

# การแนะนำหนังสือโดยใช้วิธีการกรองแบบร่วมมือ

## Book Recommendation based on Collaborative Filtering

ปริญญช ประเสริฐศิริกุล

pariyanuch.prasertsirikul@g.swu.ac.th

ภาคการศึกษาที่ 2, ประจำปีการศึกษา 2563

ภาควิชาวิทยาการคอมพิวเตอร์, คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร

### บทคัดย่อ

ปัจจุบันผู้คนต้องการความรู้ใหม่ๆ มากขึ้น นอกจากการได้รับความรู้แล้ว การเลือกซื้อหนังสือก็เป็นสิ่งที่สำคัญ ดังนั้นลูกค้าจึงให้ความสำคัญการเลือกซื้อหนังสือ โดยพิจารณาถึงรายละเอียดหนังสือมากขึ้น ทำให้การเลือกหนังสือที่ต้องการซื้อนั้นยังมีความยุ่งยาก ผู้สนใจจึงนำระบบแนะนำที่ใช้หลักการ Collaborative Filter ซึ่งให้ความสำคัญของข้อมูลรายการหนังสือ และคะแนนความคล้ายของรายการหนังสือกับความชื่นชอบของผู้ใช้งาน เพื่อหาคะแนนความคล้ายระหว่างผู้ใช้งานด้วยวิธี Cosine Similarity ดังนั้น ใช้ระบบฐานข้อมูลกราฟ Neo4j ช่วยนำข้อมูลไปสร้างแบบจำลองระบบแนะนำหนังสือ พบว่าสามารถแนะนำหนังสือได้ตรงตามความชอบของลูกค้าโดยอ้างอิงจากคะแนนหนังสือ อีกทั้งยังช่วยประหยัดเวลาในการค้นหาหนังสือที่ชื่นชอบ หรือตัดสินใจเลือกซื้อหนังสือ บทความนี้ผู้สนใจได้นำเสนอเพียงเทคนิคการวัดคะแนนความคล้ายระหว่างผู้ใช้งานด้วยวิธี Cosine Similarity เทคนิคนี้ยังคงมีข้อจำกัดคือ หากผู้ใช้งานไม่มีการให้คะแนนหนังสือในระบบ เมื่อนำข้อมูลนี้ไปสร้างรายการแนะนำหนังสือให้กับผู้ใช้งาน อาจจะส่งผลให้รายการแนะนำที่ไม่ตรงตามความชอบของผู้ใช้งาน

**คำสำคัญ** – Recommendation System, Collaborative Filtering, Cosine Similarity, Graph Database, Neo4j

### 1. บทนำ

หนังสือมีความสำคัญต่อชีวิตมนุษย์มาก ทำให้มีหนังสือจำนวนมากขึ้น โดยมีวัตถุประสงค์ที่แตกต่างกัน และผู้แต่งที่แตกต่างกัน เช่น หนังสือนวนิยาย หนังสือเพื่อการศึกษา และหนังสืองานอดิเรก นอกจากนี้ระบบแนะนำสามารถแนะนำโดยใช้

อินเทอร์เน็ตตามการกรอง การประมวลผลข้อมูลรายละเอียดของหนังสือยังสามารถให้ข้อมูลแก่ผู้ใช้งานที่ต้องการให้แนะนำหนังสือ เช่น เรื่องย่อ ประเภทหนังสือ สารบัญ คะแนนหนังสือ ผู้แต่ง [1]

ระบบแนะนำ (Recommender System) เป็นหนึ่งในเทคนิคที่สำคัญที่สุดที่ใช้ในการแนะนำข้อมูลเกี่ยวกับความต้องการของผู้ใช้งาน รวมถึงการบริการที่เกี่ยวข้องโดยวิเคราะห์การกระทำของผู้ใช้ [2] ระบบแนะนำสามารถใช้เทคนิคการทำนายการเชื่อมโยง (Link Prediction) เพื่อสร้างความสัมพันธ์ใหม่ในเครือข่ายที่มีโอกาสเกิดขึ้นในอนาคต หรือความสัมพันธ์ที่ไม่ได้สังเกต โดยทำนายได้จากความสัมพันธ์ที่มีอยู่ในปัจจุบัน [3] สำหรับระบบแนะนำนั้นมีเทคนิคหลัก 3 ประเภท [4] ได้แก่ เทคนิคการกรองแบบร่วมมือ (Collaborative Filtering) เทคนิคการพิจารณาจากเนื้อหา (Content based) และเทคนิคผสมผสาน (hybrid) โดยเฉพาะเทคนิคการกรองแบบร่วมมือที่ได้รับความนิยมนำมาใช้ในการสร้างการแนะนำ เนื่องจากเป็นเทคนิคที่สามารถหาผู้ใช้งานที่มีความคล้ายกันได้ดี การวัดความคล้ายระหว่างผู้ใช้งานมีหลายเทคนิค มีเทคนิคหนึ่งที่ได้รับการใช้งานอย่างแพร่หลาย ได้แก่ Cosine Similarity, Pearson Correlation Coefficient, Euclidean Distance, Jaccard Coefficient [5]

