนของจำนวนข้อมูลที่สนใจมีผลเป็นบวกและผลการทำนายเป็นบวก (TP) ต่อจำนวนผลการทำนายที่เป็นบวกทั้งหมด (TP+FP)

Precision@k คือการหาค่าความแม่นยำของแบบจำลอง ณ ผลลัพธ์ที่ k โดยไม่สนใจผลลัพธ์ในตำแหน่งอื่น ๆ ที่ตามมา เช่นหาค่าความแม่นยำ ณ ตำแหน่งที่ 6 ของผลลัพธ์ของแบบจำลองอันดับการแนะนำ ซึ่งได้จากการนำค่าทำนายคะแนนมาเรียงลำดับจากคะแนนมากที่สุด โดยพิจารณาควบคู่กับชุดข้อมูลที่จัดเก็บข้อมูลความเกี่ยวข้องกับผู้ใช้เอาไว้ ในการพิจารณาจะพิจารณาอันดับการแนะนำตั้งแต่อันดับที่ 6 ขึ้นไปกับชุดข้อมูลความเกี่ยวข้อง หากพบว่าใน 6 อันดับมีข้อมูลที่เกี่ยวข้องกับผู้ใช้ 3 ข้อมูล จะได้ค่าความเป็นยำคือ 3 หารด้วย 6 เท่ากับ 0.5 (ไกรศักดิ์ เกษร, 2558)

2.5.3 Normalized Discounted Cumulative Gain (NDCG) การประเมินผลนี้นิยมใช้ในการวัดประสิทธิภาพของอัลกอริทึมประเภทการเรียงลำดับ การสืบค้นเอกสารจากเว็บสืบค้น หรือแอปพลิเคชันที่เกี่ยวข้อง โดยกำหนดให้เกณฑ์คะแนน และให้ความสำคัญกับข้อมูลที่เกี่ยวข้องในอันดับต้น ๆ ค่า NDCG จะมีค่าอยู่ระหว่าง 0.0 ถึง 1.0 ซึ่งค่ายิ่งมากจะหมายถึงยิ่งเรียงลำดับได้ดี

โดยตัวชี้วัด nDCG นี้มีข้อดีกว่าการใช้ตัวชี้วัด Precision เนื่องจากมีการคำนึงถึงการจัดเรียงลำดับของผลค้นคืนที่ได้ด้วย กล่าวคือค่าประสิทธิผลของการค้นคืนจะถูกหักลดลงตามสัดส่วนของระดับความสำคัญของเอกสาร (Relevance Grade) ที่ปรากฏในการจัดลำดับการค้นคืน (พิจิตรา, 2017)

2.5.4 Mean Average Precision (MAP) คือการวัดประสิทธิภาพโดยการหาค่าเฉลี่ยความแม่นยำของข้อมูลที่เกี่ยวข้องในการจัดลำดับ โดยจะคำนวนหาค่าความแม่นยำของข้อมูลทุกตำแหน่งในรายการผลลัพธ์ หลังจากนั้นจึงนำมาหาค่าเฉลี่ย (ไกรศักดิ์ เกษร, 2558)

2.6 Rapid miner studio 9.6

Rapid miner studio 9.6 เป็นเครื่องมือที่ช่วยในการคำนวณวิเคราะห์ประมวลผลโดยมีโอเปอร์เรเตอร์ต่าง ๆ มีสามารถนำมาใช้ในขั้นตอนต่าง ๆ ตั้งแต่การเตรียมข้อมูล จนถึงขั้นตอนการสร้างแบบจำลองให้เป็นไปตามที่ผู้ใช้ต้องการ ซึ่งในงานวิจัยนี้ผู้ศึกษาวิจัยเลือกใช้ Rapid miner studio ที่เป็นประเภทสำหรับการศึกษา (Educational Program) ดังภาพประกอบ 6 เพื่อให้สามารถดำเนินการกับชุดข้อมูลที่มากกว่า 10,000 Rows ได้

ภาพประกอบ 6 รูปโปรแกรม Rapid miner studio 9.6

2.6.1 Read CSV

ภาพประกอบ 7 กล่องการใช้งาน Data Access : Read CSV

จากภาพประกอบ 7 แสดงภาพของโอเปอร์เรเตอร์ Read CSV ในโปรแกรม Rapid Miner ซึ่งเป็นตัวที่ใช้ในการ import ชุดข้อมูลสกุล.CSV (comma-separated values) เข้ามาในโปรแกรม Rapid miner studio โดยมีรายละเอียดดังนี้

Input

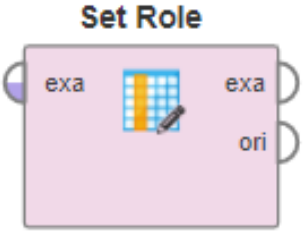
- File (ไฟล์) เอกสารหรือไฟล์นามสกุล .CSV

Output

- Output (ตารางข้อมูล)

พอร์ต (Port) นี้จะส่งออกชุดข้อมูล ExampleSet ที่สร้างจากไฟล์ CSV ที่นำเข้าจากพอร์ตอินพุต

2.6.2 Set Role



ภาพประกอบ 8 กล่องการใช้งาน Blending : Set Role

จากภาพประกอบ 8 แสดงภาพโอเปอร์เรเตอร์ Set Role ซึ่งเป็นโอเปอร์เรเตอร์ที่ใช้ในการกำหนดบทบาทของแอตทริบิวต์ เพื่ออธิบายว่าตัวดำเนินการอื่น ๆ จะดำเนินการกับแอตทริบิวต์นี้อย่างไร บทบาทเริ่มต้นเป็น Regular สำหรับบทบาทอื่น ๆ ถูกจัดประเภทเป็นบทบาทพิเศษ ซึ่งชุดข้อมูล ExampleSet สามารถมีแอตทริบิวต์บทบาทพิเศษได้หลายค่า แต่ละบทบาทพิเศษสามารถปรากฏได้เพียงครั้งเดียว หากมีการกำหนดบทบาทพิเศษให้มากกว่าหนึ่งแอตทริบิวต์ บทบาททั้งหมดจะเปลี่ยนเป็นปกติ ยกเว้นแอตทริบิวต์สุดท้าย การกำหนดบทบาทมีรายละเอียดดังนี้

Input

- Example set (ตารางข้อมูล)

พอร์ตอินพุตจะรับค่าชุดข้อมูล ExampleSet

Output

- Example set (ตารางข้อมูล)

พอร์ตเอาต์พุตจะส่งค่าชุดข้อมูล ExampleSet ที่กำหนดบทบาทเรียบร้อยแล้ว

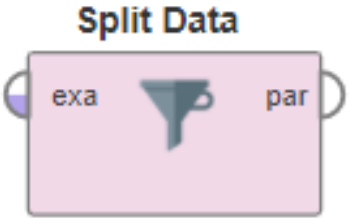
- Original (ตารางข้อมูล)

พอร์ตนี้จะส่งค่าชุดข้อมูลตั้งต้นที่ได้รับผ่านทางพอร์ตอินพุต โดยที่ไม่ได้ผ่านขั้นตอนการตั้งค่าบทบาทใด ๆ

Parameters

- attribute_name ชื่อของแอตทริบิวต์ที่ต้องการกำหนดบทบาท
- target_role บทบาทที่ต้องการกำหนดให้แก่แอตทริบิวต์
- set_additional_roles พารามิเตอร์นี้ใช้เพื่อตั้งค่าบทบาทแอตทริบิวต์มากกว่าหนึ่งแอตทริบิวต์พร้อมกัน

2.6.3 Split Data



ภาพประกอบ 9 กล่องการใช้งาน Blending : Split Data

จากภาพประกอบ 9 แสดงภาพโอเปอร์เรเตอร์ Split Data ซึ่งโอเปอร์เรเตอร์นี้จะสร้างชุดข้อมูลย่อยของชุดข้อมูลตั้งต้น เป็นจำนวนที่กำหนด โดยจะแบ่งชุดข้อมูลตั้งต้นออกเป็นชุดย่อยตามขนาดอัตราส่วน (Ratio) ที่ระบุ ซึ่งผลรวมของอัตราส่วนทั้งหมดควรมีค่าเท่ากับ 1 รายละเอียดเพิ่มเติมดังนี้

Input

- Example set (ตารางข้อมูล)

พอร์ตอินพุตจะรับค่าชุดข้อมูล

Output

- Partition (ตารางข้อมูล)

โอเปอร์เรเตอร์นี้สามารถมีพอร์ตเอาต์พุต หรือพาร์ติชันได้หลายพอร์ต จำนวนพอร์ตพาร์ติชันที่มีขึ้นอยู่กับจำนวนชุดข้อมูลย่อย หรือการกำหนดอัตราส่วน Ratio

Parameters

- partitions พารามิเตอร์ที่ใช้ในการกำหนดจำนวนพาร์ติชันและอัตราส่วนของแต่ละพาร์ติชัน โดยอัตราส่วนควรอยู่ระหว่าง 0 ถึง 1 ผลรวมของอัตราส่วนทั้งหมดควรเป็น 1

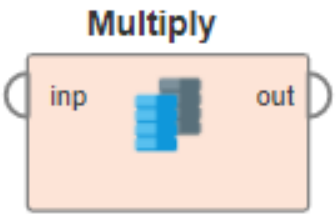
- sampling_type สามารถระบุประเภทหรือรูปแบบของการสุ่มในการสร้างชุดข้อมูลย่อย

- use_local_random_seed ใช้สำหรับการสุ่มตัวอย่างของชุดข้อมูลย่อย เนื่องจากเป็นการใช้ค่า local random seed จึงจะทำให้เกิดชุดย่อยเหมือนกัน พารามิเตอร์นี้จะใช้ได้เฉพาะเมื่อเลือกสุ่มตัวอย่างแบบสุ่ม (Shuffled) หรือแบบแบ่งชั้น (Stratified sampling) ไม่

สามารถใช้ได้กับการสุ่มตัวอย่างเชิงเส้น (Linear Sampling) เนื่องจากไม่ได้ใช้วิธีการสุ่ม แต่ตัวอย่างจะถูกเลือกตามลำดับ

- local_random_seed พารามิเตอร์นี้ใช้ระบุ local random seed หรือค่าที่ต้องการให้ผลสุ่มออกมาเป็นค่าเดิมตลอด พารามิเตอร์นี้จะใช้ได้เฉพาะเมื่อเลือกใช้พารามิเตอร์ use_local_random_seed

2.6.4 Multiply



ภาพประกอบ 10 กล่องการใช้งาน Utility : Multiply

จากภาพประกอบ 10 แสดงภาพโอเปอร์เรเตอร์ Multiply ซึ่งเป็นโอเปอร์เรเตอร์ที่จะนำค่าที่ได้รับจากพอร์ตอินพุตมาทำการคัดลอกเป็นสำเนา และส่งสำเนาไปยังพอร์ตเอาต์พุต พอร์ตที่เชื่อมต่อแต่ละพอร์ตจะสร้างสำเนาอิสระ ดังนั้นการเปลี่ยนสำเนาหนึ่งชุดจึงไม่มีผลกับสำเนาอื่น ๆ รายละเอียดมีดังนี้

Input

- Input (ตารางข้อมูล)

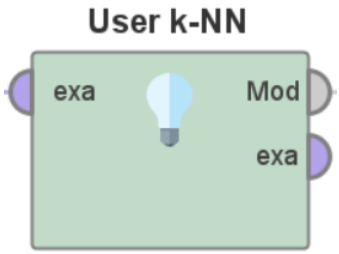
พอร์ตอินพุตจะรับค่าข้อมูลซึ่งเป็นค่าใดก็ได้ใน Rapid Minor

Output

- Output (ตารางข้อมูล)

สำเนาข้อมูลจากพอร์ตอินพุต ซึ่งสามารถมีได้หลายพอร์ตเอาต์พุต หลายสำเนา

2.6.5 User k-NN



ภาพประกอบ 11 กล่องการใช้งาน Extensions : User k-NN

จากภาพประกอบ 11 แสดงภาพโอเปอร์เรเตอร์ User k-NN ซึ่งเป็นโอเปอร์เรเตอร์ที่ใช้ในการสร้างแบบจำลอง k-Nearest Neighbor (k-NN) หรือเพื่อนบ้านที่ใกล้เคียงที่สุด โดยโอเปอร์เรเตอร์นี้จะสร้างแบบจำลอง k-NN ด้วยการวิธีการวิเคราะห์จากการกรองร่วมกัน (Collaborative Filtering) ซึ่งยึดตามความคล้ายคลึงของผู้ใช้ (User-base) โดยดำเนินการด้วยวิธีการ cosine similarity รายละเอียดดังนี้

Input

- example set (ตารางข้อมูล)

พอร์ตอินพุตจะรับค่าชุดข้อมูล ExampleSet

Output

- Model (แบบจำลอง)

พอร์ตเอาต์พุตให้ค่าแบบจำลอง k-NN

- example set (ตารางข้อมูล)

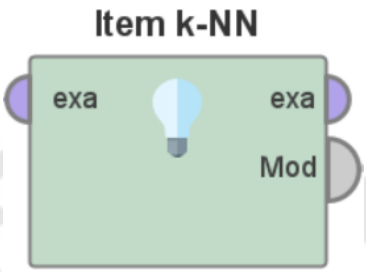
พอร์ตนี้จะส่งค่าชุดข้อมูล ExampleSet ตั้งต้นที่ได้รับผ่านทางพอร์ตอินพุต โดยที่ไม่ได้ผ่านขั้นตอนการตั้งค่าบทบาทใด ๆ

Parameters

- k คือการค้นหาข้อมูลจากชุดตัวอย่างที่ผ่านการฝึก (trainingExample) จำนวน k ที่ใกล้เคียงที่สุดกับชุดข้อมูลที่ไม่รู้จักเป็นขั้นตอนแรกของอัลกอริทึม k-NN ถ้า k = 1 ชุดตัวอย่างจะถูกกำหนดให้กับคลาสของเพื่อนบ้านที่ใกล้ที่สุด

- Weighted kNN หากตั้งค่าพารามิเตอร์นี้ค่าระยะห่างระหว่างชุดตัวอย่างจะถูกนำมาพิจารณาด้วย จะมีประโยชน์ในการถ่วงน้ำหนักของเพื่อนบ้าน เพื่อให้เพื่อนบ้านที่อยู่ใกล้มีค่าสนับสนุนมากกว่าเพื่อนบ้านที่อยู่ไกล

2.6.6 Item k-NN



ภาพประกอบ 12 กล่องการใช้งาน Extensions : Item k-NN

จากภาพประกอบ 12 โอเปอร์เรเตอร์ที่ใช้ในการสร้างแบบจำลอง k-Nearest Neighbor (k-NN) หรือเพื่อนบ้านที่ใกล้เคียงที่สุด โดยโอเปอร์เรเตอร์นี้จะสร้างแบบจำลอง k-NN ด้วยการวิธีการวิเคราะห์จากการกรองร่วมกัน (Collaborative Filtering) ซึ่งยึดตามความคล้ายคลึงของสิ่งของ (Item-base) โดยดำเนินการด้วยวิธีการ cosine similarity รายละเอียดดังนี้

Input

- example set (ตารางข้อมูล)

พอร์ตอินพุตจะรับค่าชุดข้อมูล ExampleSet

Output

- Model (แบบจำลอง)

พอร์ตเอาต์พุตให้ค่าแบบจำลอง k-NN

- example set (ตารางข้อมูล)

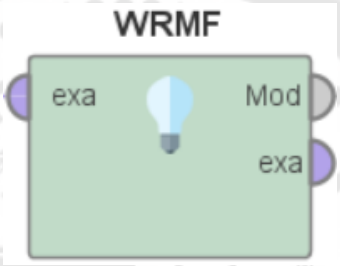
พอร์ตนี้จะส่งค่าชุดข้อมูล ExampleSet ตั้งต้นที่ได้รับผ่านทางพอร์ตอินพุต โดยที่ไม่ได้ผ่านขั้นตอนการตั้งค่าบทบาทใด ๆ

Parameters

- k คือการค้นหาข้อมูลชุดตัวอย่างที่ผ่านการฝึก (trainingExample) จำนวน k ที่ใกล้เคียงที่สุดกับชุดตัวอย่างที่ไม่รู้จักเป็นขั้นตอนแรกของอัลกอริทึม k-NN ถ้า k = 1 ชุดตัวอย่างจะถูกกำหนดให้กับคลาสของเพื่อนบ้านที่ใกล้ที่สุด

- Weighted kNN หากตั้งค่าพารามิเตอร์นี้ค่าระยะห่างระหว่างชุดตัวอย่างจะถูกนำมาพิจารณาด้วย จะมีประโยชน์ในการถ่วงน้ำหนักของเพื่อนบ้าน เพื่อให้เพื่อนบ้านที่อยู่ใกล้มีค่าสนับสนุนมากกว่าเพื่อนบ้านที่อยู่ไกล

2.6.7 Weighted Regularized Matrix Factorization (WRMF)



ภาพประกอบ 13 กล่องการใช้งาน Extensions : Weighted Regularized Matrix Factorization

จากภาพประกอบ 13 แสดงภาพโอเปอร์เรเตอร์ WRMF ซึ่งเป็นโอเปอร์เรเตอร์ที่ใช้ในการสร้างแบบจำลอง Weighted Regularized Matrix Factorization (WRMF) หรือการแยกตัวประกอบของแมทริกซ์แบบถ่วงน้ำหนักอย่างเป็นระบบ ด้วยการวิธีการวิเคราะห์จากการกรองร่วมกัน (Collaborative Filtering) รายละเอียดดังนี้

Input

- example set (ตารางข้อมูล)

พอร์ตอินพุตจะรับค่าชุดข้อมูล ExampleSet

Output

- Model (แบบจำลอง)

พอร์ตเอาต์พุตให้ค่าแบบจำลอง WRMF

- example set (ตารางข้อมูล)

พอร์ตนี้จะส่งค่าชุดข้อมูล ExampleSet ตั้งต้นที่ได้รับผ่านทางพอร์ตอินพุต โดยที่ไม่ได้ผ่านขั้นตอนการตั้งค่าบทบาทใด ๆ

Parameters

- Num Factors กำหนดค่าปัจจัยแฝง (Latent Factor) หรือค่าความชอบที่ผู้ใช้ (User) มีต่อสิ่งของ (Items)

- Regularization กำหนดค่าในการให้น้ำหนักและความสำคัญของข้อมูลที่จะใช้ในการสร้าง model

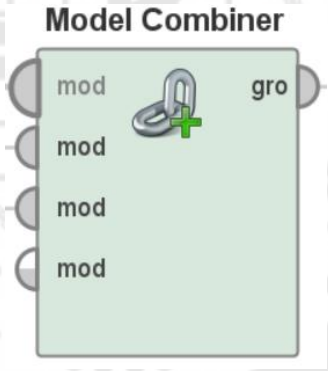
- C position กำหนดค่าน้ำหนัก หรือค่าความเชื่อมั่นที่เกิดจากการสังเกตเชิงบวก

- Iteration number กำหนดจำนวนการทำซ้ำ

- Initial mean กำหนดค่าเฉลี่ยเริ่มต้น

- Initial stdev กำหนดค่ามาตรฐานเริ่มต้น

2.6.8 Model Combiner



ภาพประกอบ 14 กล่องการใช้งาน Extensions : Model Combiner

จากภาพประกอบ 14 แสดงภาพโอเปอร์เรเตอร์ Model Combiner ซึ่งโอเปอร์เรเตอร์นี้จัดกลุ่มโมเดลอินพุตทั้งหมดเข้าด้วยกันเป็นโมเดลที่จัดกลุ่ม (รวมกัน) โมเดลนี้สามารถนำไปใช้กับข้อมูลใหม่ได้อย่างสมบูรณ์ ตัวโอเปอร์เรเตอร์จะส่งคืนคะแนนเฉลี่ยถ่วงน้ำหนักของโมเดลอินพุตแต่ละแบบ รายละเอียดดังนี้

Input

- Model (แบบจำลอง)

พอร์ตอินพุตสามารถรับค่าแบบจำลองได้หลายพอร์ต ซึ่งขั้นต่ำในการใช้โอเปอร์เรเตอร์นี้จะต้องมีอย่างน้อย 2 อินพุต

Output

- group of model (กลุ่มแบบจำลอง)

แบบจำลองที่กำหนดจะถูกจัดกลุ่มเป็นแบบจำลองรวมกัน และส่งค่ากลุ่มแบบจำลองออกมาที่เอาต์พุตพอร์ต

Parameters

- default weight ค่าน้ำหนักเริ่มต้นสำหรับทุกแบบจำลอง ที่ไม่ได้ระบุไว้ในรายการ model weights

- model weights ค่าน้ำหนักสำหรับหลายแบบจำลอง เกณฑ์น้ำหนักที่ไม่ได้กำหนดไว้ในรายการนี้ถูกตั้งค่าเป็น default weight

2.6.9 Apply Model

ภาพประกอบ 15 กล่องการใช้งาน Scoring : Apply Model

จากภาพประกอบ 15 แสดงภาพโอเปอร์เรเตอร์ Apply Model ซึ่งโอเปอร์เรเตอร์นี้นำไปใช้กับแบบจำลอง กับ ExampleSet โดยแบบจำลองจะมีข้อมูลที่ผ่านการเรียนรู้มาแล้ว ซึ่งข้อมูลนี้ใช้สำหรับแนะนำรายการใหม่ให้กับผู้ใช้ โดยการสร้างรายการแนะนำพร้อมอันดับ ซึ่งแอตทริบิวต์ของข้อมูลที่ผ่านการเรียนรู้เพื่อสร้างแบบจำลองนั้นจะต้องมีค่าตรงกันกับชุดข้อมูล ExampleSet รายละเอียดดังนี้

Input

- query set (ตารางข้อมูล)

พอร์ตอินพุตสำหรับรับค่า ExampleSet ที่มีค่าแอตทริบิวต์ของข้อมูลตรงกับแอตทริบิวต์ของข้อมูลที่ผ่านการเรียนรู้ของแบบจำลอง

- Model (แบบจำลอง)

พอร์ตอินพุตรับค่าแบบจำลอง ที่แอตทริบิวต์ของข้อมูลที่ผ่านการเรียนรู้เพื่อสร้างแบบจำลองนั้นมีค่าตรงกันกับชุดข้อมูล ExampleSet

Output

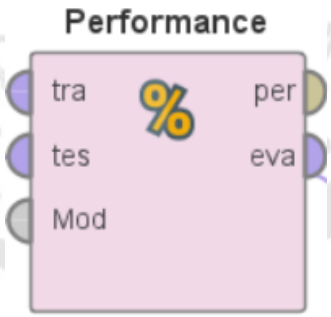
- result set (ตารางข้อมูล)

ExampleSet ที่ส่งมาจากพอร์ตนี้ถูกเปลี่ยนโดยใช้แบบจำลอง ในที่นี้จะเพิ่มค่ารายการแนะนำพร้อมการอันดับ

- Model (แบบจำลอง)

พอร์ตนี้จะส่งค่าแบบจำลองตั้งต้นที่ได้รับผ่านทางพอร์ตอินพุต โดยที่ไม่ได้ผ่านขั้นตอนการตั้งค่าบทบาทใด ๆ

2.6.10 Performance



ภาพประกอบ 16 กล่องการใช้งาน Validation : Performance

จากภาพประกอบ 16 แสดงภาพโอเปอร์เรเตอร์ Performance ซึ่งโอเปอร์เรเตอร์นี้ใช้สำหรับวัดประสิทธิภาพของแบบจำลองสำหรับการแนะนำรายการใหม่ โดยคำนวณพื้นที่ภายใต้เส้นโค้ง (AUC), ความแม่นยำที่ตำแหน่ง K (prec@k), Normalized discounted cumulative gain (NDCG) และค่าเฉลี่ยความแม่นยำเฉลี่ย (MAP)

2.7 งานวิจัยที่เกี่ยวข้อง

Mihelcic, Antulov-Fantulin, Bosnjak, and Smuc (2012)ได้นำเสนอการสร้างระบบการแนะนำด้วยส่วนขยาย Recommender ในซอฟต์แวร์ Rapid Miner โดยแสดงรูปแบบการสร้างแบบจำลองด้วยเทคนิค k-NN, Matrix Factorization และการรวมเทคนิคต่าง ๆ เข้าด้วยกัน ซึ่งในซอฟต์แวร์ RapidMiner สามารถดำเนินการได้ด้วยโอเปอร์เรเตอร์ที่ชื่อว่า Model Combiner ซึ่งสามารถให้น้ำหนักแก่แบบจำลองที่นำมารวมได้ ทั้งนี้ Mihelcic และคณะได้ทำการเปรียบเทียบประสิทธิภาพการทำงานของแบบจำลองด้วยเทคนิคต่าง ๆ โดยผลลัพธ์ประสิทธิภาพที่ได้จากการแนะนำสินค้าให้ผู้ใช้จำนวน 5000 คนนั้นพบว่าเทคนิคการรวมแบบจำลองให้ค่าประสิทธิภาพดีที่สุด

Tang and Wen (2015) ได้นำเสนอการสร้างแบบจำลองระบบแนะนำด้วยเทคนิค Collaborative Filtering โดยใช้ส่วนขยาย Recommender ของซอฟต์แวร์ RapidMiner ซึ่งได้อธิบายการสร้างแบบจำลองด้วยโอเปอร์เรเตอร์ต่าง ๆ ของซอฟต์แวร์ RapidMiner โดยเลือกใช้เทคนิค k-NN ในการศึกษาวิจัยนี้ ในการสร้างแบบจำลองนั้น โอเปอร์เรเตอร์ Collaborative Recommender จะใช้เมทริกซ์ระหว่างผู้ใช้และสินค้า (user-item) หรือเมทริกซ์ที่แสดงประวัติของผู้ใช้ต่อสินค้าในการสร้างแบบจำลอง โดยผลลัพธ์ที่ได้จากแบบจำลองนี้จะเป็นรายการแนะนำสินค้าแก่ผู้ใช้จำนวน n อันดับ ซึ่งเป็นรายการสินค้าใหม่ที่ไม่ได้อยู่ในประวัติของผู้ใช้

Jain and Vishwakarma (2017) ได้นำเสนอระบบการแนะนำหนังด้วยเทคนิค Collaborative Filtering โดยใช้ส่วนขยาย Recommender ของซอฟต์แวร์ RapidMiner ซึ่งประโยชน์ของการใช้ RapidMiner คือช่วยให้สามารถมองเห็นภาพรวมของระบบที่ง่ายและประมวลผลได้อย่างรวดเร็วในการเปรียบเทียบหาเทคนิคในการสร้างแบบจำลองระบบการแนะนำที่เหมาะสมกับชุดข้อมูลที่ที่สุด โดยงานวิจัยนี้วัดประสิทธิภาพของระบบผู้แนะนำโดยใช้แบบจำลองเดียว (Single Model) คือการสร้างแบบจำลองด้วยเทคนิค k-NN และการใช้แบบจำลองหลายแบบ (Multiple Models) คือการสร้างแบบจำลองด้วยเทคนิค k-NN ร่วมกับ Matrix Factorization ซึ่งผลลัพธ์ที่ได้พบว่าวิธีการใช้แบบจำลองหลายแบบสำหรับสร้างระบบการแนะนำจะให้ประสิทธิภาพที่ดีกว่าการใช้แบบจำลองเดียว

นุชรี เปรมชัยสวัสดิ์. (2020) ได้นำเสนอบทความเรื่องระบบผู้แนะนำแบบหลายเกณฑ์จากข้อมูลแบบไฮบริดเพื่อลดภาระของผู้ใช้ในการให้คะแนนความชื่นชอบที่มีต่อสิ่งของต่าง ๆ และลดปัญหาความไม่สอดคล้องกันของข้อมูลที่เกิดจากการรวบรวมจากผู้ใช้โดยตรง โดยเป็นการรวมกันของข้อมูลความชื่นชอบของผู้ใช้ต่อคุณลักษณะของสิ่งของร่วมกับข้อมูลคุณลักษณะต่าง ๆ

ของสิ่งของ เพื่อสร้างเป็นข้อมูลความชอบของผู้ใช้ต่อคุณลักษณะของสิ่งของต่าง ๆ ซึ่งมีประโยชน์ในการแนะนำสิ่งของให้กับผู้ใช้แต่ละคน และสำหรับการพัฒนาระบบผู้แนะนำแบบหลายเกณฑ์ที่เพิ่มความสามารถในการทำงานของระบบให้เพิ่มขึ้นจากการใช้ข้อมูลเดิมที่มีแค่เกณฑ์เดียว อีกทั้งยังลดปัญหาที่ผู้ใช้ไม่สะดวกที่จะให้คะแนนคุณลักษณะต่าง ๆ ของสิ่งของที่มีคุณลักษณะมากกว่าหนึ่งคุณลักษณะ และความไม่สอดคล้องของข้อมูลความชื่นชอบของผู้ใช้ต่อสิ่งของได้อีกด้วย

นภวรรณ ดุษฎีเวทกุล (2560) ได้นำเสนองานวิจัยเรื่องการปรับปรุงวิธีการกรองร่วมสำหรับระบบแนะนำด้วยข้อมูลความสัมพันธ์จากสื่อสังคมออนไลน์ โดยมีแนวคิดที่ว่าใช้รูปแบบความสัมพันธ์ระหว่างเพื่อนในสังคมออนไลน์มาปรับปรุงความแม่นยำในการแนะนำสินค้าจากวิธีการกรองร่วม ซึ่งปรับปรุงในส่วนของการหาความคล้ายคลึงระหว่างผู้ใช้ที่อยู่ในระบบ (User Similarity) โดยแบ่งออกเป็น 2 วิธี คือ 1) วิธีการหาความคล้ายคลึงแบบโคซายน์ ที่หาความคล้ายคลึงด้วยการเพิ่มข้อมูลของเพื่อนออนไลน์ที่มีความเป็นได้ว่าจะรู้จักกันในชีวิตจริง 2) การหาความคล้ายคลึงแบบโคซายน์ ที่หาความคล้ายคลึงด้วยการเพิ่มข้อมูลของเพื่อนออนไลน์ที่มีผู้ติดตามจำนวนมาก โดยผลการทดลองพบว่าการเพิ่มความสัมพันธ์ของเพื่อนในสังคมออนไลน์เข้าไปนั้น ทำให้สามารถแนะนำสินค้าได้แม่นยำว่าการใช้วิธีการกรองร่วมแบบเดิม

บทที่ 3
วิธีดำเนินการวิจัย

ขั้นตอนในการดำเนินงานวิจัยเริ่มต้นจากกรอบแนวคิดในการศึกษาหาเทคนิคการทำเหมืองข้อมูลที่ใช้ในการสร้างแบบจำลองการแนะนำหนังสือมีดังนี้

1. ศึกษารายละเอียดของชุดข้อมูล
2. จัดเตรียมข้อมูลที่จะนำมาวิเคราะห์
3. สร้างและวัดประสิทธิภาพของแบบจำลองการแนะนำ

3.1 ศึกษารายละเอียดของชุดข้อมูล

ชุดข้อมูล rating.csv ที่ใช้ในงานวิจัยนี้เป็นข้อมูลการให้คะแนน (Rating) หนังสือยอดนิยมจำนวนหนึ่งหมื่นเล่ม โดยสมมุติว่าค่าคะแนนที่ได้เป็นการให้คะแนนผ่านอินเทอร์เน็ต ซึ่งค่าการให้คะแนนมีตั้งแต่ 1 – 5 (Foxtrot, 2017) ดังตัวอย่างในภาพประกอบ 17 โดยประกอบไปด้วย 981,756 Rows และ 3 Attributes รายละเอียดของแต่ละ Attributes แสดงดังตางราง 3

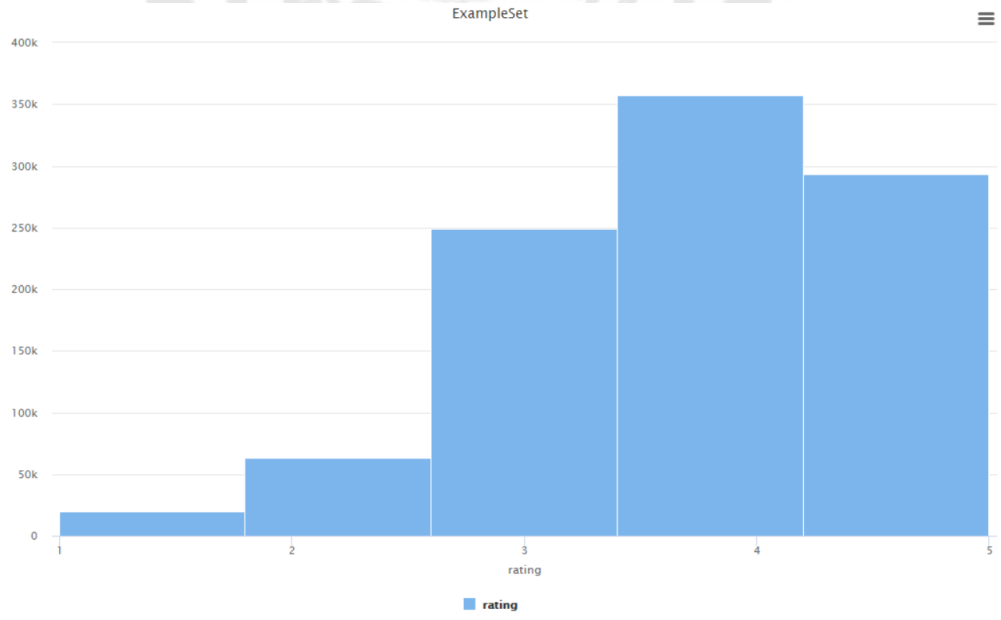
ตาราง 3 ตารางแสดงรายละเอียด Attribute ทั้งหมดของชุดข้อมูล rating.csv

Attribute	Descript	Type	Value
book_id	รหัสหนังสือ	Integer	ตั้งแต่ 1 – 10000
user_id	รหัสผู้ใช้	Integer	ตั้งแต่ 1 - 53424
rating	ค่าคะแนนที่ผู้ใช้ให้แก่หนังสือ	Integer	ตั้งแต่ 1- 5

	A	B	C
1	book_id	user_id	rating
2	1	314	5
3	1	439	3
4	1	588	5
5	1	1169	4
6	1	1185	4
7	1	2077	4
8	1	2487	4
9	1	2900	5

ภาพประกอบ 17 แสดง Attribute ของชุดข้อมูลในเอกสาร rating.csv

งานวิจัยฉบับนี้ต้องการทำแบบจำลองการแนะนำหนังสือ โดยเราจะทำนายจากชุดข้อมูลที่มีจำนวน 981,756 Rows ซึ่งเป็นข้อมูลการให้คะแนนหนังสือแต่ละเล่มของผู้ใช้ เมื่อเรานำชุดข้อมูลเข้ามาในโปรแกรม Rapid Miner Studio เรียบร้อยแล้ว ในขั้นตอนแรกคือการทำ Exploratory Data Analysis (EDA) หรือกระบวนการในการสำรวจตรวจสอบข้อมูลเบื้องต้น เพื่อให้เข้าใจว่าชุดข้อมูลที่นำมาใช้มากขึ้น โดยใช้วิธีการทำ Data visualization เพื่อแสดงข้อมูลในรูปแบบที่เข้าใจได้ง่ายมากยิ่งขึ้น จากภาพประกอบ 18 พบว่าการให้คะแนนของผู้ใช้งานส่วนใหญ่ให้คะแนนหนังสืออยู่ที่ 3 - 5 คะแนน และพบว่าไม่มีผู้ใช้ใดให้คะแนนหนังสือต่ำกว่า 1 คะแนน



ภาพประกอบ 18 กราฟแสดงจำนวนผู้ใช้งานที่ให้คะแนนหนังสือตั้งแต่ 1 – 5

จากนั้นดำเนินการนับ (Count) จำนวนครั้งในการให้คะแนนหนังสือของผู้ใช้แต่ละคนเพื่อหาจำนวนครั้งในการให้คะแนนที่มากที่สุด และน้อยที่สุด จากภาพประกอบ 19 พบว่าผู้ใช้งาน มีการให้คะแนนหนังสืออย่างน้อย 2 ครั้ง และจากภาพประกอบ 20 พบว่าผู้ใช้งานมีการให้คะแนนหนังสือมากที่สุดถึง 200 ครั้ง

user_id *Number*	count(rating) *Number*
43420	2
24765	2
20937	2
40540	2
6787	2
8272	2
15968	2
48816	2
48982	2
50435	2

53,424 rows - 2 columns (2 numerical)

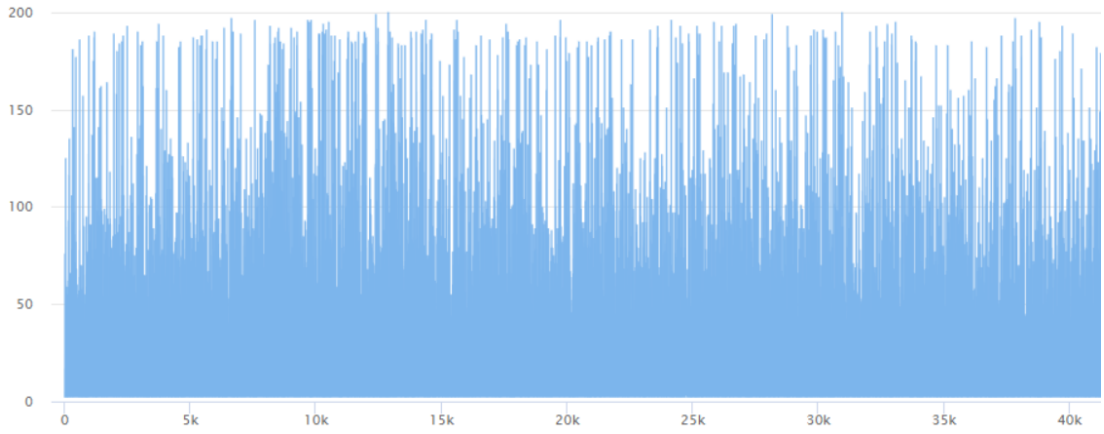
ภาพประกอบ 19 แสดงจำนวนครั้งจากน้อยไปมากในการให้คะแนนหนังสือของผู้ใช้แต่ละคน

user_id *Number*	count(rating) *Number*
12874	200
30944	200
28158	199
52036	199
12381	199
6630	197
37834	197
45554	197
7563	196
14372	196

53,424 rows - 2 columns (2 numerical)

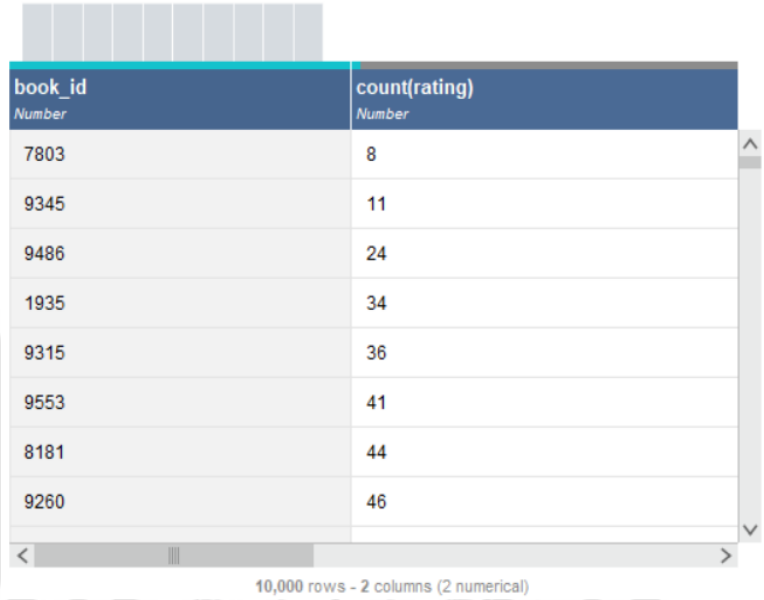
ภาพประกอบ 20 แสดงจำนวนครั้งจากมากไปน้อยในการให้คะแนนหนังสือของผู้ใช้แต่ละคน

และเมื่อนำข้อมูลไปสร้างเป็นกราฟเส้นดังแสดงในภาพประกอบ 21 จะเห็นว่ากราฟในช่วงจำนวนครั้งในการให้คะแนนหนังสือตั้งแต่ 100 ครั้งลงมานั้นมีความหนาแน่นมาก และในช่วงตั้งแต่ 100 ครั้งขึ้นไปมีความหนาแน่นน้อย ซึ่งหมายความว่าผู้ใช้ส่วนใหญ่จะให้คะแนนหนังสืออยู่ในช่วง 2 – 100 ครั้ง



ภาพประกอบ 21 แสดงจำนวนครั้งในการให้คะแนนหนังสือของผู้ใช้แต่ละคน

จากนั้นจึงตรวจสอบว่าหนังสือทั้งหมดมีการให้คะแนนจากผู้ใช้งานจำนวนสูงที่สุดกี่ครั้ง และจำนวนน้อยที่สุดกี่ครั้ง จากภาพประกอบ 22 พบว่าหนังสือที่ได้รับการให้คะแนนจากผู้ใช้งานน้อยที่สุดจำนวน 8 ครั้ง และจากภาพประกอบ 23 พบว่าหนังสือที่ได้รับการให้คะแนนจากผู้ใช้งานมากที่สุดมีจำนวน 100 ครั้ง



book_id *Number*	count(rating) *Number*
7803	8
9345	11
9486	24
1935	34
9315	36
9553	41
8181	44
9260	46

10,000 rows - 2 columns (2 numerical)