ทั้งนี้ผู้สนใจจึงมีแนวคิดที่จะนำเทคนิคการสร้างระบบแนะนำมาแก้ปัญหาดังกล่าวด้วยการหาความคล้ายระหว่างผู้ใช้งานด้วยวิธี Cosine Similarity โดยใช้ Neo4j ซึ่งเป็นฐานข้อมูลประเภทกราฟ (Graph Database Management System) ที่นิยมในปัจจุบัน และใช้ CQL (Cypher Query Language) ในการ Query ข้อมูลต่างๆ สำหรับบทความนี้ชุดข้อมูล (Dataset) ที่

นำมาทดลองกับระบบแนะนำ ซึ่งเป็นข้อมูลการให้คะแนนหนังสือจาก goodreads.com ที่ทาง Kaggle จัดให้ [6] เพื่อสร้างแบบจำลองระบบแนะนำหนังสือที่สามารถแนะนำหนังสือได้ตรงตามความชอบของลูกค้าโดยอ้างอิงจากคะแนนหนังสือ อีกทั้งยังช่วยประหยัดเวลาในการค้นหาหนังสือที่ชื่นชอบ หรือตัดสินใจเลือกซื้อหนังสือ

## 2. งานวิจัยที่เกี่ยวข้อง

Rofeca Giri Rymmai และ Saleema JS [7] ได้พัฒนาระบบแนะนำหนังสือด้วยวิธีการวัดความคล้ายแบบ Cosine และมีการประยุกต์ใช้การทำเหมืองข้อความ เนื่องจากหนังสือมักอ้างอิงตามผู้แต่งประเภทและการจัดอันดับหนังสือ หนังสือแต่ละเล่มมีบทสรุป ซึ่งแตกต่างจากคำอธิบายในส่วนที่สำคัญของหนังสือ จึงค้นหาคำศัพท์ในหนังสือสองเล่มโดยใช้การคำนวณความคล้ายกันแบบ Cosine พบว่าสามารถแนะนำหนังสือให้กับลูกค้าด้วยวิธีการนี้

RamniHarbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula และ Gaurav Srivastav [8] ได้พัฒนาระบบแนะนำภาพยนตร์โดยใช้การกรองเนื้อหา (Content Based Filtering) ในระบบแนะนำภาพยนตร์ อัลกอริทึม KNN ถูกนำไปใช้ในระบบพร้อมกับหลักการของความคล้ายกันด้วย Cosine พบว่ามีความแม่นยำมากกว่าการวัดแบบอื่น ๆ ทำให้สามารถแนะนำภาพยนตร์ด้วยการวัดความคล้ายด้วย Cosine และอัลกอริทึม KNN

Neda Rajabpour, Amirmahdi Mohammadighavam, Ali NaserasadiAli และ NaserasadiMajid Estilayee [9] ได้พัฒนาระบบแนะนำอาหารนักท่องเที่ยวโดยใช้วิธีการกรองความร่วมมือ (Collaborative Filtering) นักท่องเที่ยวส่วนใหญ่ไม่คุ้นเคยกับอาหารของประเทศที่พวกเขาเดินทางไป มีความเป็นไปได้ว่าพวกเขาเลือกอาหารชนิดหนึ่งที่พวกเขาไม่ชอบหรือเป็นอันตรายต่อสุขภาพ ระบบนี้จึงช่วยแก้ปัญหาดังกล่าว พบว่าสามารถแนะนำอาหารพิเศษให้กับนักท่องเที่ยวที่ควบคุมอาหาร ซึ่งแสดงให้เห็นว่ามีความถูกต้องของระบบคือ 86.3%