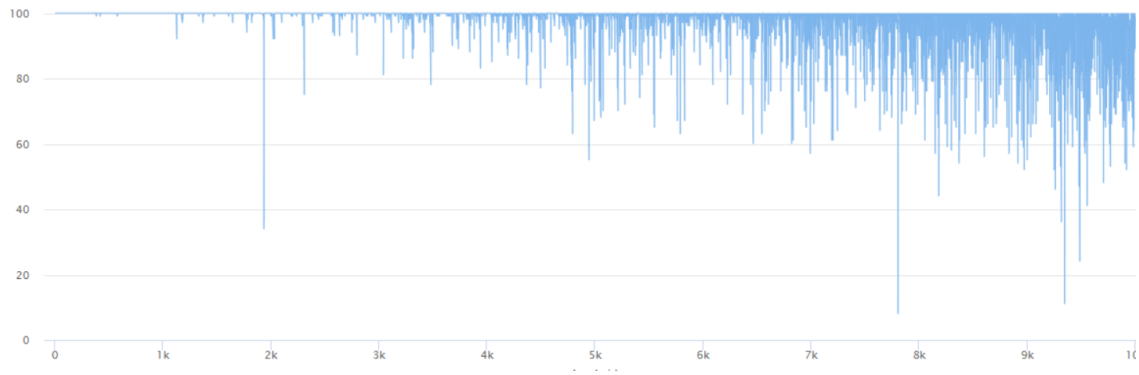
ภาพประกอบ 22 แสดงจำนวนครั้งที่ได้รับการให้คะแนนจากน้อยไปมากโดยผู้ใช้งานของหนังสือแต่ละเล่ม

book_id *Number*	count(rating) *Number*
1	100
2	100
3	100
4	100
5	100
6	100
7	100
8	100
9	100
10	100

10,000 rows - 2 columns (2 numerical)

ภาพประกอบ 23 แสดงจำนวนครั้งที่ได้รับการให้คะแนนจากมากไปน้อยโดยผู้ใช้งานของหนังสือแต่ละเล่ม

และเมื่อนำข้อมูลไปสร้างเป็นกราฟเส้นดังแสดงในภาพประกอบ 24 จะเห็นว่ากราฟจำนวนครั้งในการได้รับคะแนนของหนังสือในช่วง 60 – 100 นั้นมีความหนาแน่นค่อนข้างมาก จึงเป็นไปได้ว่าหนังสือส่วนใหญ่ได้รับการให้คะแนนจากผู้ใช้ประมาณ 60 – 100 ครั้ง



ภาพประกอบ 24 แสดงกราฟจำนวนครั้งที่ได้รับการให้คะแนนของผู้ใช้งานของหนังสือแต่ละเล่ม

3.2 จัดเตรียมข้อมูลที่จะนำไปวิเคราะห์

ในขั้นตอนการจัดเตรียมข้อมูลที่จะนำไปใช้วิเคราะห์นั้นสิ่งแรกที่จะทำคือการทำความสะอาดข้อมูลเพื่อให้ได้ข้อมูลที่ดีที่สุด ซึ่งจากภาพประกอบ 25 ได้ทำการตรวจสอบชุดข้อมูลพบว่ามีข้อมูลผู้ใช้ที่ให้คะแนนหนังสือเล่มเดิมมากกว่า 1 ครั้ง ซึ่งผู้ใช้ควรจะให้คะแนนหนังสือเพียง 1 ครั้งต่อ 1 เล่ม เราจึงจำเป็นต้องนำข้อมูลที่ซ้ำนี้ออก โดยข้อมูลซ้ำมีทั้งสิ้น 4,487 Rows

user_id *Number*	book_id *Number*	count(book_id) *Number*
3204	8946	5
4359	2515	4
691	6472	4
38259	3996	4
42	8946	4
34548	7420	4
5091	8946	4
2308	8946	4
27740	1208	3
48450	6189	3
30942	1036	3
202	6629	3

ภาพประกอบ 25 แสดงจำนวนครั้งในการให้คะแนนหนังสือแต่ละเล่มของผู้ใช้

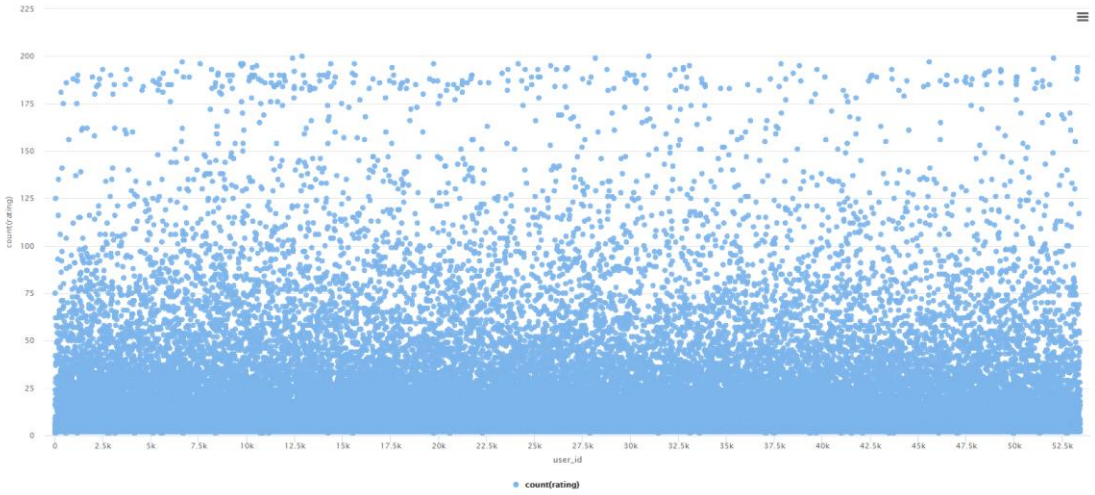
เมื่อดำเนินการลบข้อมูลซ้ำออกจากชุดข้อมูล โดยพิจารณาให้คงข้อมูลที่ผู้ใช้ให้คะแนนหนังสือนั้น ๆ สูงที่สุดเพียง 1 Rows เรียบร้อยแล้ว ชุดข้อมูลจะเหลือทั้งสิ้น 979,478 Rows ดังภาพประกอบ 26

book_id *Number*	user_id *Number*	rating *Number*
1	314	5
1	439	3
1	588	5
1	1169	4
1	1185	4
1	2077	4
1	2487	4
1	2900	5
1	3662	4
1	3922	5
1	5379	5
1	5461	3
1	5885	5
1	6630	5
1	7563	3
1	9246	1
1	10140	4
1	10146	5
1	10246	4

979,478 rows - 3 columns (3 numerical)

ภาพประกอบ 26 แสดงชุดข้อมูลหลังจากดำเนินการลบข้อมูลที่ซ้ำเรียบร้อยแล้ว

จากนั้นเพื่อให้ชุดข้อมูลที่ใช้ในการทำนายมีความชัดเจนมากยิ่งขึ้น ผู้ศึกษาวิจัยจึงกรอง (Filter) ชุดข้อมูล โดยการพิจารณาการกระจายของข้อมูลการให้คะแนนของผู้ใช้แต่ละคน จากภาพประกอบ 27 จะพบว่าข้อมูลการให้คะแนนหนังสือของผู้ใช้แต่ละคนมีจำนวนที่ค่อนข้างกระจายกัน และเนื่องจากการสร้างแบบจำลองการแนะนำโดยใช้วิธีการพิจารณาการกรองร่วม (Collaborative filtering) นั้นจะพิจารณาจากประวัติความชอบต่อสิ่งของนั้น ๆ ของผู้ใช้ โดยยิ่งผู้ใช้มีข้อมูลการให้คะแนนหนังสือมากจะยิ่งดี ผู้ศึกษาวิจัยจึงเลือกกรองข้อมูลของผู้ใช้งานที่มีการให้คะแนนหนังสือตั้งแต่ 20 เล่ม



ภาพประกอบ 27 กราฟการกระจายของจำนวนครั้งในการให้คะแนนหนังสือของผู้ใช้แต่ละคน

หลังจากดำเนินการกรองข้อมูลเรียบร้อยแล้ว ชุดข้อมูลที่จะนำไปใช้ในการสร้างแบบจำลองการแนะนำหนังสือจำนวนทั้งสิ้น 720,212 Rows โดยเป็นข้อมูล user_id จำนวน 14,612 คน และข้อมูล book_id จำนวน 9,999 เล่ม

3.3 สร้างและวัดประสิทธิภาพของแบบจำลองการแนะนำ

ในการสร้างแบบจำลองการแนะนำ เนื่องจากชุดข้อมูลที่นำมาใช้ในการสร้างแบบจำลองนั้นเป็นชุดข้อมูลของการให้คะแนนหนังสือจากผู้ใช้งาน โดยเป็นการให้คะแนนที่มีความชัดเจน ซึ่งจากการศึกษางานวิจัยพบว่าระบบแนะนำส่วนใหญ่ที่นำคะแนนมาพิจารณาในการหาสินค้าที่คาดว่าผู้ใช้จะสนใจนั้น คือเทคนิคการพิจารณาการกรองร่วม (Collaborative Filtering) และเทคนิคที่ส่วนใหญ่นำมาใช้คือ k-Nearest Neighbor (k-NN) และ Matrix Factorization

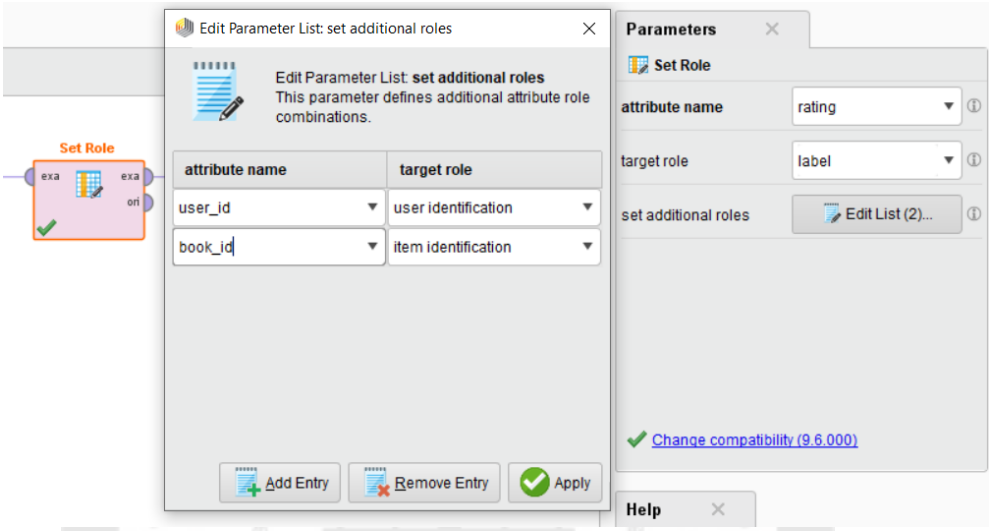
ดังนั้นในงานวิจัยนี้จะสร้างแบบจำลองการแนะนำด้วยเทคนิค k-Nearest Neighbor (k-NN), Matrix Factorization และ Model Combiner (k-NN + MF) และทำการเปรียบเทียบประสิทธิภาพของเทคนิคข้างต้น โดยใช้ซอฟต์แวร์ Rapid Miner Studio 9.6

เมื่อนำเข้าชุดข้อมูล rating.csv ผ่านโอเปอร์เรเตอร์ ReadCSV เรียบร้อยแล้ว จึงกำหนดบทบาทของแต่ละ Attribute โดยใช้โอเปอร์เรเตอร์ Set Role ระบุให้ Attribute ต่าง ๆ มีบทบาทดังภาพประกอบ 28 โดยมีรายละเอียดดังนี้

- ratings ให้มีบทบาทเป็น label เพื่อเป็น attribute เป้าหมายที่ต้องการนำไปใช้ในการเรียนรู้

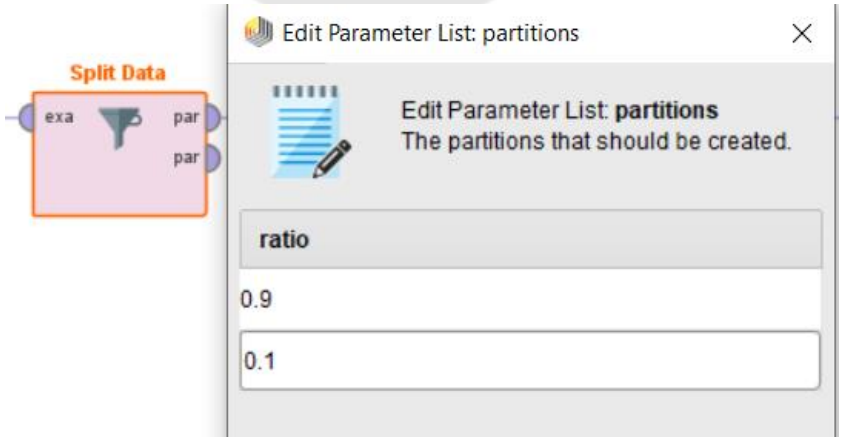
- user_id ให้มีบทบาทเป็น user identification เพื่อเป็นการระบุถึงผู้ใช้งาน

- book_id ให้มีบทบาทเป็น item identification เพื่อระบุถึงสิ่งของ



ภาพประกอบ 28 การตั้งค่า Parameter ภายใน Operator Set Role

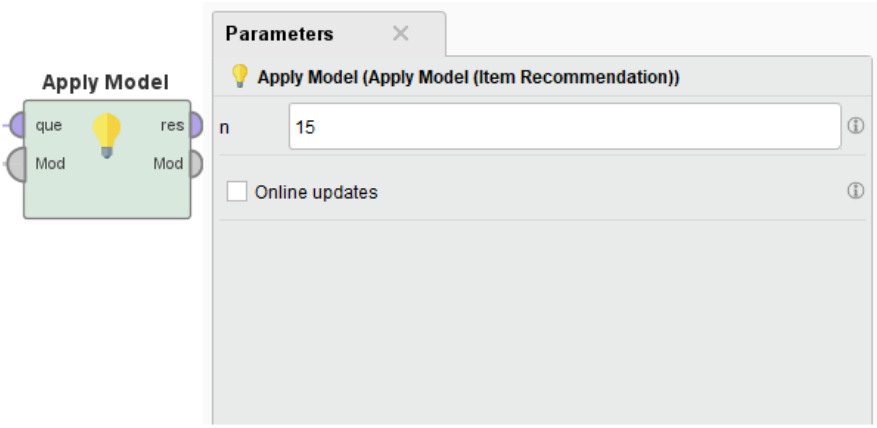
หลังจากนั้นจึงแบ่งชุดข้อมูลออกเป็นข้อมูลเรียนรู้ (Train data) และข้อมูลทดสอบ (Test data) ผ่านโอเปอร์เรเตอร์ Split Data ดังภาพประกอบ 29



ภาพประกอบ 29 Split ชุดข้อมูลเป็น Train data และ Test data

หลังจากดำเนินการเตรียมข้อมูลเรียบร้อยแล้วจึงเข้าสู่กระบวนการสร้างแบบจำลอง และการวัดประสิทธิภาพการทำงานของแบบจำลอง

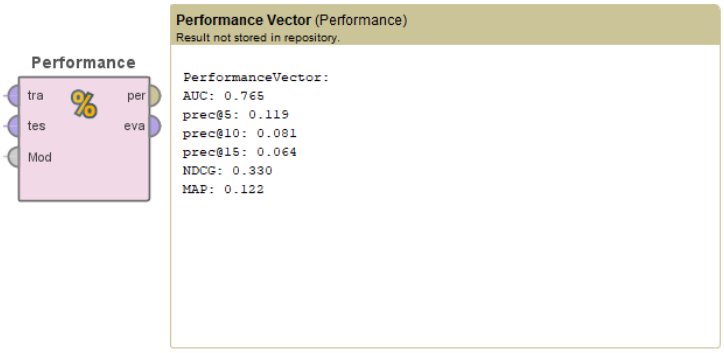
โดยโอเปอร์เรเตอร์ที่ผู้ศึกษาวิจัยเลือกใช้นั้นจะเป็นโอเปอร์เรเตอร์ส่วนขยาย Recommenders ซึ่งจะอยู่ในหัวข้อ Collaborative Filtering Item Recommendation โดยต่อชุดข้อมูลเรียนรู้เข้าสู่โอเปอร์เรเตอร์ที่เป็นอัลกอริทึมที่เราต้องการสร้างแบบจำลอง เพื่อให้อัลกอริทึมเรียนรู้ชุดข้อมูล จึงจะได้แบบจำลองการแนะนำออกมา จากนั้นต่อแบบจำลองเข้าสู่โอเปอร์เรเตอร์ Apply Model พร้อมกับข้อมูลทดสอบ เพื่อให้โปรแกรมดำเนินการนำแบบจำลองที่ได้มาใช้กับชุดข้อมูลทดสอบ แล้วได้ผลลัพธ์คือหนังสือ 15 อันดับที่แบบจำลองแนะนำให้แก่ผู้ใช้แต่ละคนในข้อมูลทดสอบ แสดงดังภาพประกอบ 30 โดยเป็นการแนะนำหนังสือเล่มใหม่ที่ผู้ใช้คนนั้น ๆ ไม่เคยให้คะแนนมาก่อน



ภาพประกอบ 30 กำหนดจำนวนอันดับการแนะนำสำหรับผู้ใช้แต่ละคนใน Test data

ในการวัดประสิทธิภาพการทำงานของแบบจำลองจะดำเนินการโดยการต่อชุดข้อมูล Train data, Test data และแบบจำลองเข้าสู่โอเปอร์เรเตอร์ Performance (Item Recommendation) ซึ่งจะอยู่ในหัวข้อ Performance Evaluation ภายใต้ส่วนขยาย Recommender เช่นเดียวกัน จากภาพประกอบ 31 นั้นโปรแกรมจะให้ผลลัพธ์เป็นค่าประสิทธิภาพต่าง ๆ ของแบบจำลอง คือ Area Under The Curve (AUC), Precision at 5 (prec@5), Precision

at 10 (prec@10), Precision at 15 (prec@15), Normalized Discounted Cumulative Gain (NDCG) และ Mean Average Precision (MAP)

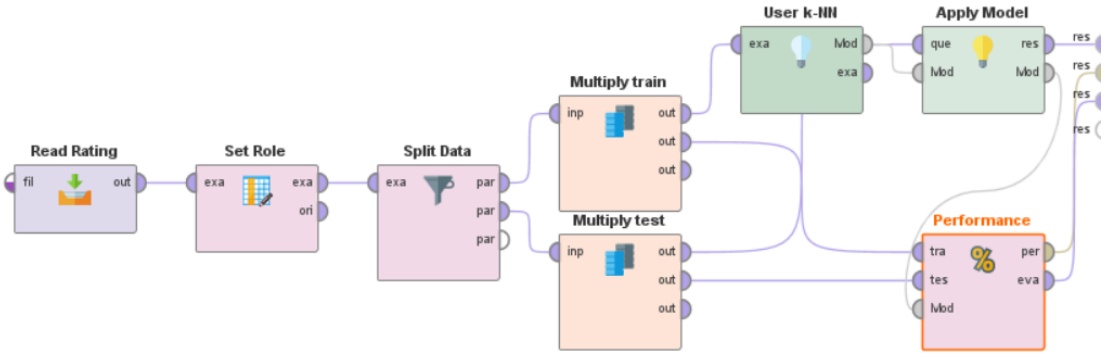


ภาพประกอบ 31 แสดงผลลัพธ์ของโอเปอร์เรเตอร์ Performance ที่ใช้ในการวัดประสิทธิภาพโมเดลการแนะนำ

รูปแบบการเชื่อมต่อของโอเปอร์เรเตอร์ต่าง ๆ ในซอฟต์แวร์ Rapid Miner ที่ใช้สำหรับการสร้างแบบจำลองของอัลกอริทึมที่ใช้ในการศึกษาวิจัย แสดงดังนี้

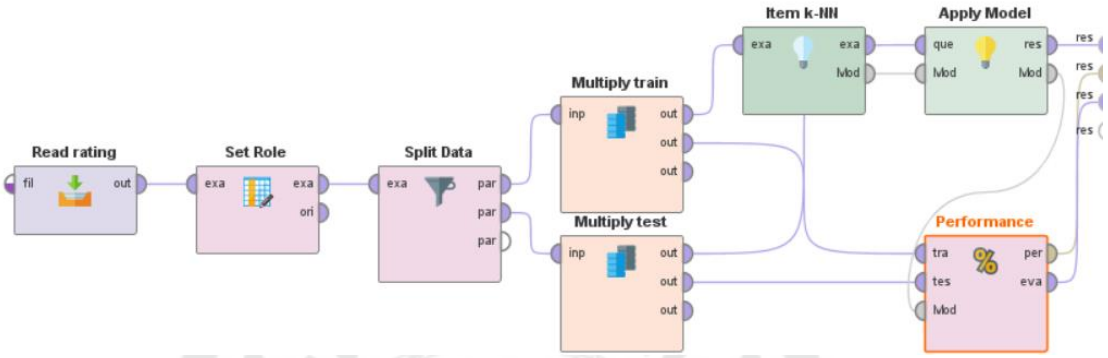
3.3.1 สร้างแบบจำลองแนะนำด้วยเทคนิค k-Nearest Neighbor (k-NN) โดยแบ่งออกเป็น 2 รูปแบบดังนี้