ทัศนวรรณ แก้วใส และ สุพจน์ นิตยสุวัฒน์ [10] ได้พัฒนาระบบแนะนำภาพยนตร์ โดยใช้เทคนิคการกรองร่วมมือ

(Collaborative Filtering) ร่วมกับวิธีเคมีน (K-Mean) หรือการทำเหมืองข้อมูลเพื่อแบ่งกลุ่มข้อมูลก่อนใช้เทคนิคตัวกรองเชิงร่วมมือ สามารถช่วยลดขนาดของข้อมูลได้ และสามารถแนะนำภาพยนตร์ให้กับผู้ใช้งานระบบได้รวดเร็ว และสร้างความพึงพอใจให้กับผู้ใช้งานต่อระบบอยู่ในระดับดี

ธนพล พุกเส่ง, สุชา สมานชาติ และ สุนันทา สดสี [11] ได้ศึกษางานวิจัยทางด้านระบบผู้แนะนำที่ได้นำความไว้วางใจและผู้เชี่ยวชาญมาเป็นส่วนประกอบ ตลอดจนการนำทั้งสองเทคนิคมาใช้งานร่วมกันเพื่อพัฒนาระบบผู้แนะนำ พบว่าสามารถแก้ปัญหาแบบผู้แนะนำและปรับปรุงประสิทธิภาพกระบวนการแนะนำได้เป็นอย่างดี

วรนุช ศรีพลัง และ วงกต ศรีอุไร [12] ได้พัฒนาต้นแบบของระบบแนะนำข้อมูลสำหรับห้องสมุดออนไลน์โดยใช้วิธีการกรองข้อมูลแบบพึ่งพาผู้เข้าร่วม (Collaborative Filtering) และข้อมูลส่วนบุคคล (User Profile) พบว่าระบบสามารถแนะนำหนังสือที่คาดว่าจะตรงกับความต้องการของผู้ใช้ได้ และได้รับความพึงพอใจเป็นอย่างดีจากผู้ใช้งาน

## 3. แนวคิดและทฤษฎีที่เกี่ยวข้อง

### 3.1 ระบบแนะนำ (Recommender System)

ระบบแนะนำ เป็นระบบที่ถูกนำมาใช้เพื่อแนะนำเสนอสินค้าและบริการที่คาดว่าจะน่าสนใจ โดยอ้างอิงจากสมมติฐานการเรียนรู้ข้อมูลความชอบหรือความต้องการ เช่น ระบบแนะนำในการเลือกซื้ออาหาร ซีดีเพลง หรือผลิตภัณฑ์อื่นๆ ประกอบด้วย 4 ส่วนหลัก [13] ดังนี้

1. ส่วนข้อมูลพื้นฐานที่จำเป็นต้องใช้ในการประมวลผล เช่น Profile ของผู้ใช้งานแต่ละคน เป็นต้น
2. ส่วนการป้อนข้อมูล เป็นข้อมูลที่ได้จากการป้อนข้อมูลเข้ามาของผู้ใช้งาน เช่น การให้คะแนนสินค้า ซึ่งมีอยู่ 2 แบบ คือแบบชัดเจน (Explicit) เช่น ระดับความนิยมตั้งแต่ 1 ถึง 5 เป็นต้น และแบบไม่ชัดเจน (Implicit) เช่น ประวัติการใช้งานของระบบในอดีต เป็นต้น
3. ส่วนของอัลกอริทึม เป็นส่วนสำคัญที่สุดที่ใช้ในการประมวลผลข้อมูล เพื่อให้การแนะนำข้อมูลออกมาให้ตรงกับความต้องการ

ต้องการของลูกค้ามากที่สุด โดยวิธีหลักๆ ได้แก่ Collaborative Recommendation, Contented-Based Recommendation

4. ส่วนของการนำเสนอคำแนะนำหรือนำเสนอสินค้าหรือบริการที่ตรงกับความต้องการของผู้ใช้มากที่สุด

### 3.2 การกรองแบบร่วมมือ (Collaborative Filtering)

Collaborative filtering เป็นการแนะนำสินค้าหรือบริการ โดยการพิจารณาความคล้ายคลึงกันของผู้ใช้งานที่เป็นเป้าหมายในการแนะนำกับผู้ที่มีความชื่นชอบคล้ายกัน ซึ่งความคล้ายกันระหว่างบุคคลนั้นสามารถคำนวณได้จากความคล้ายกันของลักษณะการให้คะแนนสินค้าหรือบริการในอดีต [11]

### 3.3 การคำนวณคะแนนความคล้าย (Similarity measure) ด้วยวิธี Cosine Similarity

Cosine Similarity เป็นการหาค่าความคล้ายด้วยองศา ระหว่างผู้ใช้งาน 2 คน ซึ่งเป็นวิธีที่นิยมใช้ในการคำนวณหาค่าความคล้าย ดังสมการที่ 1

$$\text{Cosine Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

จากสมการข้างต้น Cosine Similarity หมายถึง ค่าความคล้ายระหว่างผู้ใช้งาน  $A$  กับผู้ใช้งาน  $B$  สำหรับ  $A_i$  และ  $B_i$  ซึ่งเป็นค่าคะแนนของผู้ใช้งาน  $A$  กับผู้ใช้งาน  $B$  ที่ได้ให้ไว้ในสินค้า  $i$  ซึ่ง  $i = \{1, \dots, n\}$  โดยที่  $n$  หมายถึง จำนวนสินค้าทั้งหมด ค่าความคล้ายที่ได้จะมีค่าอยู่ระหว่าง 0 ถึง 1 หากค่าความคล้ายใกล้เคียงเท่ากับ 1 แสดงว่าผู้ใช้งาน  $A$  และผู้ใช้งาน  $B$  มีความคล้ายกันมาก ส่วนค่าความคล้ายที่ได้เท่ากับ 0 หมายถึง ผู้ใช้งาน  $A$  และผู้ใช้งาน  $B$  ไม่มีความคล้ายกันใดๆ [12]

### 3.4 ระบบการจัดการฐานข้อมูลประเภทกราฟโดยใช้ Neo4j (A Graph Database Management System Developed by Neo4j)

Neo4j เป็นหนึ่งในฐานข้อมูลประเภทกราฟ (Graph Database) ที่นิยมในปัจจุบัน ซึ่งถูกพัฒนามาบนพื้นฐานภาษา JAVA และใช้ภาษา CQL (Cypher Query Language) ในการ Query ข้อมูลต่างๆ เพื่ออำนวยความสะดวกในการวิเคราะห์ความสัมพันธ์ของข้อมูลเชิงลึก องค์ประกอบที่สำคัญของ

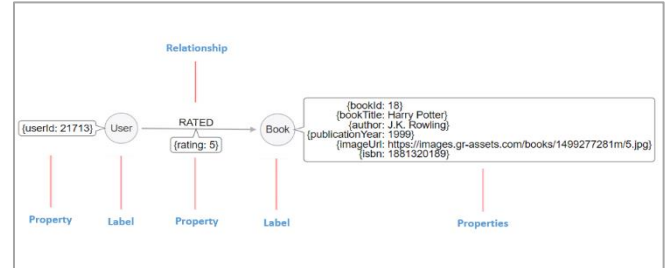
ฐานข้อมูลประเภทกราฟ คือแบบจำลองกราฟ (Graph Model) [14] ประกอบ 3 ส่วน ดังนี้

1.1 โหนด (Node) เป็นส่วนที่เก็บข้อมูลต่างๆ ไว้ภายในเป็นวัตถุ

1.2 คุณสมบัติ (Property) เป็นสิ่งที่ถูกบรรจุอยู่ในโหนด เพื่อเก็บข้อมูลต่างๆ

1.3 ความสัมพันธ์ (Relationship) เป็นความสัมพันธ์ระหว่างโหนด 2 โหนด

ในระบบฐานข้อมูลประเภทกราฟจะใช้ภาษา Cypher ในการเรียกดูข้อมูล โดยมีการกำหนดชื่อโหนดในวงเล็บ ได้แก่ (:User), (:Book) และภายในโหนดมีการกำหนดป้ายชื่อ (Label) และคุณสมบัติ (Properties) ในแต่ละ Node เช่น โหนดผู้ใช้งาน (u:User{userId: '21713'}), โหนดหนังสือ (b:Book{bookTitle: 'Harry Potter'}) ส่วนความสัมพันธ์จะใช้สัญลักษณ์ [ ] เชื่อมกับโหนด ทิศทางความสัมพันธ์จะใช้สัญลักษณ์ “-” หรือ “--” หรือเส้นลูกศร “->” ในการระบุทิศทางความสัมพันธ์ระหว่างโหนดได้ เช่น (u:User)-[:RATED]->(b:Book) [14] ดังรูปที่ 1