1) User-base ซึ่งจะพิจารณาจากผู้ใช้ในระบบที่ให้คะแนนความชื่นชอบต่อหนังสือในลักษณะคล้ายคลึงกัน (Bosnjak et al., 2011), (Wen, 2015) โดยโอเปอร์เรเตอร์ต่าง ๆ ที่ใช้สร้างแบบจำลองดังภาพประกอบ 32



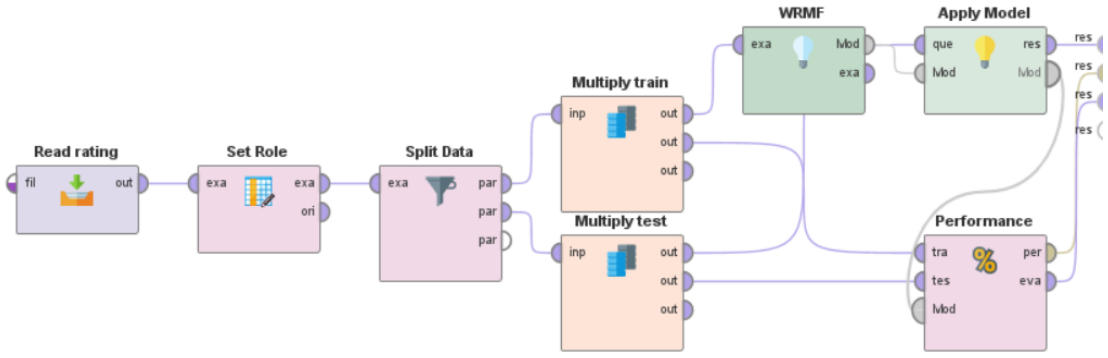
ภาพประกอบ 32 แบบจำลองด้วยเทคนิค User-base k-NN ที่สร้างบน Rapid Miner Studio 9.6

2) Item-base จะพิจารณาจากหนังสือที่ได้รับคะแนนความชื่นชอบจากผู้ใช้หลายคนไปทิศทางเดียวกัน (Bosnjak et al., 2011), (Wen, 2015) โดยโอเปอร์เรเตอร์ต่าง ๆ ที่ใช้สร้างแบบจำลองดังภาพประกอบ 33



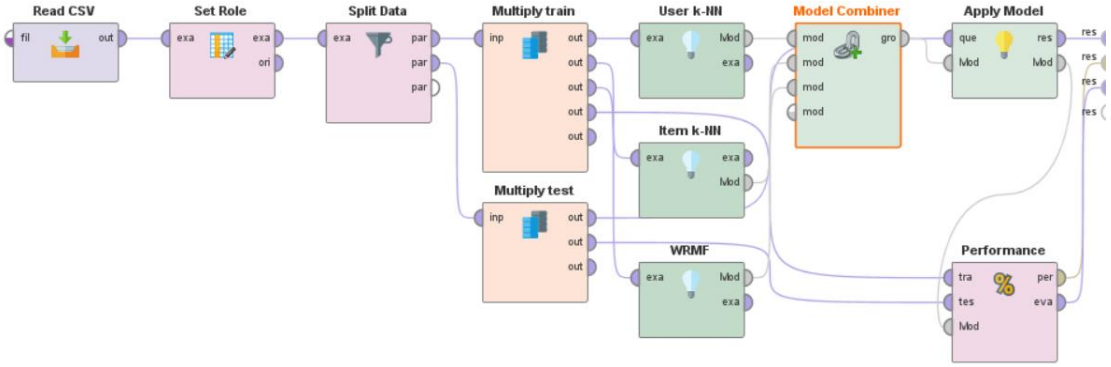
ภาพประกอบ 33 แบบจำลองด้วยเทคนิค Item-base k-NN ที่สร้างบน Rapid Miner Studio 9.6

3.3.2 สร้างแบบจำลองแนะนำด้วยเทคนิค Matrix Factorization (Mihelcic et al., 2012) โดยโอเปอร์เรเตอร์ต่าง ๆ ที่ใช้สร้างแบบจำลองดังภาพประกอบ 34



ภาพประกอบ 34 แบบจำลองด้วยเทคนิค Matrix Factorization ที่สร้างบน Rapid Miner Studio 9.6

3.3.3 Model Combiner โดยการรวมแบบจำลอง User-base k-NN, Item-base k-NN และ Matrix Factorization (Jain & Vishwakarma, 2017), (Mihelcic et al., 2012) โดยโอเปอร์เรเตอร์ต่าง ๆ ที่ใช้สร้างแบบจำลองดังภาพประกอบ 35



ภาพประกอบ 35 แบบจำลองด้วยเทคนิค Model combiner ที่สร้างบน Rapid Miner Studio 9.6

บทที่ 4
ผลการศึกษา

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค k-Nearest Neighbor (k-NN), Matrix Factorization และ Model Combiner (k-NN + MF) และทำการเปรียบเทียบประสิทธิภาพของเทคนิคข้างต้น โดยผู้วิจัยได้ทำการวิเคราะห์ข้อมูล และนำเสนอผลการศึกษาดังนี้

1. ผลการศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค k-Nearest Neighbor (k-NN)
2. ผลการศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค Matrix Factorization
3. ผลการศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค Model Combiner (k-NN + MF)
4. ผลการเปรียบเทียบประสิทธิภาพของแบบจำลองด้วยเทคนิคต่าง ๆ

4.1 ผลการศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค k-Nearest Neighbor (k-NN)

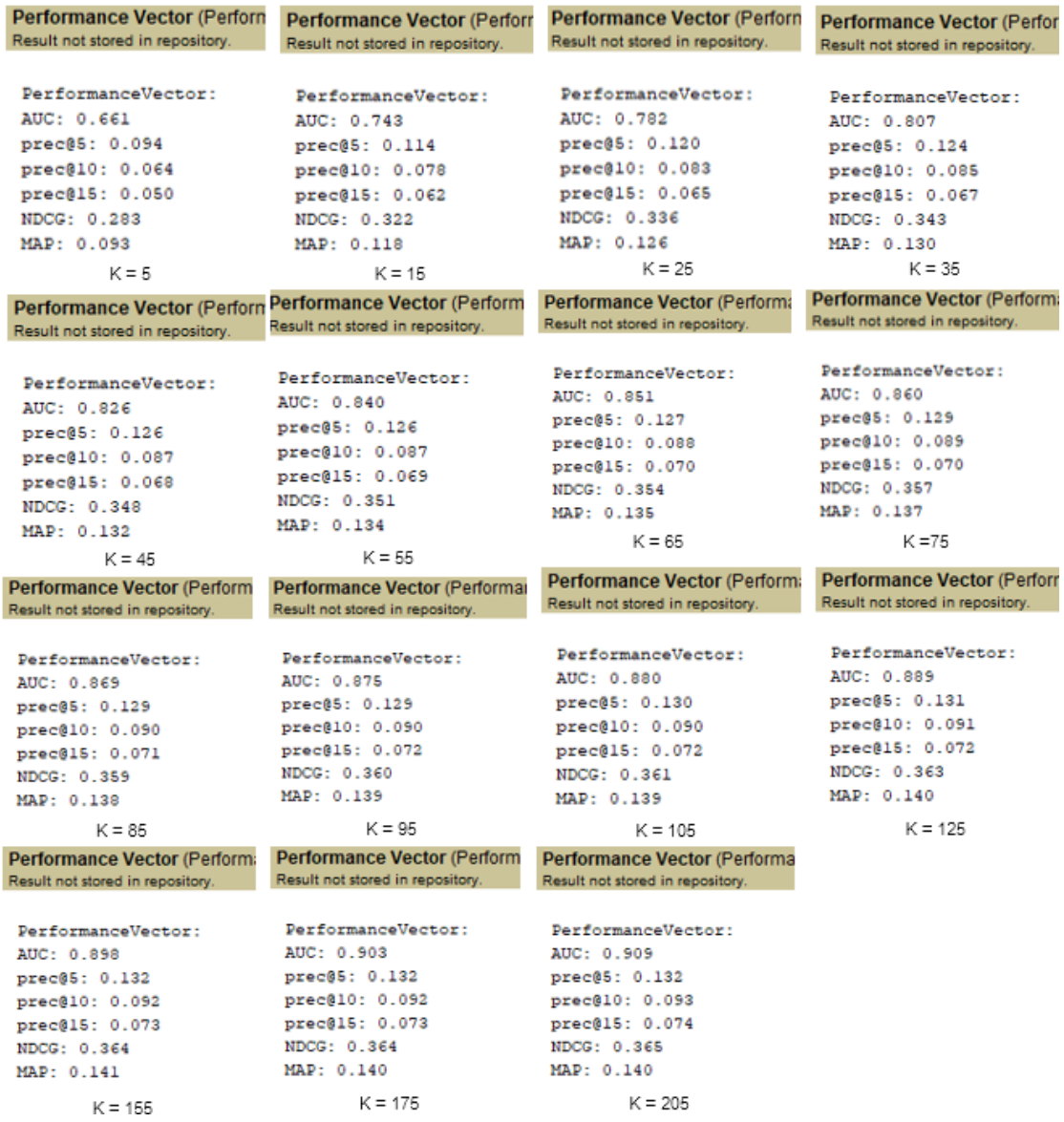
k-Nearest Neighbors (k-NN) หรือวิธีการเพื่อนบ้านใกล้ที่สุด เป็นอัลกอลิทึมที่ใช้ในการจัดกลุ่มของข้อมูล โดยพิจารณาจำนวน k ที่อยู่ใกล้ที่สุด ซึ่งมีความคล้ายคลึงกับสิ่งที่ต้องการพิจารณา

โดยซอฟต์แวร์ Rapid Miner นั้น โอเปอร์เรเตอร์ k-NN จะใช้วิธีการ Cosine Similarity ในการคำนวณหาค่าความคล้ายคลึง โดยกำหนดให้มุมระหว่างตัวแปรที่พิจารณามีค่าเป็น 0 คือตัวแปรทั้งสองมีความคล้ายคลึงกัน และหากมุมระหว่างตัวแปรทั้งสองมีค่าเท่ากับ 90 หมายถึงตัวแปรทั้งสองเป็นอิสระต่อกัน หรือไม่มีความคล้ายคลึงกันเลย (Bosnjak et al., 2011), (Wen, 2015)

เมื่อนำเข้าชุดข้อมูล rating.csv เข้าสู่โปรแกรมเรียบร้อยแล้วนั้น จะได้จำนวน Rows ทั้งหมดของชุดข้อมูล 720,212 Rows โดยประกอบไปด้วยข้อมูลผู้ใช้จำนวน 14,612 คน ข้อมูลหนังสือจำนวน 9,999 เล่ม และข้อมูลการให้คะแนนหนังสือที่มีค่าตั้งแต่ 1 – 5 คะแนน

ในการศึกษาวิจัยนี้ ผู้ศึกษาวิจัยได้ทำการสุ่มหาค่า k ที่ทำให้แบบจำลองให้ผลลัพธ์ที่ดีที่สุด โดยที่เวลาในการประมวลผลมีความเหมาะสม โดยสุ่มค่า k เป็น 5, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, 125, 155, 175 และ 205 และกำหนดสัดส่วนในการแบ่งชุดข้อมูลออกเป็น train data และ test data จำนวน 3 รูปแบบ คือ สัดส่วน 90:10, 80:20 และ 75:25

1) User-base จะเป็นการพิจารณาจากข้อมูลของผู้ใช้ในระบบที่ให้คะแนนความชื่นชอบต่อหนังสือที่มีความคล้ายคลึงกัน โดยผลลัพธ์ที่ได้จากการประมวลผลแบบจำลองด้วยการแบ่งชุดข้อมูลด้วยอัตราส่วน 90:10 แสดงดังภาพประกอบ 36



ภาพประกอบ 36 แสดงผลลัพธ์จากการทดสอบสุ่มหาค่า k ที่ดีที่สุดของเทคนิค User k-NN ในโปรแกรม Rapid Miner ของการแบ่งข้อมูล 90:10

จากตาราง 4 พบว่ายิ่งค่า k มีจำนวนมากขึ้น ค่าประสิทธิภาพของแบบจำลองหรือ AUC จะยิ่งเพิ่มขึ้น ในขณะที่ค่าความแม่นยำที่ตำแหน่งที่ 5 หรือ prec@5 ที่มีแนวโน้มเพิ่มขึ้นเมื่อจำนวน

k มากขึ้น แต่เมื่อดูจากตาราง 4 จะพบว่าเมื่อเข้าสู่ช่วง Scenario6 หรือกำหนดค่า k = 55 ผลลัพธ์ที่จะได้จะเริ่มให้ค่าที่ซ้ำกับ Scenario ก่อนหน้า เช่นเดียวกันกับผลลัพธ์ของค่าความแม่นยำ ณ ตำแหน่งที่ 10 ที่เริ่มให้ค่าซ้ำตั้งแต่ Scenario6 และค่าความแม่นยำ ณ ตำแหน่งที่ 15 จะเริ่มให้ค่าซ้ำตั้งแต่ Scenario8 สำหรับค่า NDCG ได้ผลลัพธ์ที่เพิ่มขึ้นเรื่อย ๆ จนถึง Scenario14 ที่ให้ค่าซ้ำและสุดท้าย ค่าความแม่นยำเฉลี่ย หรือ MAP ที่ได้ผลลัพธ์เพิ่มขึ้นเรื่อย ๆ จนถึง Scenario14 ที่ให้ผลลัพธ์น้อยกว่าค่าใน Scenario13 ผู้ศึกษาวิจัยจึงหยุดสุ่มค่า k

หากดูจากผลลัพธ์ประสิทธิภาพของแบบจำลองที่ได้ของทุกการสุ่มค่า k นั้น จะได้ว่า scenario13 หรือสุ่มค่า k = 155 แบบจำลองให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด

ตาราง 4 แสดงผลลัพธ์จากการทดสอบสุ่มหาค่า k ที่ดีที่สุด ของเทคนิค User k-NN ด้วยการแบ่งข้อมูลเป็นอัตราส่วน 90:10

		AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.661	0.094	0.064	0.05	0.283	0.093	1.53
Scenario2	15	0.743	0.114	0.078	0.062	0.322	0.118	3.27
Scenario3	25	0.782	0.12	0.083	0.065	0.336	0.126	5.10
Scenario4	35	0.807	0.124	0.085	0.067	0.343	0.13	6.57
Scenario5	45	0.826	0.126	0.087	0.068	0.348	0.132	8.50
Scenario6	55	0.84	0.126	0.087	0.069	0.351	0.134	10.34
Scenario7	65	0.851	0.127	0.088	0.07	0.354	0.135	12.13
Scenario8	75	0.86	0.129	0.089	0.07	0.357	0.137	14.22
Scenario9	85	0.869	0.129	0.09	0.071	0.359	0.138	16.13
Scenario10	95	0.875	0.129	0.09	0.072	0.36	0.139	18.18
Scenario11	105	0.88	0.13	0.09	0.072	0.361	0.139	20.22
Scenario12	125	0.889	0.131	0.091	0.072	0.363	0.14	23.55
Scenario13	155	0.898	0.132	0.092	0.073	0.364	0.141	29.55
Scenario14	175	0.903	0.132	0.092	0.073	0.364	0.14	33.26
Scenario15	205	0.909	0.132	0.093	0.074	0.365	0.14	39.52

จากนั้นดำเนินการเปลี่ยนแปลงอัตราส่วนในการแบ่งชุดข้อมูล train data และ test data เป็น 80:20 โดยกำหนดให้การสุ่มค่า k ใช้ค่าเดียวกันกับการทดลองแบ่งชุดข้อมูลด้วยอัตราส่วน 90:10 ผลลัพธ์ที่ได้จากโปรแกรม Rapid Miner แสดงดังภาพประกอบ 37

ภาพประกอบ 37 แสดงผลลัพธ์จากการทดสอบสุ่มหาค่า k ที่ดีที่สุดของเทคนิค User k-NN ในโปรแกรม Rapid Miner ของการแบ่งข้อมูล 80:20

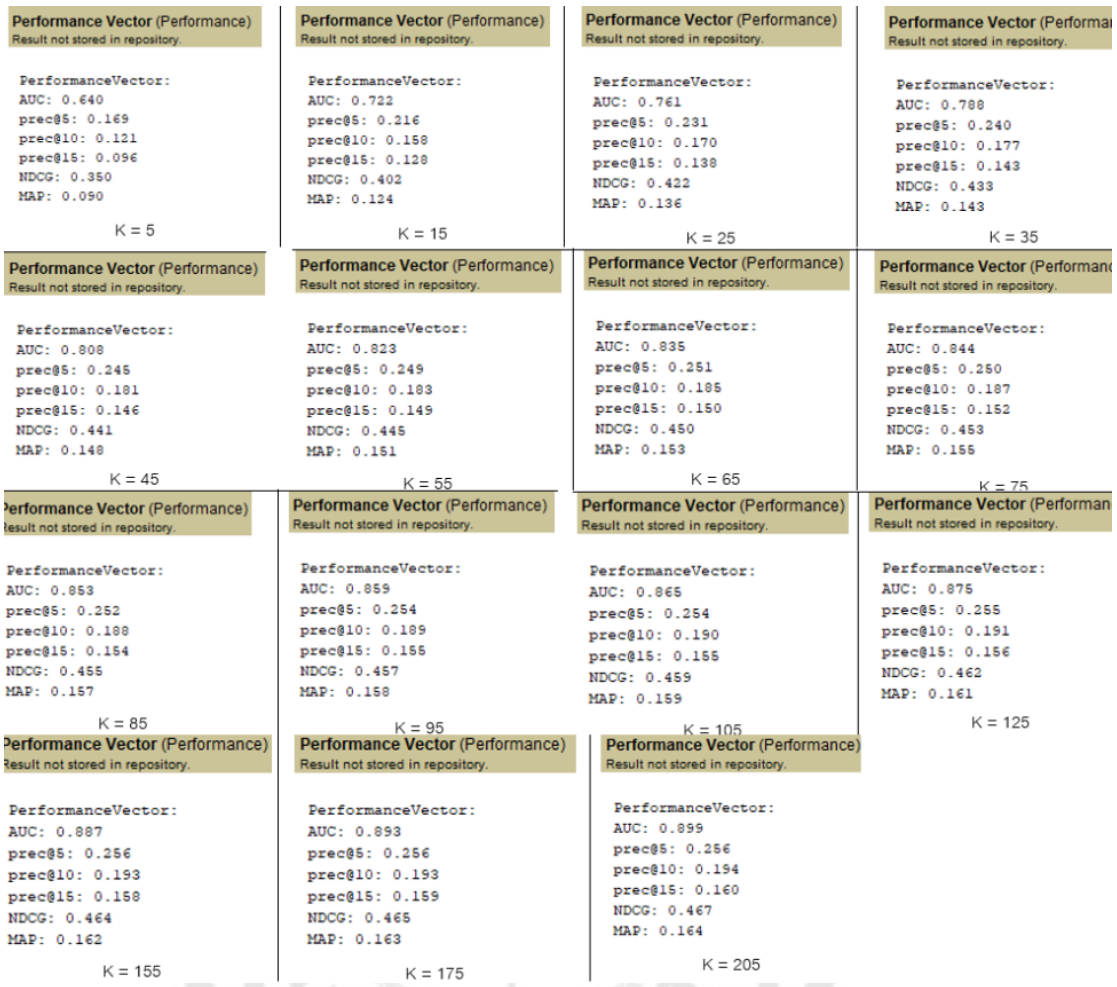
จากตาราง 5 พบว่าผลลัพธ์ที่ได้มีทิศทางเช่นเดียวกันกับการทดลองด้วยการแบ่งชุดข้อมูลด้วยอัตราส่วน 80:20 คือ ยิ่งค่า k มีจำนวนมากขึ้น ค่าประสิทธิภาพของแบบจำลองหรือ AUC จะยิ่งเพิ่มขึ้น ในขณะที่ค่าความแม่นยำที่ตำแหน่ง 5 หรือ prec@5 ที่มีแนวโน้มเพิ่มขึ้นเมื่อจำนวน k มากขึ้น แต่เมื่อดูจากตาราง 5 จะพบว่าเมื่อเข้าสู่ช่วง Scenario11 หรือกำหนดค่า k = 105 ผลลัพธ์ที่จะได้จะเริ่มให้ค่าที่ซ้ำกับ Scenario ก่อนหน้า และเมื่อเข้าสู่ Scenario15 ผลลัพธ์ที่ได้นั้นมีค่าน้อยกว่า Scenario ก่อนหน้า และผลลัพธ์ของค่าความแม่นยำ ณ ตำแหน่งที่ 10 ที่เริ่มให้ค่าซ้ำตั้งแต่ Scenario9 และค่าความแม่นยำ ณ ตำแหน่งที่ 15 นั้นจะให้ผลลัพธ์ที่เพิ่มขึ้นเรื่อย ๆ

จะเริ่มให้ค่าซ้ำใน Scenario15 สำหรับค่า NDCG ได้ผลลัพธ์ที่เพิ่มขึ้นเรื่อย ๆ และสุดท้าย ค่าความแม่นยำเฉลี่ย หรือ MAP ที่ได้ผลลัพธ์เพิ่มขึ้นเรื่อย ๆ จนถึง Scenario15ที่ให้ผลลัพธ์ซ้ำกับ scenario ก่อนหน้า สรุปแล้วการสุ่มค่า k ที่ให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด ในการทดลองแบ่งชุดข้อมูลด้วยอัตราส่วน 80:20 คือ Scenario14 ซึ่งกำหนดค่า k = 175

ตาราง 5 แสดงผลลัพธ์จากการทดสอบสุ่มหาค่า k ที่ดีที่สุด ของการแบ่งข้อมูล 80:20

	K	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.649	0.151	0.107	0.085	0.335	0.092	1.42
Scenario2	15	0.729	0.188	0.135	0.108	0.382	0.123	3.20
Scenario3	25	0.769	0.2	0.144	0.115	0.4	0.133	4.58
Scenario4	35	0.794	0.207	0.149	0.119	0.41	0.14	6.45
Scenario5	45	0.814	0.211	0.152	0.122	0.417	0.144	8.37
Scenario6	55	0.829	0.213	0.154	0.124	0.422	0.146	10.28
Scenario7	65	0.841	0.216	0.156	0.126	0.426	0.149	12.29
Scenario8	75	0.851	0.218	0.158	0.127	0.429	0.15	14.09
Scenario9	85	0.859	0.217	0.158	0.128	0.431	0.152	16.12
Scenario10	95	0.865	0.219	0.16	0.129	0.433	0.153	18.16
Scenario11	105	0.871	0.22	0.16	0.13	0.434	0.154	20.06
Scenario12	125	0.881	0.22	0.162	0.131	0.436	0.155	25.07
Scenario13	155	0.891	0.221	0.162	0.132	0.438	0.156	29.40
Scenario14	175	0.897	0.222	0.163	0.133	0.44	0.157	33.03
Scenario15	205	0.903	0.221	0.164	0.133	0.441	0.157	38.53

จากนั้นดำเนินการเปลี่ยนแปลงอัตราส่วนในการแบ่งชุดข้อมูล train data และ test data เป็น 75:25 โดยกำหนดให้การสุ่มค่า k ใช้ค่าเดียวกันกับการทดลองแบ่งชุดข้อมูลด้วยอัตราส่วน 90:10 ผลลัพธ์ประสิทธิภาพของแบบจำลองที่ได้จากการสุ่มค่า k จากโปรแกรม Rapid Miner ดังภาพประกอบ 38



ภาพประกอบ 38 แสดงผลลัพธ์จากการทดสอบสุ่มหาค่า k ที่ดีที่สุดของเทคนิค User k-NN ในโปรแกรม Rapid Miner ของการแบ่งข้อมูล 75:25

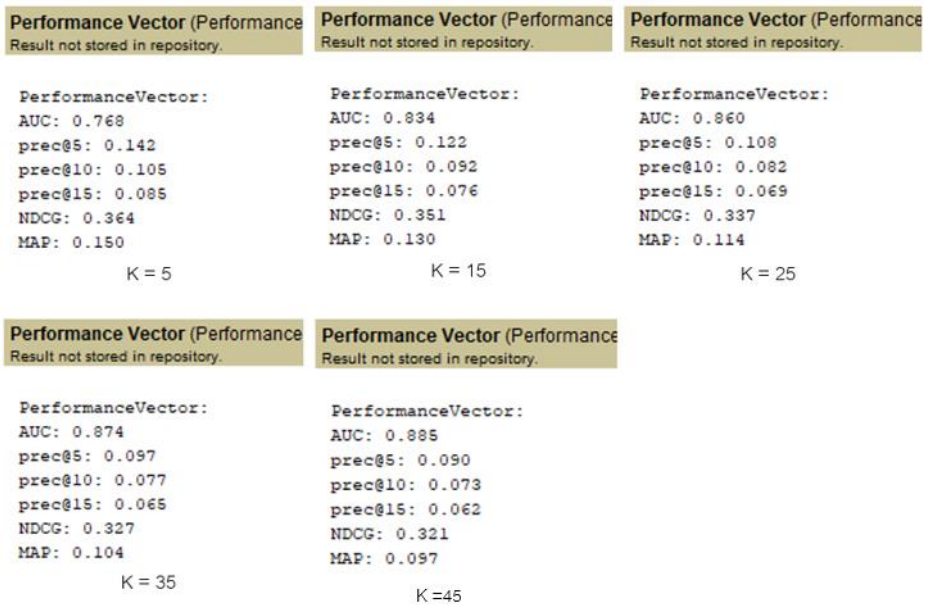
จากตาราง 6 นี้พบว่าผลลัพธ์ที่ได้มีทิศทางเช่นเดียวกันกับการทดลองทั้ง 2 แบบก่อนหน้า คือ ยิ่งค่า k มีจำนวนมากขึ้น ค่าประสิทธิภาพของแบบจำลองหรือ AUC จะยิ่งเพิ่มขึ้น ในขณะที่ค่าความแม่นยำที่ตำแหน่งที่ 5 หรือ prec@5 ที่มีแนวโน้มเพิ่มขึ้นเมื่อจำนวน k มากขึ้น แต่เมื่อดูจากตาราง 6 จะพบว่าเมื่อเข้าสู่ช่วง Scenario11 หรือกำหนดค่า k = 105 ผลลัพธ์ที่จะได้จะเริ่มให้ค่าเท่ากับ Scenario10 และเมื่อเข้าสู่ Scenario14 – Senario15 ผลลัพธ์ที่ได้นั้นมีค่าเท่ากับ Scenario13 ทั้งสองค่า และผลลัพธ์ของค่าความแม่นยำ ณ ตำแหน่งที่ 10 มีแนวโน้มผลลัพธ์ที่ให้ค่าเพิ่มขึ้นเรื่อย ๆ ยกเว้น Scenario14 ที่ให้ค่าเท่ากันกับ Scenario13 และค่าความแม่นยำ ณ ตำแหน่งที่ 15 นั้นจะให้ผลลัพธ์ที่เพิ่มขึ้นเรื่อย ๆ สำหรับค่า NDCG ได้ผลลัพธ์ที่เพิ่มขึ้นเรื่อย ๆ และสุดท้าย ค่าความแม่นยำเฉลี่ย หรือ MAP ได้ผลลัพธ์เพิ่มขึ้นเรื่อย ๆ เช่นกัน สรุปแล้วการสุ่มค่า k ที่

ให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด ในการทดลองแบ่งชุดข้อมูลด้วยอัตราส่วน 75:25 คือ Scenario15 ซึ่งกำหนดค่า k = 205

ตาราง 6 แสดงผลลัพธ์จากการทดสอบสุ่มหาค่า k ที่ดีที่สุด ของการแบ่งข้อมูล 75:25

	K	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.64	0.169	0.121	0.96	0.35	0.9	1.44
Scenario2	15	0.722	0.216	0.158	0.128	0.402	0.124	3.14
Scenario3	25	0.761	0.231	0.17	0.138	0.422	0.136	5.09
Scenario4	35	0.788	0.24	0.177	0.143	0.433	0.143	6.42
Scenario5	45	0.808	0.245	0.181	0.146	0.441	0.148	8.56
Scenario6	55	0.823	0.249	0.183	0.149	0.445	0.151	10.20
Scenario7	65	0.835	0.251	0.185	0.15	0.45	0.45	12.02
Scenario8	75	0.844	0.25	0.187	0.152	0.453	0.155	15.03
Scenario9	85	0.853	0.252	0.188	0.154	0.455	0.157	16.05
Scenario10	95	0.859	0.254	0.189	0.155	0.457	0.158	17.34
Scenario11	105	0.865	0.254	0.19	0.155	0.459	159	19.36
Scenario12	125	0.875	0.255	0.191	0.156	0.462	0.161	23.35
Scenario13	155	0.887	0.256	0.193	0.158	0.464	0.162	31.09
Scenario14	175	0.893	0.256	0.193	0.159	0.465	0.163	34.42
Scenario15	205	0.899	0.256	0.194	0.16	0.467	0.164	38.32

2) Item k-NN จะพิจารณาจากหนังสือที่ได้รับคะแนนความชื่นชอบจากผู้ใช้หลายคนไปทิศทางเดียวกัน ผลลัพธ์ประสิทธิภาพของแบบจำลองที่ได้จากการสุ่มค่า k ด้วยการแบ่งชุดข้อมูลด้วยอัตราส่วน 90:10 จากโปรแกรม Rapid Miner ดังภาพประกอบ 39



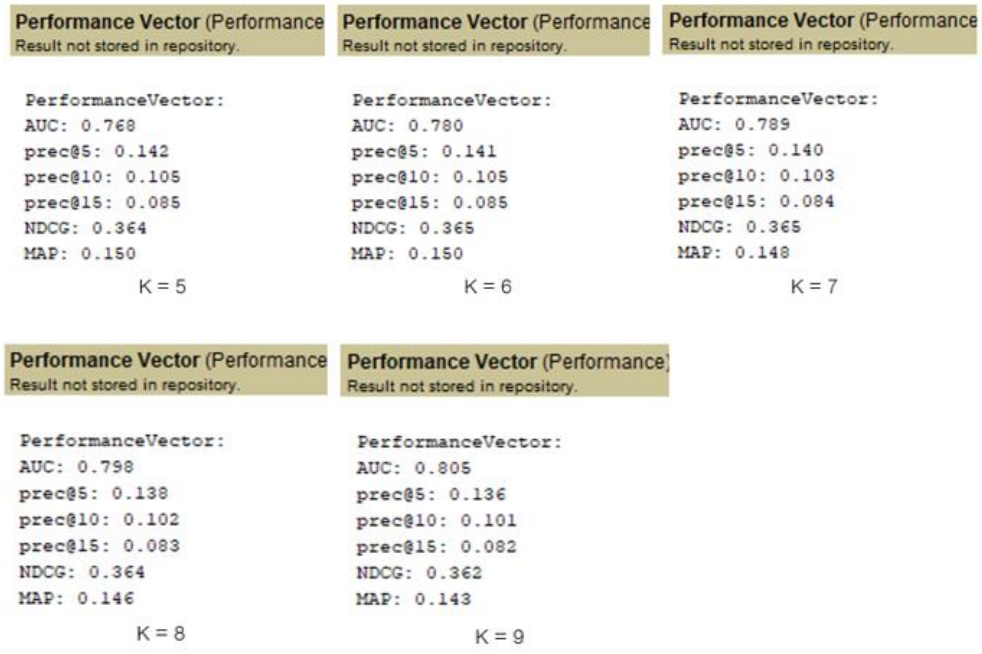
ภาพประกอบ 39 แสดงผลลัพธ์จากการทดสอบสุ่มหาค่า k ที่ดีที่สุดของเทคนิค Item k-NN ด้วยการแบ่งข้อมูลด้วยอัตราส่วน 90:10 ในโปรแกรม Rapid Miner

จากตาราง 7 แสดงผลการทดสอบสุ่มค่า k เป็น 5, 15, 25,35 และ 45 ตามการทดลองใน User k-NN นั้น พบว่าแนวโน้มของค่า AUC มีผลลัพธ์ที่เพิ่มมากขึ้น แต่ขณะเดียวกัน ค่า prec@5, prec@10, prec@15, NDCG และ MAP มีผลลัพธ์ที่น้อยลงเรื่อย ๆ เมื่อค่า k มีค่าเพิ่มขึ้น

ตาราง 7 แสดงผลการทดสอบสุ่มค่า k ที่ดีที่สุดของเทคนิค Item k-NN ด้วยการแบ่งข้อมูลด้วยอัตราส่วน 90:10

	K	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.768	0.142	0.105	0.105	0.364	0.150	2.25
Scenario2	15	0.834	0.122	0.092	0.076	0.351	0.130	4.48
Scenario3	25	0.860	0.108	0.082	0.069	0.337	0.114	7.45
Scenario4	35	0.874	0.097	0.077	0.065	0.327	0.104	10.57
Scenario5	45	0.885	0.090	0.073	0.062	0.321	0.097	12.4

ผู้ศึกษาวิจัยจึงหยุดการสุ่มค่า k และเปลี่ยนจำนวนในการสุ่มค่า k ใหม่โดยอ้างอิงจากผลลัพธ์ที่ได้ดังตาราง 7 จึงกำหนดค่า k เป็น 5, 6, 7, 8 และ 9 ตามลำดับ เพื่อทดสอบให้ได้ผลลัพธ์ค่าประสิทธิภาพที่ดีที่สุดใหม่อีกครั้ง โดยผลลัพธ์ที่ได้จากโปรแกรม Rapid Miner ดังภาพประกอบ 40



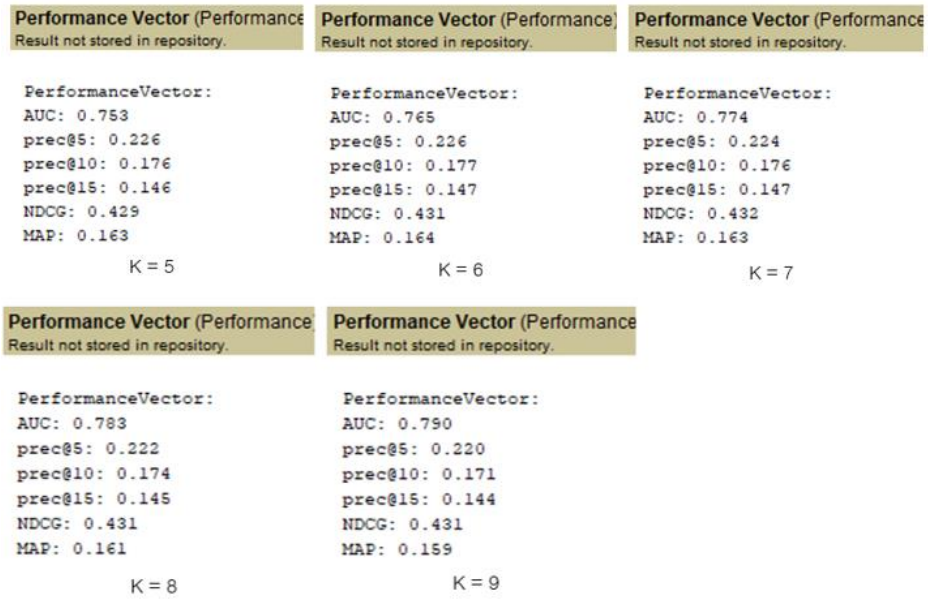
ภาพประกอบ 40 แสดงผลการทดสอบสุ่มค่า k = 5,6,7,8 และ 9 ของเทคนิค Item k-NN ด้วยการแบ่งข้อมูลด้วยอัตราส่วน 90:10 ในโปรแกรม Rapid Miner

จากตาราง 8 นั้น พบว่าค่าประสิทธิภาพของแบบจำลอง (AUC) ตั้งแต่ scenario1 ถึง scenario5 มีค่ามากขึ้นเรื่อย ๆ แต่ขณะเดียวกันค่าประสิทธิภาพอื่น ๆ นั้นกลับให้ค่าน้อยกว่าเดิมลงเรื่อย ๆ จึงสรุปได้ว่าในการทดลองแบบแบ่งข้อมูลด้วยอัตราส่วน 90:10 นั้น scenario1 หรือกำหนดค่า k = 5 จึงจะให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด

ตาราง 8 แสดงผลการทดสอบสุ่มค่า k = 5,6,7,8 และ 9 ของเทคนิค Item k-NN ด้วยการแบ่งข้อมูลด้วยอัตราส่วน 90:10

	K	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.768	0.142	0.105	0.105	0.364	0.150	2.25
Scenario2	6	0.780	0.141	0.105	0.085	0.365	0.150	2.34
Scenario3	7	0.789	0.140	0.103	0.084	0.365	0.148	2.54
Scenario4	8	0.798	0.138	0.102	0.083	0.364	0.146	3.17
Scenario5	9	0.805	0.136	0.101	0.082	0.362	0.143	3.22

จากนั้นดำเนินการเปลี่ยนแปลงอัตราส่วนในการแบ่งชุดข้อมูล train data และ test data เป็น 80:20 โดยกำหนดให้การสุ่มค่า k ใช้ค่าเดียวกันกับการทดลองแบ่งชุดข้อมูลด้วยอัตราส่วน 90:10 ผลลัพธ์ประสิทธิภาพของแบบจำลองที่ได้จากการสุ่มค่า k จากโปรแกรม Rapid Miner แสดงดังภาพประกอบ 41



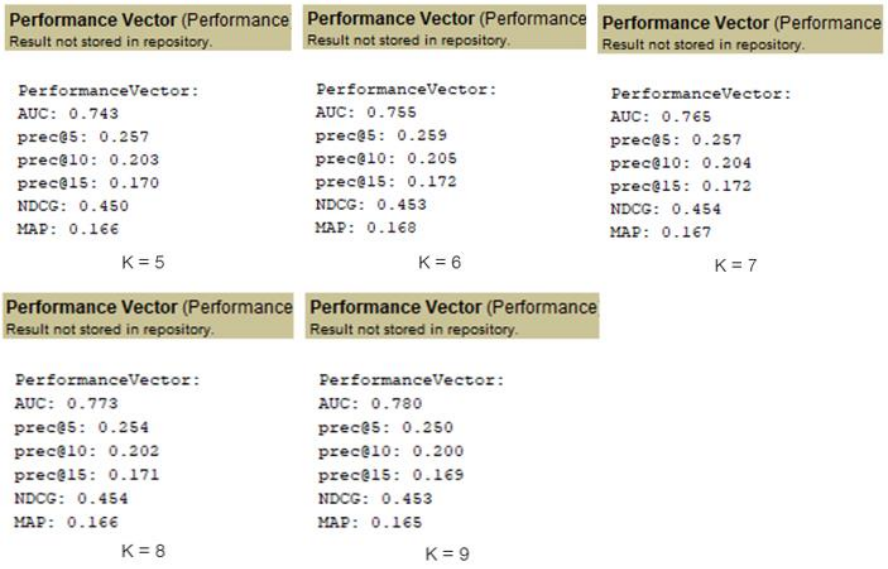
ภาพประกอบ 41 แสดงผลการทดสอบสุ่มค่า k = 5,6,7,8 และ 9 ของเทคนิค Item k-NN ด้วยการแบ่งข้อมูลด้วยอัตราส่วน 80:20 ในโปรแกรม Rapid Miner

จากตาราง 9 จะพบว่าในการทดสอบครั้งนี้ scenario2 หรือกำหนดค่า k = 6 นั้นให้ผลลัพธ์ที่ดีที่สุด กล่าวคือ ค่าประสิทธิภาพของแบบจำลอง (AUC) ตั้งแต่ scenario1 ถึง scenario5 มีค่ามากขึ้นเรื่อย ๆ แต่ขณะเดียวกันค่าประสิทธิภาพอื่น ๆ นั้นกลับให้ค่าน้อยกว่าเดิมลงเรื่อย ๆ จึงสรุปได้ว่าในการทดลองแบบแบ่งข้อมูลด้วยอัตราส่วน 80:20 นั้น scenario2 หรือกำหนดค่า k = 6 จึงจะให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด

ตาราง 9 แสดงผลการทดสอบสุ่มค่า k = 5,6,7,8 และ 9 ของเทคนิค item k-NN ด้วยการแบ่งข้อมูลด้วยอัตราส่วน 80:20

	K	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.753	0.226	0.176	0.146	0.429	0.163	2.15
Scenario2	6	0.765	0.226	0.177	0.147	0.431	0.164	2.26
Scenario3	7	0.774	0.224	0.176	0.147	0.432	0.163	2.4
Scenario4	8	0.783	0.222	0.174	0.145	0.431	0.161	2.54
Scenario5	9	0.790	0.220	0.171	0.144	0.431	0.159	3.03

จากนั้นดำเนินการเปลี่ยนแปลงอัตราส่วนในการแบ่งชุดข้อมูล train data และ test data เป็น 75:25 โดยกำหนดให้การสุ่มค่า k ใช้ค่าเดียวกันกับการทดลองแบ่งชุดข้อมูลด้วยอัตราส่วน 80:20 ผลลัพธ์ประสิทธิภาพของแบบจำลองที่ได้จากการสุ่มค่า k จากโปรแกรม Rapid Miner แสดงดังภาพประกอบ 42



ภาพประกอบ 42 แสดงผลการทดสอบสุ่มค่า k = 5,6,7,8 และ 9 ของเทคนิค Item k-NN ด้วยแบ่งข้อมูลด้วยอัตราส่วน 75:25 ในโปรแกรม Rapid Miner

จากตาราง 10 จะพบว่าในการทดสอบครั้งนี้จะพบว่า scenario2 หรือกำหนดค่า k = 6 นั้นให้ผลลัพธ์ที่ดีที่สุด กล่าวคือ ค่าประสิทธิภาพของแบบจำลอง (AUC) ตั้งแต่ scenario1 ถึง scenario5 มีค่ามากขึ้นเรื่อย ๆ แต่ขณะเดียวกันค่าประสิทธิภาพอื่น ๆ นั้นกลับให้ค่าน้อยกว่าเดิมลงเรื่อย ๆ จึงสรุปได้ว่าในการทดลองแบบแบ่งข้อมูลด้วยอัตราส่วน 75:25 นั้น scenario2 หรือกำหนดค่า k = 6 จึงจะให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด

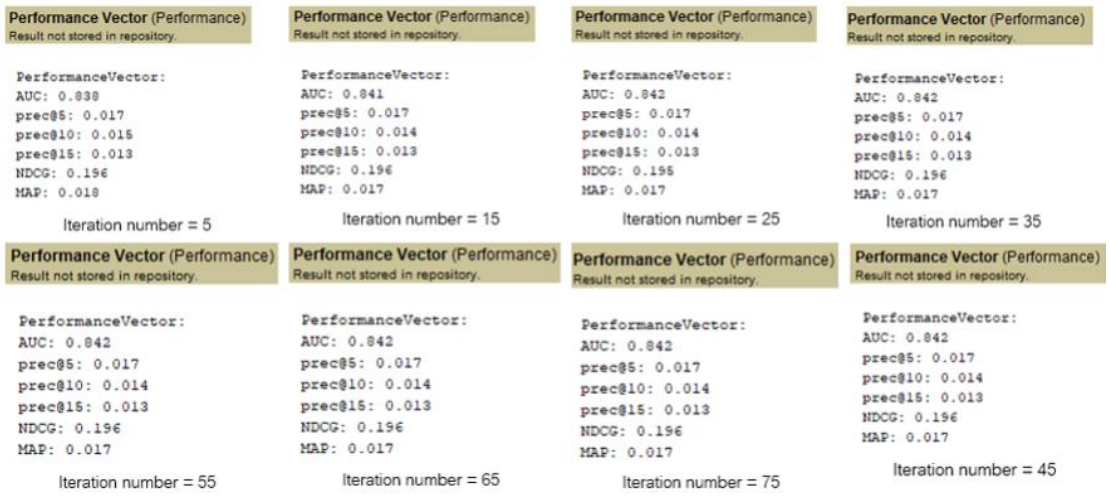
ตาราง 10 แสดงผลการทดสอบสุ่มค่า k = 5,6,7,8 และ 9 ของเทคนิค Item k-NN ด้วยการแบ่งข้อมูลด้วยอัตราส่วน 75:25

	K	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.743	0.257	0.203	0.170	0.450	0.166	2.02
Scenario2	6	0.755	0.259	0.205	0.172	0.453	0.168	2.19
Scenario3	7	0.765	0.257	0.204	0.172	0.454	0.167	2.28
Scenario4	8	0.773	0.254	0.202	0.171	0.454	0.166	2.47
Scenario5	9	0.780	0.250	0.200	0.169	0.453	0.165	2.57

4.2 ผลการศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค ผลการศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค Matrix Factorization

Matrix Factorization คือการพยายามแยกเมทริกซ์ให้มาเป็นเมทริกซ์ย่อย ๆ ลงไป โดยที่ผลคูณของเมกทริกซ์ย่อย ๆ นั้น จะได้กลับมาเป็นเมทริกซ์ตั้งต้น ยกตัวอย่างเช่น เมทริกซ์การให้คะแนนของผู้ใช้แต่ละคนต่อหนังสือแต่ละเล่ม (R) โดยจะทำการแยกเมทริกซ์นี้ออกเป็น 2 เมทริกซ์ คือ เมทริกซ์ผู้ใช้ (X) และเมทริกซ์หนังสือ (Y) โดยเมื่อนำเมทริกซ์ทั้ง X และ Y นี้มาคูณกัน ก็จะได้กลับมาเป็นเมทริกซ์การให้คะแนน (R) ตั้งต้น

ในโปรแกรม rapid miner มีให้กำหนดพารามิเตอร์หนึ่งชื่อว่า Iteration หรือการกำหนดค่าในการวนซ้ำในการหาค่าผิดพลาดระหว่างค่าที่ทำนายกับค่าจริงที่ต่ำที่สุดเพื่อให้ได้ข้อมูลน้ำหนักของเมทริกซ์ผู้ใช้ และแมทริกซ์หนังสือ (Mihelcic et al., 2012) ในการศึกษาวิจัยนี้ ผู้ศึกษาวิจัยได้ทำการสุ่มหาค่า Iteration ที่ทำให้แบบจำลองให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด โดยกำหนดค่าเป็น 5, 15, 25, 35, 45, 55, 65 และ 75 และกำหนดการแบ่งชุดข้อมูล train data และ test data เป็นอัตราส่วน 90:10 ผลลัพธ์จากโปรแกรม Rapid Miner แสดงดังภาพประกอบ 43



ภาพประกอบ 43 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 90:10 จากโปรแกรม Rapid Miner

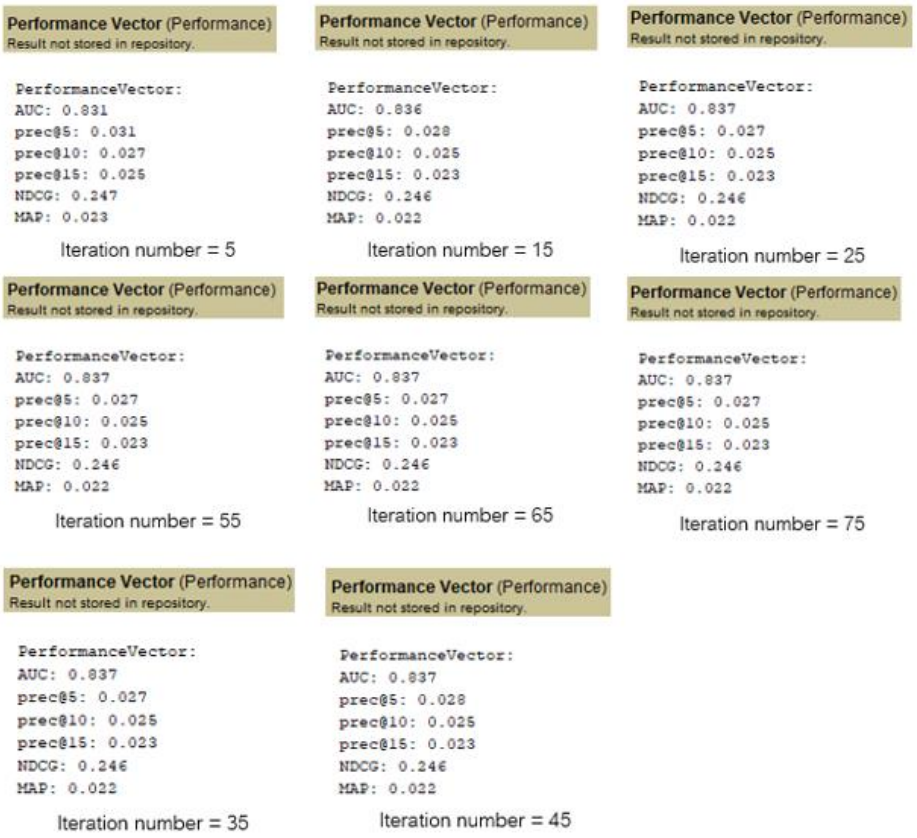
จากตาราง 11 จะพบว่า Scenario1 ถึง Scenario3 นั้น แบบจำลองให้ผลลัพธ์ประสิทธิภาพของค่าประสิทธิภาพของแบบจำลอง (AUC) ที่เพิ่มขึ้น แต่ค่าความแม่นยำ ณ ตำแหน่งที่ 5 มีค่าที่เท่ากันทั้ง 3 Scenario ขณะเดียวกันค่าความเป็นยำ ณ ตำแหน่งที่ 10 ใน

Scenario2 กลับให้ค่าน้อยกว่า Scenario1 และเริ่มให้ค่าคงที่ไปจนถึง Scenario8 สำหรับค่าความแม่นยำ ณ ตำแหน่งที่ 15 นั้น ให้ผลลัพธ์เท่ากันทุก Scenario ค่า NDCG ก็เช่นเดียวกัน มีเพียง scenario3 ที่ให้ค่าแตกต่างเท่านั้น สุดท้ายเป็นค่าความแม่นยำเฉลี่ยที่แบบจำลองให้ผลลัพธ์ใน Scenario2 น้อยกว่า Scenario1 และให้ค่าเท่าเดิมจนถึง Scenario8 โดยสรุปคือ Scenario4 หรือกำหนด Iteration = 35 ครั้ง แบบจำลองให้ผลลัพธ์ประสิทธิภาพดีที่สุด

ตาราง 11 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 90:10

	Iteration	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.838	0.017	0.015	0.013	0.196	0.018	2.07
Scenario2	15	0.841	0.017	0.014	0.013	0.196	0.017	1.27
Scenario3	25	0.842	0.017	0.014	0.013	0.195	0.017	1.36
Scenario4	35	0.842	0.017	0.014	0.013	0.196	0.017	1.31
Scenario5	45	0.842	0.017	0.014	0.013	0.196	0.017	2.04
Scenario6	55	0.842	0.017	0.014	0.013	0.196	0.017	2.04
Scenario7	65	0.842	0.017	0.014	0.013	0.196	0.017	2.12
Scenario8	75	0.842	0.017	0.014	0.013	0.196	0.017	2.15

จากนั้นดำเนินการกำหนดการแบ่งชุดข้อมูล train data และ test data เป็นอัตราส่วน 80:20 และสุ่มหาค่า Iteration ที่จะทำให้แบบจำลองให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด โดยกำหนดค่าเป็น 5, 15, 25, 35, 45, 55, 65 และ 75 เช่นเดียวกันกับการทดสอบก่อนหน้า โดยผลลัพธ์จากโปรแกรม Rapid Miner แสดงดังภาพประกอบ 44



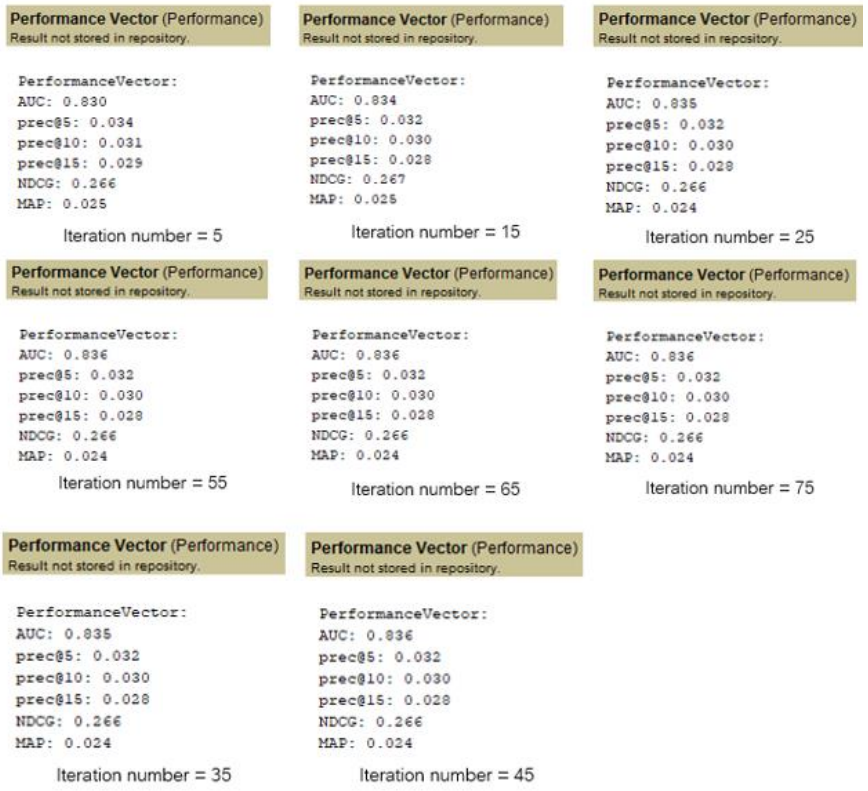
ภาพประกอบ 44 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 80:20 จากโปรแกรม Rapid Miner

จากตารางที่ 12 จะพบว่าค่าประสิทธิภาพแบบจำลอง (AUC) ให้ค่าเพิ่มขึ้นตั้งแต่ Scenario1 ถึง Scenrio3 หลังจากนั้นจึงเป็นค่าเดิมไปจนถึง Scenario8 ขณะที่ค่าความแม่นยำ ณ ตำแหน่งที่ 5 นั้นให้ค่าน้อยลงเรื่อย ๆ จนถึง Scenario3 จากนั้นจึงให้ค่าเดิมไปจนถึง Scenario8 แต่ที่ค่าความแม่นยำ ณ ตำแหน่งที่ 5 ให้ค่าผลลัพธ์น้อยลงจนถึง Scenario3 จากนั้นจึงให้ค่าเดิมไปเรื่อย ๆ และมีค่าเพิ่มขึ้นใน Scenario5 แล้วใน Scenario6 จึงกลับมามีค่าเท่ากันกับ Scenario3 และให้ค่าเดิมไปจนถึง Scenario8 ในขณะที่ค่าความแม่นยำที่ตำแหน่งที่ 10, ค่าความแม่นยำ ณ ตำแหน่งที่ 15, ค่า NDCG และค่าความแม่นยำเฉลี่ยนั้นมีค่าผลลัพธ์ที่น้อยถึง Scenario2 แล้วให้ค่าเดิมไปจนถึง Scenario8 ดังนั้นจึงสรุปได้ว่าแบบจำลองจะให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุดคือ Senario3 หรือกำหนดการวนซ้ำ Iteration = 25 ครั้ง

ตาราง 12 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 80:20

	Iteration	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.831	0.031	0.027	0.025	0.247	0.023	1.33
Scenario2	15	0.836	0.028	0.025	0.023	0.246	0.022	1.33
Scenario3	25	0.837	0.027	0.025	0.023	0.246	0.022	1.38
Scenario4	35	0.837	0.027	0.025	0.023	0.246	0.022	1.56
Scenario5	45	0.837	0.028	0.025	0.023	0.246	0.022	1.34
Scenario6	55	0.837	0.027	0.025	0.023	0.246	0.022	1.52
Scenario7	65	0.837	0.027	0.025	0.023	0.246	0.022	2.00
Scenario8	75	0.837	0.027	0.025	0.023	0.246	0.022	2.01

จากนั้นดำเนินการกำหนดการแบ่งชุดข้อมูล train data และ test data เป็นอัตราส่วน 75:25 และสุ่มหาค่า Iteration ที่จะทำให้แบบจำลองให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด โดยกำหนดค่าเป็น 5, 15, 25, 35, 45, 55, 65 และ 75 เช่นเดียวกันกับการทดสอบก่อนหน้า โดยผลลัพธ์จากโปรแกรม Rapid Miner แสดงดังภาพประกอบ 45



ภาพประกอบ 45 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 75:25 จากโปรแกรม Rapid Miner

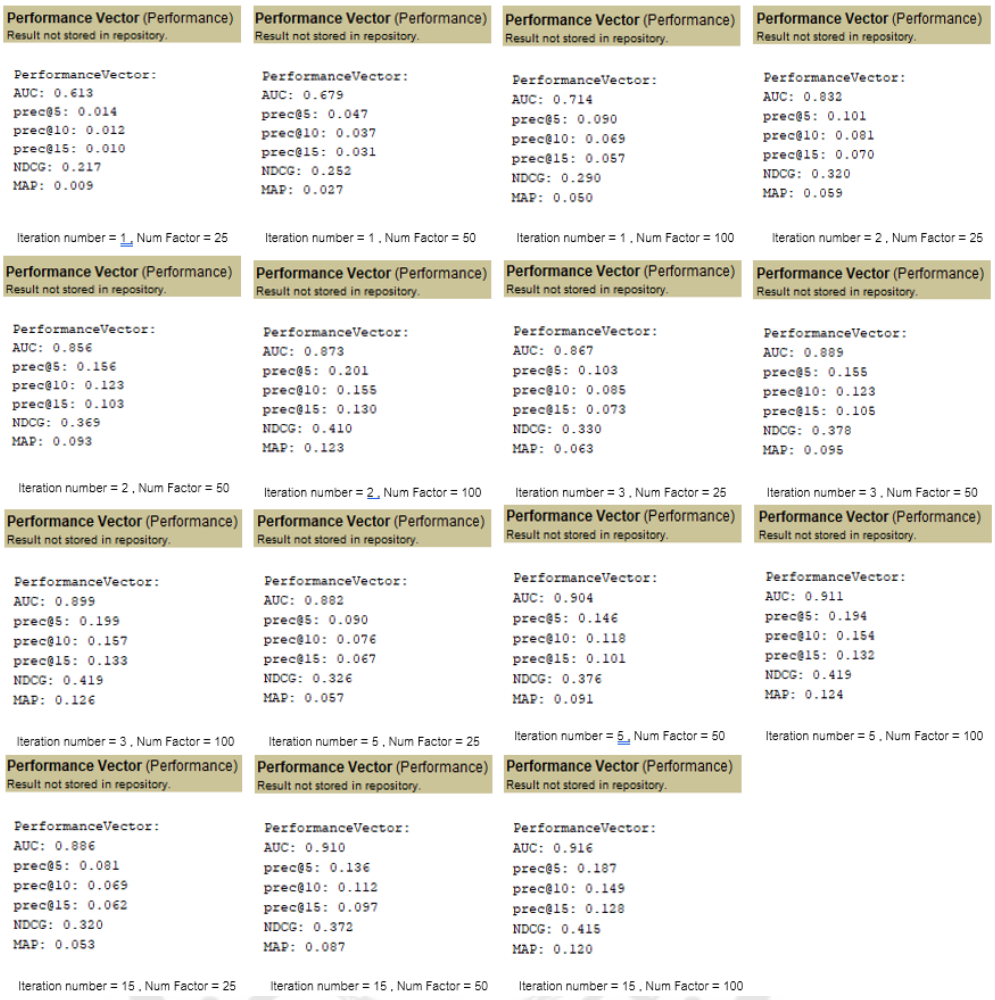
จากตาราง 13 จะพบว่าค่าประสิทธิภาพแบบจำลอง (AUC) ให้ค่าเพิ่มขึ้นตั้งแต่ Scenario1 ถึง Scenrio4 ที่ให้ค่าเดิมเหมือนกับ Scenario3 แต่ใน Scenari5 จะเพิ่มขึ้น หลังจากนั้นจึงเป็นค่าเดิมไปจนถึง Scenario8 ขณะที่ค่าความแม่นยำ ณ ตำแหน่งที่ 5, ค่าความแม่นยำ ณ ตำแหน่งที่ 10 และค่าความแม่นยำ ณ ตำแหน่งที่ 15 นั้นให้ค่าน้อยลงใน Scenario2 จากนั้นจึงให้ค่าเดิมไปจนถึง Scenario8 แต่ที่ค่าNDGC ให้ค่าผลลัพธ์มากขึ้นใน Scenario2 แล้วน้อยลงใน Scenario3 จากนั้นจึงให้ค่าเดิมไปจนถึง Scenario8 ในขณะที่ค่าความแม่นยำเฉลี่ยนั้น ที่ Scenario2 ให้ค่าซ้ำเดิมกับ Scenario1 แล้วให้ค่าที่น้อยตั้งแต่ Scenario3 และให้ค่าเดิมไปจนถึง Scenario8 ดังนั้นจึงสรุปได้ว่าแบบจำลองจะให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุดคือ Senario2 หรือกำหนดการวนซ้ำ Iteration = 25 ครั้ง

ตาราง 13 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 75:25

	Iteration	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	5	0.830	0.034	0.031	0.029	0.266	0.025	1.44
Scenario2	15	0.834	0.032	0.030	0.028	0.267	0.025	1.28
Scenario3	25	0.835	0.032	0.030	0.028	0.266	0.024	1.14
Scenario4	35	0.835	0.032	0.030	0.028	0.266	0.024	1.42
Scenario5	45	0.836	0.032	0.030	0.028	0.266	0.024	1.47
Scenario6	55	0.836	0.032	0.030	0.028	0.266	0.024	2.08
Scenario7	65	0.836	0.032	0.030	0.028	0.266	0.024	2.00
Scenario8	75	0.836	0.032	0.030	0.028	0.266	0.024	2.17

โดยสรุปจะได้ว่าการทดสอบด้วยการกำหนดการแบ่งชุดข้อมูล train data และ test data ด้วยอัตราส่วน 75:25 และกำหนดค่าการวนซ้ำหรือ Iteration เท่ากับ 25 ครั้งนั้น แบบจำลองให้ผลลัพธ์ค่าประสิทธิภาพที่ดีที่สุด และจากการทดสอบทั้งสามรูปแบบนี้พบว่าผลลัพธ์ที่ได้มีทิศทางเช่นเดียวกัน นั่นคือการกำหนดค่าการวนซ้ำหรือ Iteration เมื่อกำหนดค่าเพิ่มจนถึงจุดหนึ่งแล้วผลลัพธ์ค่าประสิทธิภาพของแบบจำลองนั้นจะไม่เปลี่ยนแปลง หรือให้ค่าเพิ่มขึ้นแล้ว ทางผู้ศึกษาวิจัยจึงเลือกกำหนดตัวแปรของโอเปอร์เรเตอร์ Matrix Factorization ในส่วนของตัวแปรขั้นสูงเพิ่มเติมเพื่อทดสอบว่าจะสามารถเพิ่มค่าประสิทธิภาพของแบบจำลองได้หรือไม่

โดยกำหนดค่าจำนวนปัจจัยแฝง หรือค่า Num factor ปัจจัยแฝงนี้หมายถึงคุณสมบัติบางประการที่อยู่ในหนังสือแต่ละเล่มและคุณสมบัติบางประการของหนังสือที่ผู้ใช้แต่ละคนให้ความสำคัญ เริ่มต้นนั้นโอเปอร์เรเตอร์นี้จะกำหนดค่าปัจจัยแฝงหรือ Num factor มีค่าเริ่มต้นเท่ากับ 10 ซึ่งในการทดลองทั้ง 3 แบบก่อนหน้านี้ผู้ศึกษาวิจัยได้ใช้ค่าตั้งต้นในส่วนข้อตัวแปรขั้นสูงตามที่ซอฟต์แวร์กำหนดทั้งหมด ในการทดสอบต่อไปนี้จึงจะทำการสุ่มค่าตัวแปรสองตัวคือจำนวนการวนซ้ำ Iteration = 1, 2, 3, 5, 15 และจำนวนปัจจัยแฝง Num factor = 25, 50, 100 ผลลัพธ์จากโปรแกรม Rapid Miner แสดงดังภาพประกอบ 46



ภาพประกอบ 46 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำและ Num Factor ของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 75:25 จากโปรแกรม Rapid Miner

ในการทดสอบนี้จะกำหนดการแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบด้วยอัตราส่วน 75:25 อ้างอิงจากการทดสอบสุ่มค่าจำนวการวนซ้ำก่อนหน้าเนื่องจากให้ผลลัพธ์ดีที่สุดจากทั้ง 3 แบบ จากตางราง 14 พบว่าเมื่อกำหนดปัจจัยแฝงที่มีค่ามากขึ้น จะทำให้ผลลัพธ์ที่ได้นั้นมีค่าที่ดียิ่งขึ้น โดย Scenario15 ที่กำหนดค่าการวนซ้ำ iteration = 15 ครั้ง และกำหนดปัจจัยแฝง Num factor = 100 จะทำให้ค่าประสิทธิภาพของแบบจำลอง (AUC) ให้ค่าที่ดีที่สุดคือ 0.916 แต่ค่าประสิทธิภาพอื่น ๆ นั้นกลับให้ค่าลดลง ซึ่งน้อยกว่า Scenario6 ที่กำหนดค่าจำนวนปัจจัยแฝงเท่ากันคือ 100 แต่กำหนดการวนซ้ำเพียง 2 ครั้ง ต่อมาเลือกตรวจสอบใน Scenario อื่น ๆ ที่กำหนดค่าปัจจัยแฝงเท่ากับ 100 ด้วยเช่นเดียวกัน เนื่องจากให้ค่าประสิทธิภาพของแบบจำลอง

(AUC) สูง ซึ่งคือ Scenario3, Scenario9, Scenario12 ซึ่งเมื่อเปรียบเทียบค่าผลลัพธ์ประสิทธิภาพที่ได้แล้วนั้นจะพบว่า Scenario12 ให้ผลลัพธ์ไปในทิศทางที่ดีกว่า โดยกำหนดค่าวนซ้ำ Iteration = 5 และค่าปัจจัยแฝง Num factor = 100

ตาราง 14 แสดงผลการทดสอบสุ่มจำนวนครั้งในการวนซ้ำและปัจจัยแฝงของแบบจำลองการแยกตัวประกอบแมทริกซ์ ที่แบ่งชุดข้อมูลด้วยอัตราส่วน 75:25

	Iteration	num factor	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario1	1	25	0.613	0.014	0.012	0.01	0.217	0.009	1.00
Scenario2	1	50	0.679	0.047	0.037	0.031	0.252	0.027	1.14
Scenario3	1	100	0.714	0.09	0.069	0.057	0.29	0.05	2.03
Scenario4	2	25	0.832	0.101	0.081	0.07	0.32	0.059	1.03
Scenario5	2	50	0.856	0.156	0.123	0.103	0.369	0.093	1.19
Scenario6	2	100	0.873	0.201	0.155	0.13	0.41	0.123	2.45
Scenario7	3	25	0.867	0.103	0.085	0.073	0.33	0.063	1.04
Scenario8	3	50	0.889	0.155	0.123	0.105	0.378	0.095	1.27
Scenario9	3	100	0.899	0.199	0.157	0.133	0.419	0.126	3.24
Scenario10	5	25	0.882	0.09	0.076	0.067	0.326	0.057	1.07
Scenario11	5	50	0.904	0.146	0.118	0.101	0.376	0.091	1.42
Scenario12	5	100	0.911	0.194	0.154	0.132	0.419	0.124	4.50
Scenario13	15	25	0.886	0.081	0.069	0.062	0.32	0.053	1.23
Scenario14	15	50	0.91	0.136	0.112	0.097	0.372	0.087	3.00
Scenario15	15	100	0.916	0.187	0.149	0.128	0.415	0.12	11.51

4.3 ผลการศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิค Model Combiner (k-NN + MF)

แบบจำลองนี้ จะเป็นการรวมอัลกอริทึมทั้ง 3 เทคนิคเข้ามาด้วยกันซึ่งจะเรียกว่า Combiner Model โดยเทคนิคนี้เป็นการรวมข้อดีของแต่ละแบบจำลองเข้าด้วยกัน เพื่อให้ได้ค่าประสิทธิภาพดียิ่งขึ้น โดยจากรูปที่ 34 นั้น โอเปอร์เรเตอร์ Model Combiner นั้น จะดำเนินการรวมแบบจำลองขาเข้าทั้งหมดเข้าด้วยกัน โดยจะทำงานแบบคู่ขนานกันไป และผลลัพธ์ที่ได้จาการรวมแบบจำลองนี้จะเป็นในลักษณะของค่าถ่วงน้ำหนักเฉลี่ยของแบบจำลองรวมทั้งหมด

ในการทดลองนี้จะทำการหาค่าประสิทธิภาพของแบบจำลองโดยการเปรียบเทียบแบบจำลองที่กำหนดค่าตัวแปรต่าง ๆ เป็นค่าเริ่มต้นของซอฟต์แวร์ RapidMiner กับแบบจำลองที่

กำหนดตัวแปรตามผลลัพธ์ที่ได้จากการทดลองในหัวข้อที่ 4.1- 4.2 พร้อมทั้งกำหนดการแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบด้วยอัตราส่วน 90:10, 80:20 และ 75:50 ผลลัพธ์จากโปรแกรม Rapid Miner ดังภาพประกอบ 47

Scenario 1

Row No.	AUC	prec@5	prec@10	prec@15	NDCG	MAP
1	0.930	0.124	0.087	0.070	0.362	0.136

Scenario 2

Row No.	AUC	prec@5	prec@10	prec@15	NDCG	MAP
1	0.937	0.160	0.115	0.093	0.405	0.174

Scenario 3

Row No.	AUC	prec@5	prec@10	prec@15	NDCG	MAP
1	0.932	0.256	0.196	0.163	0.481	0.194

Scenario 4

Row No.	AUC	prec@5	prec@10	prec@15	NDCG	MAP
1	0.929	0.293	0.229	0.192	0.506	0.201

ภาพประกอบ 47 แสดงผลการทดลองเปรียบเทียบแบบจำลอง Model Combiner จากโปรแกรม Rapid Miner

จากตาราง 15 จะพบว่า Scenario 1 ที่เป็นกำหนดค่าตัวแปรตั้งต้นของโอเปอร์เรเตอร์ รวมถึงการแบ่งข้อมูลเรียนรู้และข้อมูลทดสอบเป็นอัตราส่วน 90:10 ซึ่งเป็นค่าตั้งต้นของโอเปอร์เรเตอร์ด้วยเช่นกันนั้น แบบจำลองให้ค่าผลลัพธ์ประสิทธิภาพของค่าความแม่นยำ ณ ตำแหน่งที่ 5, ค่าความแม่นยำ ณ ตำแหน่งที่ 10 และค่าความแม่นยำ ณ ตำแหน่งที่ 15 มีค่าที่น้อยกว่า Scenario อื่น ๆ มาก เนื่องจาก Scenario2 ถึง Scenario4 นั้น ตัวแปรต่าง ๆ ของแบบจำลองทั้ง 3 ในโอเปอร์เรเตอร์ Model Combiner มาจากการทดลองที่มีการปรับปรุงตัวแปรเหล่านั้นให้ได้ค่าผลลัพธ์ประสิทธิภาพที่ดีที่สุดแล้ว ต่างกันเพียงการกำหนดอัตราส่วนข้อมูลเรียนรู้และข้อมูลทดสอบเท่านั้น ซึ่งจะเห็นได้ว่าค่าประสิทธิภาพแบบจำลอง (AUC) ของ Scenario2 ให้ค่าที่ดีที่สุด คือ 0.937 แต่ให้ค่าผลลัพธ์ความแม่นยำ ณ ตำแหน่งที่ 5, ความแม่นยำ ณ ตำแหน่งที่ 10 และความแม่นยำ ณ ตำแหน่งที่ 15 มีค่าน้อยกว่า Scenario4 ขณะเดียวกัน Scenaro4 กลับให้ค่า

ประสิทธิภาพของแบบจำลอง (AUC) น้อยกว่า Scenario2 จากการทดลองนี้จึงสรุปได้ว่า Scenario4 แบบจำลองให้ค่าผลลัพธ์ที่ดีที่สุด ซึ่งใน Scenario นี้เป็นการกำหนดค่าตัวแปรตามผลการทดลองในหัวข้อที่ 4.1 – 4.2 คือ เทคนิค User k-NN กำหนดค่า K = 205, เทคนิค Item k-NN กำหนดค่า K = 5 และเทคนิค Matrix Factorization กำหนดค่าการวนซ้ำ Iteration = 5 และปัจจัยแฝง Num factor = 100 และกำหนดอัตราส่วนการแบ่งข้อมูลเรียนรู้และข้อมูลทดสอบเป็น 75:25

ตาราง 15 แสดงผลการทดลอง Model Combiner ระหว่าง User KNN และ MF ตามการทดลองที่กำหนด

	Train / Test Data set	K-user	K-item	Num Factor	Iteration	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
Scenario 1	90/10	80	80	10	30	0.93	0.124	0.087	0.07	0.362	0.136	39.53
Scenario 2	90/10	205	5	100	5	0.937	0.16	0.115	0.093	0.405	0.174	45.53
Scenario 3	80/20	205	5	100	5	0.932	0.256	0.196	0.163	0.481	0.194	46.11
Scenario 4	75/25	205	5	100	5	0.929	0.293	0.229	0.192	0.506	0.201	45.41

4.4 ผลการเปรียบเทียบประสิทธิภาพของแบบจำลองด้วยเทคนิคต่าง ๆ

จากการทดลองในหัวข้อที่ 4.1 – 4.3 นั้น สรุปได้ว่าสร้างแบบจำลองแนะนำหนังสือจากชุดข้อมูล goodbooks-10k ด้วยเทคนิคการรวมแบบจำลองหรือ Model Combiner นั้นให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด เนื่องจากเป็นการทำงานร่วมกันของทั้ง 3 เทคนิค คือ Item k-NN, User k-NN และ Matrix Factorization โดยการกำหนดค่าตัวแปรแต่ละตัวของทั้ง 3 เทคนิคนั้นได้ผ่านการทดสอบเพื่อหาค่าประสิทธิภาพที่ดีที่สุดแล้ว จึงทำให้เทคนิคการรวมแบบจำลองนั้นให้ค่าประสิทธิภาพออกมาดีที่สุดจากเทคนิคทั้งหมด

โดยเทคนิคการรวมแบบจำลองดังกล่าวจะกำหนดอัตราส่วนการแบ่งข้อมูลเรียนรู้และข้อมูลทดสอบเป็น 75:25 และกำหนดค่าตัวแปรของเทคนิคทั้ง 3 แบบ คือ เทคนิค User k-NN

กำหนดค่า K = 205, เทคนิค Item k-NN กำหนดค่า K = 5 และเทคนิค Matrix Factorization กำหนดค่าการวนซ้ำ Iteration = 5 และปัจจัยแฝง Num factor = 100 ซึ่งแบบจำลองจะให้ค่าประสิทธิภาพของแบบจำลอง (AUC) = 0.929, ค่าความแม่นยำ ณ ตำแหน่งที่ 5 (prec@5) = 0.293, ค่าความแม่นยำ ณ ตำแหน่งที่ 10 (prec@10) = 0.229, ค่าความแม่นยำ ณ ตำแหน่งที่ 15 (prec@15) = 0.192, ค่า NDCG = 0.506 และค่าความแม่นยำเฉลี่ย (MAP) = 0.201

บทที่ 5
สรุปผลการวิจัย

ในการศึกษาวิจัยการสร้างแบบจำลองการแนะนำหนังสือด้วยเทคนิคเหมืองข้อมูล โดยการใช้ RapidMiner โดยใช้ชุดข้อมูล goodbooks-10k ใน ผู้วิจัยได้เปรียบเทียบประสิทธิภาพของแบบจำลองแต่ละอัลกอริทึมเพื่อให้ได้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด โดยแบ่งหัวข้อการสรุปผลดังนี้

1.สรุปผลการวิจัย

2.ข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในการศึกษาวิจัยนี้เนื่องจากชุดข้อมูลที่ผู้ศึกษาวิจัยเลือกใช้นั้นเป็นข้อมูลการให้คะแนนหนังสือของผู้ใช้ ซึ่งชุดข้อมูลการให้คะแนนที่ชัดเจน (Explicit rating) ทำให้ทราบว่าผู้ใช้คนนั้น ๆ มีความชื่นชอบหนังสือเล่มนั้น ๆ มากน้อยเพียงใด ผู้ศึกษาวิจัยจึงเลือกใช้หลักการในสร้างแบบจำลองการแนะนำหนังสือด้วยวิธี Collaborative Filtering ที่เป็นที่นิยมใช้กับระบบที่มีข้อมูลการให้คะแนนความชื่นชอบ (Rating) ต่อสินค้า (Item) และเป็นการพิจารณาหาความคล้ายคลึงระหว่างผู้ใช้ โดยจะพิจารณาจากประวัติของผู้ใช้เป้าหมาย กับผู้ใช้คนอื่นที่อยู่ในระบบว่ามีความชื่นชอบในสินค้าเป็นไปในทิศทางเดียวกันหรือไม่ จากนั้นจึงจะคำนวณหาค่าคะแนนทำนายของผู้ใช้เป้าหมายต่อสินค้าที่อยู่ในประวัติของผู้ใช้ที่คล้ายกันแต่ไม่เคยอยู่ในประวัติของผู้ใช้เป้าหมาย แล้วจึงเลือกค่าคะแนนทำนายที่สูงที่สุดเพื่อทำการแนะนำสินค้าให้แก่ผู้ใช้เป้าหมาย ในการศึกษาวิจัยนี้ได้เลือกใช้เทคนิคเพื่อนบ้านใกล้เคียง (k-Nearest Neighbors) และเทคนิคการแยกเมทริกซ์ (Matrix Factorization) ซึ่งเป็นเทคนิคที่นิยมนำมาใช้ในระบบแนะนำด้วยวิธี Collaborative Filtering โดยในการศึกษาและเปรียบเทียมอัลกอริทึมนี้ได้เลือกใช้โปรแกรมชื่อว่า RapidMiner Studio เป็นเครื่องมือที่ใช้ในการศึกษาวิจัย ซึ่งโปรแกรม RapidMiner Studio นั้นจะเป็นประเภทสำหรับการศึกษาเพื่อให้โปรแกรมสามารถดำเนินการกับชุดข้อมูลที่มากกว่า 10,000 rows เนื่องจากชุดข้อมูลที่ผู้วิจัยใช้นั้นมีจำนวน 981,756 rows ซึ่งหากเป็นประเภทที่ไม่ต้องเสียค่าใช้จ่ายนั้นจะไม่สามารถดำเนินการได้ เมื่อศึกษาเกี่ยวกับโปรแกรม RapidMiner Studio พบว่าโปรแกรมนอกจากจะมีเครื่องมือที่สามารถใช้ในการวิเคราะห์เชิงสถิติและสร้างเป็นแผนภูมิประเภทต่าง ๆ รวมถึงเครื่องมือที่ใช้ในการสร้างแบบจำลองพื้นฐานแล้วนั้น ยังสามารถสร้างแบบจำลองด้วยการรวมอัลกอริทึมได้อีกด้วย ผู้วิจัยจึงเลือกศึกษาเทคนิครวมแบบจำลอง (Model combiner) เพิ่มเติมและนำมาเปรียบเทียบประสิทธิภาพการทำงานของแบบจำลองกับอีกสองเทคนิคข้างต้น

ก่อนนำเข้าชุดข้อมูลไปใช้ในการสร้างแบบจำลอง ผู้วิจัยได้ทำการศึกษาชุดข้อมูลและพบข้อมูลซ้ำจำนวน 4,487 rows จึงทำการลบข้อมูลผู้ใช้ที่ให้คะแนนหนังสือซ้ำออก โดยเลือกให้เหลือเพียงการให้คะแนนที่ผู้ใช้ให้ต่อหนังสือเล่มนั้น ๆ ที่สูงสุด จากนั้นจึงกรองชุดข้อมูลด้วยการพิจารณาข้อมูลการให้คะแนนของผู้ใช้ส่วนใหญ่ที่ให้คะแนนหนังสือตั้งแต่ 20 เล่มขึ้นไป เพื่อให้ได้ข้อมูลผู้ใช้ที่มากพอกับการพิจารณาหาความคล้ายคลึงกัน จากนั้นจึงเริ่มศึกษาการสร้างแบบจำลองด้วยเทคนิคต่าง ๆ โดยการศึกษาวิจัยมีการปรับค่าตัวแปรที่เกี่ยวข้องกับเทคนิคต่าง ๆ เพื่อให้แบบจำลองให้ค่าผลลัพธ์ประสิทธิภาพที่ดีที่สุด ซึ่งมีรายละเอียดดังนี้

1. การศึกษาการสร้างแบบจำลองการแนะนำหนังสือด้วยเทคนิคเพื่อนบ้านใกล้เคียง (k-Nearest Neighbors) นั้นได้แบ่งเป็น 2 รูปแบบ คือ 1)User-based หรือการหาความคล้ายคลึงระหว่างผู้ใช้ 2)Item-based หรือหาความคล้ายคลึงระหว่างสินค้า ซึ่งในโปรแกรม RapidMiner Studio มีโอเปอร์เรเตอร์ที่รองรับการทำงานทั้ง 2 รูปแบบ โดยผู้วิจัยได้แบ่งการทดสอบด้วยการสุ่มจำนวนค่า k ที่ใช้ในการพิจารณาหาความคล้ายคลึงระหว่างผู้ใช้และความคล้ายคลึงระหว่างหนังสือ แล้วจึงกำหนดการแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบออกเป็น 3 แบบคือ 90:10, 80:20, 75:25 ผลลัพธ์ที่ได้คือ การพิจารณาความคล้ายคลึงระหว่างผู้ใช้งานนั้นกำหนดค่า k = 205 และแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบด้วยอัตราส่วน 75:25 แบบจำลองให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด โดยผลลัพธ์ประสิทธิภาพที่ได้มีดังนี้ ค่าพื้นที่ใต้กราฟ (AUC) เท่ากับ 0.899, ค่าความแม่นยำ ณ ตำแหน่งที่ 5 (PREC@5) เท่ากับ 0.194, ค่าความแม่นยำ ณ ตำแหน่งที่ 10 (PREC@10) เท่ากับ 0.194, ค่าความแม่นยำ ณ ตำแหน่งที่ 15 (PREC@15) เท่ากับ 0.160, ค่า Normalized discounted cumulative gain (NDCG) เท่ากับ 0.467 และค่าความแม่นยำเฉลี่ย (MAP) เท่ากับ 0.164

สำหรับการพิจารณาความคล้ายคลึงระหว่างผู้ใช้งานนั้นกำหนดค่า K = 5 และแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบด้วยอัตราส่วน 75:25 แบบจำลองจะให้ค่าผลลัพธ์ประสิทธิภาพดีที่สุด โดยผลลัพธ์ประสิทธิภาพที่ได้มีดังนี้ ค่าพื้นที่ใต้โค้ง (AUC) เท่ากับ 0.743, ค่าความแม่นยำ ณ ตำแหน่งที่ 5 (PREC@5) เท่ากับ 0.257, ค่าความแม่นยำ ณ ตำแหน่งที่ 10 (PREC@10) เท่ากับ 0.203, ค่าความแม่นยำ ณ ตำแหน่งที่ 15 (PREC@15) เท่ากับ 0.170, ค่า Normalized discounted cumulative gain (NDCG) เท่ากับ 0.450 และค่าความแม่นยำเฉลี่ย (MAP) เท่ากับ 0.166

2. ศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิคการแยกเมทริกซ์ (Matrix Factorization) โดยผู้วิจัยได้แบ่งการทดสอบด้วยการแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบ

ออกเป็น 3 แบบคือ 90:10, 80:20, 75:25 และทำการสุ่มจำนวนการวนซ้ำ (Iteration) เพื่อหาค่าผิดพลาดระหว่างค่าที่ทำนายกับค่าจริงที่ต่ำที่สุดเพื่อให้ได้ข้อมูลน้ำหนักของเมทริกซ์ผู้ใช้และแมทริกซ์หนังสือ จากการทดสอบนี้จะพบว่าเมื่อทำการสุ่มค่าการวนซ้ำไปถึงจุดหนึ่ง ผลลัพธ์ประสิทธิภาพที่ได้จากแบบจำลองจะไม่มีการเปลี่ยนแปลง ผู้วิจัยจึงทำการกำหนดตัวแปรขั้นสูงที่อยู่ในโอเปอร์เรเตอร์ของโปรแกรม RapidMiner Studio เพิ่มคือจำนวนปัจจัยแฝง (Num factor) ที่จะเป็นคุณสมบัติบางประการที่อยู่ในหนังสือแต่ละเล่มและคุณสมบัติบางประการของหนังสือที่ผู้ใช้แต่ละคนให้ความสำคัญ โดยแบ่งการทำลองออกเป็น 3 แบบคือ 90:10, 80:20, 75:25 พร้อมทั้งสุ่มค่าปัจจัยแฝงและการวนซ้ำ ซึ่งค่าประสิทธิภาพที่ได้มีการเปลี่ยนแปลง และเพิ่มมากขึ้น ผลลัพธ์ที่ได้จากการทดลองนี้คือการแยกเมทริกซ์ ด้วยการแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบด้วยอัตราส่วน 75:25 กำหนดจำนวนครั้งในการวนซ้ำเท่ากับ 5 รอบ และกำหนดจำนวนปัจจัยแฝงเท่ากับ 100 แบบจำลองจะให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด โดยผลลัพธ์ประสิทธิภาพที่ได้มีดังนี้ ค่าพื้นที่ใต้โค้ง (AUC) เท่ากับ 0.911, ค่าความแม่นยำ ณ ตำแหน่งที่ 5 (PREC@5) เท่ากับ 0.194, ค่าความแม่นยำ ณ ตำแหน่งที่ 10 (PREC@10) เท่ากับ 0.154, ค่าความแม่นยำ ณ ตำแหน่งที่ 15 (PREC@15) เท่ากับ 0.132, ค่า Normalized discounted cumulative gain (NDCG) เท่ากับ 0.419 และค่าความแม่นยำเฉลี่ย (MAP) เท่ากับ 0.124

3. การศึกษาการสร้างแบบจำลองการแนะนำด้วยเทคนิคการรวมแบบจำลอง (Model Combiner) โดยผู้ศึกษาวิจัยได้แบ่งการทดสอบด้วยการแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบออกเป็น 4 แบบคือ 90:10, 80:20, 75:25 ที่กำหนดตัวแปรตัวแปรจากผลลัพธ์ที่ได้จากการทดลองด้วยเทคนิคเพื่อนบ้านใกล้เคียง (k-Nearest Neighbors) ทั้งประเภท User-based และประเภท Item-based และเทคนิคการแยกตัวประกอบ (Matrix Factorization) และสุดท้ายคือเปรียบเทียบกับแบบจำลองที่กำหนดตัวแปรต่าง ๆ และการแบ่งชุดข้อมูลเรียนรู้และข้อมูลทดสอบตามค่าเริ่มต้นของซอฟต์แวร์ RapidMiner Studio ผลลัพธ์ของการสร้างแบบจำลองด้วยเทคนิคการรวมแบบจำลองคือ ในเทคนิคเพื่อนบ้านใกล้เคียง (k-Nearest Neighbors) ประเภท User-based กำหนดค่า k เท่ากับ 205 และประเภท Item-based กำหนดค่า k เท่ากับ 5 ในเทคนิคการแยกแมทริกซ์ (Matrix Factorization) กำหนดจำนวนการวนซ้ำ Iteration เท่ากับ 5 และจำนวนปัจจัยแฝง Num factor เท่ากับ 100 ซึ่งกำหนดอัตราส่วนการแบ่งข้อมูลเรียนรู้และข้อมูลทดสอบเป็น 75:25 แบบจำลองจะให้ผลลัพธ์ประสิทธิภาพที่ดีที่สุด โดยผลลัพธ์ประสิทธิภาพที่ได้มีดังนี้ ค่าพื้นที่ใต้โค้ง (AUC) เท่ากับ 0.929, ค่าความแม่นยำ ณ ตำแหน่งที่ 5 (PREC@5) เท่ากับ 0.293, ค่าความแม่นยำ ณ ตำแหน่งที่ 10 (PREC@10) เท่ากับ 0.229, ค่าความแม่นยำ ณ ตำแหน่งที่ 15

(PREC@15) เท่ากับ 0.192, ค่า Normalized discounted cumulative gain (NDCG) เท่ากับ 0.506 และค่าความแม่นยำเฉลี่ย (MAP) เท่ากับ 0.201

4. ผลการเปรียบเทียบประสิทธิภาพของแบบจำลองการแนะนำด้วยเทคนิคเพื่อนบ้านใกล้เคียง (k-Nearest Neighbors) ประเภท User-based เทคนิคเพื่อนบ้านใกล้เคียง (k-Nearest Neighbors) ประเภท Item-based เทคนิคการแยกตัวประกอบ (Matrix Factorization) และเทคนิคการรวมแบบจำลองทั้งสามเข้าด้วยกันนั้น พบว่าการสร้างแบบจำลองการแนะนำด้วยการรวมแบบจำลองจะให้ผลลัพธ์ประสิทธิภาพดีที่สุด เนื่องจากเป็นการรวมทั้งสามเทคนิคเข้าด้วยกันนั้นจะเพิ่มประสิทธิภาพในการแนะนำได้ดีมากขึ้น อีกทั้งมีการกำหนดค่าตัวแปรต่าง ๆ ของแต่ละเทคนิคที่ได้จากการทดลองที่ให้ผลลัพธ์ที่ดีที่สุด จึงสรุปได้ว่าการสร้างแบบจำลองการแนะนำจากชุดข้อมูลการให้คะแนนหนังสือ rating ที่ชื่อว่า goodbooks-10k โดยการใช้เทคนิคการรวมแบบจำลอง User k-NN Item k-NN และ Matrix Factorization เข้าด้วยกันแบบจำลองการแนะนำหนังสือจึงจะให้ผลลัพธ์ประสิทธิภาพดีที่สุด

ตาราง 16 ค่าวัดประสิทธิภาพแต่ละแบบจำลองที่ดีที่สุดในการทดลอง

	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP
User k-NN	0.899	0.256	0.194	0.160	0.467	0.164
Item k-NN	0.743	0.257	0.203	0.170	0.450	0.166
MF	0.911	0.194	0.154	0.132	0.419	0.124
Model Combiner	**0.929**	**0.293**	**0.229**	**0.192**	**0.506**	**0.201**

ผลลัพธ์ค่าประสิทธิภาพของแบบจำลองการแนะนำหนังสือด้วยเทคนิคต่าง ๆ โดยใช้โปรแกรม RapidMiner Studio นั้นแสดงดังตาราง 16 ซึ่งจะพบว่าค่าความแม่นยำที่ได้ไม่ว่าจะด้วยเทคนิคใดก็ตามมีค่าค่อนข้างน้อย ผู้วิจัยจึงศึกษาบทความอ้างอิงเพิ่มเติมจึงพบว่าค่าความแม่นยำในบทความอ้างอิง ถึงแม้ว่าชุดข้อมูลที่ใช้จะไม่ใช่ชุดข้อมูลเดียวกัน แต่ได้ผลลัพธ์ของค่าความแม่นยำเป็นในทิศทางเดียวกันคือค่าความแม่นยำที่ได้มีค่าน้อยมาก เนื่องจากผลลัพธ์ที่แบบจำลองให้มานั้นอยู่ในรูปของการแนะนำหนังสือให้แก่ผู้ใช้งานแต่ละคนจำนวน 15 อันดับ โดยหนังสือที่

แนะนำนั้นจะเป็นหนังสือเล่มใหม่ที่ไม่เคยอยู่ในประวัติความชื่นชอบของผู้ใช้คนนั้น ๆ ซึ่งสามารถมีความเป็นไปได้ที่จะไม่พบว่าเป็นข้อมูลที่เกี่ยวข้องกับผู้ใช้งาน กล่าวคือเมื่อนำข้อมูลการแนะนำหนังสือทั้ง 15 อันดับไปเปรียบเทียบกับชุดข้อมูลความเกี่ยวข้องที่แบบจำลองเก็บไว้แล้วมีความเป็นไปได้ว่าหนังสือทั้ง 15 อันดับนั้นมีความเกี่ยวข้องกับผู้ใช้มีจำนวนน้อยมาก จึงเป็นผลให้ค่าความแม่นยำที่จะพิจารณาจากจำนวนข้อมูลที่เกี่ยวข้องกับผู้ใช้นั้นให้ค่าที่ค่อนข้างน้อย

ในบทความศึกษาวิจัยนี้นำเสนอการสร้างแบบจำลองการแนะนำหนังสือด้วยเทคนิคเหมืองข้อมูล ด้วยวิธี Collaborative Filtering โดยใช้ส่วนขยายที่มีชื่อว่า Recommender ของโปรแกรม RapidMiner Studio และเปรียบเทียบผลลัพธ์ประสิทธิภาพของแบบจำลองการแนะนำด้วยเทคนิคต่าง ๆ ซึ่งประโยชน์ของโปรแกรม RapidMiner Studio นั้น เป็นโปรแกรมที่มีเครื่องมือที่รองรับการทำงานเหมืองข้อมูล ตั้งแต่การจัดเตรียมชุดข้อมูลไปจนถึงการวัดประสิทธิภาพของแบบจำลอง โดยช่วยให้ผู้ใช้สามารถเห็นภาพรวมของการการสร้างแบบจำลองได้ง่าย ด้วยการเลือกใช้โอเปอร์เรเตอร์ที่เหมาะสมกับการสร้างแบบจำลองแล้วนำมาจัดเรียงคล้ายกับในรูปแบบของ Work Flow อีกทั้งยังประมวลผลลัพธ์ออกมาได้อย่างรวดเร็ว ทำให้สามารถปรับปรุงแบบจำลองให้ได้ผลลัพธ์ประสิทธิภาพที่เหมาะสมกับชุดของมูลที่ใช้ได้อย่างรวดเร็ว

5.2 **ข้อเสนอแนะ**

ในการศึกษวิจัยนี้เป็นการศึกษาวิจัยด้วยวิธี Collaborative Filtering หรือวิธีการพิจารณาโดยอ้างอิงข้อมูลจากผู้ใช้คนอื่น ๆ ในระบบเพียงอย่างเดียว แต่ยังมีวิธีที่ใช้ในการสร้างแบบจำลองการแนะนำอีกสองวิธีคือวิธี Content-based Filtering หรือการพิจารณาด้วยข้อมูลคุณลักษณะของสินค้า และวิธี Hybrid ที่เป็นวิธีการผสมทั้งสองวิธีข้างต้นเข้าด้วยกัน ซึ่งมีความเป็นไปได้ว่าทั้งสองวิธีที่ไม่ได้เลือกใช้นี้อาจจะให้ผลลัพธ์ประสิทธิภาพของแบบจำลองการแนะนำที่ดีกว่า และหากในอนาคตสามารถนำผลลัพธ์การแนะนำหนังสือ top-n ของผู้ใช้แต่ละคนที่ได้จากแบบจำลองที่สร้างด้วยโปรแกรม RapidMiner Studio ไปเป็นค่าจริงที่ใช้ในการเปรียบเทียบความแม่นยำของประสิทธิภาพของแบบจำลองการแนะนำหนังสือได้ก็จะทำให้สามารถเพิ่มความเชื่อมันในผลลัพธ์ที่ได้จากแบบจำลองมากยิ่งขึ้น

บรรณานุกรม

Bosnjak, M., Antulov-Fantulin, N., !`muc, T., & Gamberger, D. (2011). *Constructing recommender systems workflow templates in RapidMiner*.

Emma Grimaldi. (2018). How to build a content-based movie recommender system with Natural Language. https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243

Foxtrot. (2017). goodbooks-10k. https://www.kaggle.com/zygmunt/goodbooks-10k

Glen, S. (2019). ROC Curve Explained in One Picture.

Jain, A., & Vishwakarma, S. K. (2017). Collaborative Filtering for Movie Recommendation using RapidMiner. *International Journal of Computer Applications, 169*, 29-33.

Mihelcic, M., Antulov-Fantulin, N., Bosnjak, M., & !`muc, T. (2012). *Extending RapidMiner with recommender systems algorithms*.

Sirinart Tangruamsub. (2562). Recommendation System https://medium.com/@sinart.t/recommendation-system-%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%AA%E0%B8%A3%E0%B8%B8%E0%B8%9B%E0%B9%80%E0%B8%AD%E0%B8%B2%E0%B9%80%E0%B8%AD%E0%B8%87-ce6246f49754

Torstensson, E. (2019). Comparing neighborhood and matrix factorization models in recommendation systems: Saving the user some clicks. In.

Wen, Z. (2015). *Recommendation system based on collaborative filtering in RapidMiner Zhihang Tang*.

ไกรศักดิ์ เกษร. (2558). ระบบค้นคืนสารสนเทศ : แนวคิดและแนวทางการพัฒนาในอนาคต (*Information Retrieval System: Concepts and Future Directions)* (Vol. 1). พิษณุโลก: โฟกัส พริ้นติ้ง: ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร อ.เมือง จ.พิษณุโลก 65000.

นงคราญ คำวิชัย. (2559). *Practical Data Mining With RapidMiner Studio 7*.

นภวรรณ ดุษฎีเวทกุล. (2560). การปรับปรุงวิธีการกรองร่วมสำหรับระบบแนะนำด้วยข้อมูลความสัมพันธ์จากสื่อสังคมออนไลน์. In ท. วิรัตน์ จารีวงศ์ไพบูลย์ (Ed.): มหาวิทยาลัยธรรมศาสตร์.

บุญเสริม กิจศิริกุล. (2546). อัลกอริทึมการทำเหมืองข้อมูล. https://www.cp.eng.chula.ac.th/~boonserm/publication/AlgoDataMining.pdf

เบญจพร เอี่ยมประโคน, & ณัตติฤดี เจริญรักษ์. (2018). วิธี การ เปรียบเทียบ พื้นที่ ใต้ โค้ง ROC สำหรับ ข้อมูล ชุด เดียวกัน: กรณี ศึกษา แบบ จำลอง คะแนน เครดิต.

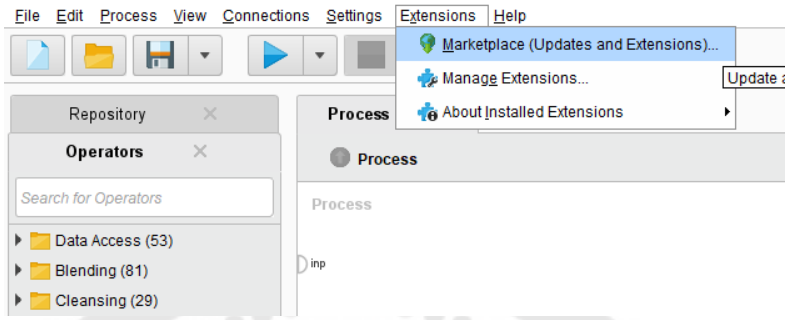
เปรมชัยสวัสดิ์, น. (2020). ระบบผู้แนะนำแบบหลายเกณฑ์จากข้อมูลแบบไฮบริด. วารสารสุทธิปริทัศน์, *26*(80), 111-128.

พิจิตรา, จ. (2017). การ เพิ่ม ประสิทธิภาพ ระบบ เครือ ข่าย สังคม ด้าน รูปภาพ ด้วย เทคนิค การ จัด อันดับ ใหม่ แบบ ร่วม. *Veridian E-journal Science and Technology Silpakorn University, 4*(3), 21-35.

ภาคผนวก

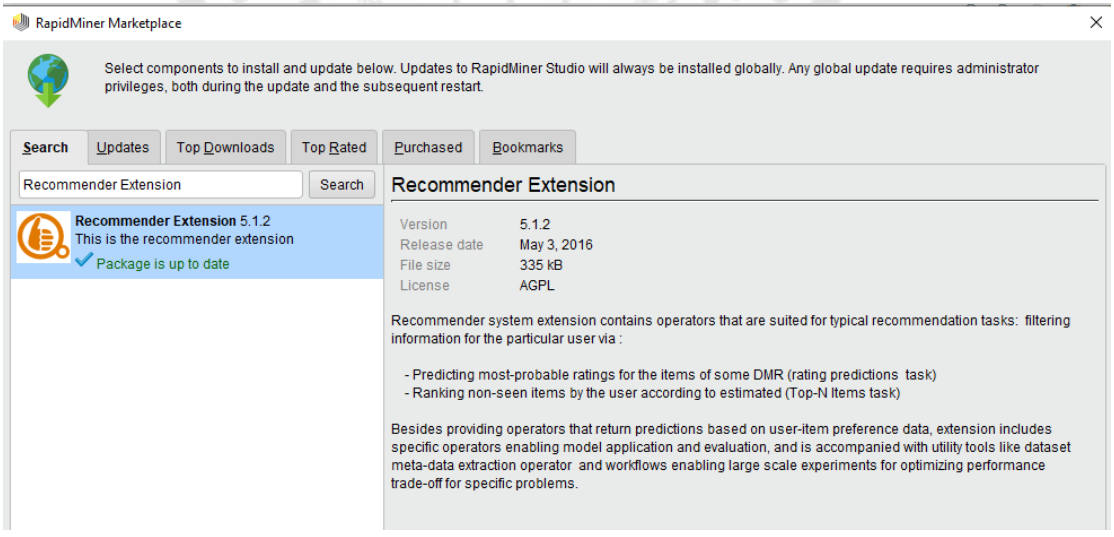
การติดตั้งส่วนเสริม Recommenders Version 5.1.2 สำหรับเลือกใช้โอเปอร์เรเตอร์ที่เกี่ยวข้องกับการแนะนำ โดยมีขั้นตอนการติดตั้ง ดังนี้

1. ไปที่เมนู Extension และคลิกเลือก Marketplace



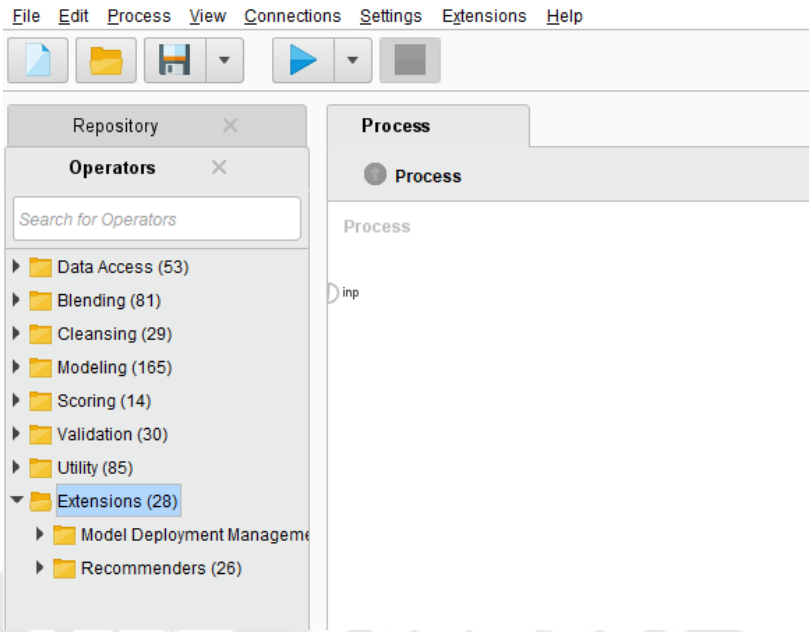
ภาพประกอบ 48 แสดงภาพเมนู Marketplace ในซอฟต์แวร์ RapidMiner

2. ค้นหาส่วนขยายโดยพิมพ์ Recommender Extension ที่แถบค้นหา
3. จากนั้นจะพบส่วนขยาย Recommender
4. คลิกที่ปุ่ม Install package เพื่อดำเนินการติดตั้งส่วนขยาย



ภาพประกอบ 49 แสดงหน้าดาวน์โหลดส่วนขยาย Recommender Extension

5. เมื่อดำเนินการติดตั้งเสร็จเรียบร้อยแล้ว ที่แถบเมนูทางด้านซ้ายมือจะปรากฏชุดส่วนขยาย Recommenders ภายใต้หัวข้อ Extensions



ภาพประกอบ 50 แสดงตำแหน่งของส่วนขยาย Recommender ในเมนูโอเปอร์เรเตอร์

ประวัติผู้เขียน

ชื่อ-สกุล	วรรษา เงินดี
วัน เดือน ปี เกิด	27 กรกฎาคม 2534
สถานที่เกิด	พิษณุโลก
วุฒิการศึกษา	พ.ศ. 2557 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ จาก มหาวิทยาลัยนเรศวร