รูปที่ 1 ตัวอย่างการสร้างแบบจำลองกราฟแสดงโหนดและความสัมพันธ์

ส่วนการเขียนคำสั่ง Cypher หลักในรูปแบบต่างๆ [14] ดังตารางที่ 1

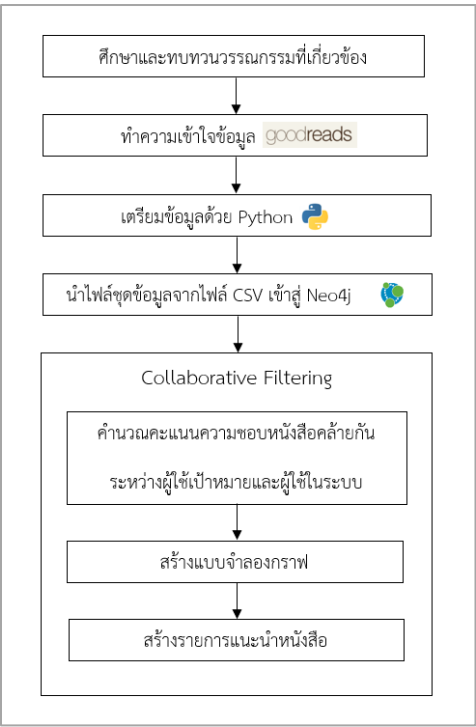
คำสั่ง Cypher	ความหมาย
1. CREATE	เพิ่มโหนด ความสัมพันธ์ และคุณสมบัติ
2. MATCH	เรียกข้อมูลจากโหนด ความสัมพันธ์ และคุณสมบัติ
3. RETURN	ส่งค่าผลลัพธ์กลับจากการเรียกข้อมูล
4. WHERE	ตั้งเงื่อนไขในการเรียกข้อมูล
5. DELETE	ลบโหนด ความสัมพันธ์
6. REMOVE	ลบคุณสมบัติของโหนด และความสัมพันธ์
7. ORDER By	เรียงข้อมูลที่เรียกมา

คำสั่ง Cypher	ความหมาย
8. SET	เพิ่มหรือปรับปรุงป้ายชื่อโหนด

ตารางที่ 1 คำสั่งต่างๆ ของ Cypher

4. วิธีดำเนินการวิจัย

ผู้สนใจได้ออกแบบโครงสร้างของระบบแนะนำหนังสือโดยใช้วิธี Collaborative Filtering ซึ่งมีขั้นตอนการดำเนินงานดังรูปที่ 2



รูปที่ 2 โครงร่างของการสร้างระบบแนะนำหนังสือ

4.1 การศึกษาและทบทวนวรรณกรรมที่เกี่ยวข้อง (Literature Review)

ผู้สนใจได้ทำการศึกษาทฤษฎีและศึกษางานวิจัยที่เกี่ยวข้องกับระบบแนะนำ (Recommender System) เทคนิคการทำนายการเชื่อมโยง (Link Prediction) การกรองแบบร่วมมือ (Collaborative Filtering) และระบบการจัดการฐานข้อมูลประเภทกราฟโดยใช้ Neo4j (A Graph Database Management System Developed by Neo4j)

4.2 การทำความเข้าใจข้อมูล (Data Understanding)

สำหรับบทความนี้ใช้ชุดข้อมูลเกี่ยวกับการให้คะแนนหนังสือจากเว็บไซต์ Goodread.com โดยมีหนังสือจำนวน

10,000 เล่ม ที่ผู้ใช้งานจำนวน 53,434 คน ให้คะแนนจำนวน 981,756 ครั้ง (1-5 คะแนน) ชุดข้อมูลนี้ได้รับการทำความสะอาดโดยผู้ใช้งาน SahilkK จาก Kaggle ผู้สนใจจึงเลือกไฟล์ที่ใช้สำหรับการสร้างแบบจำลองกราฟ ซึ่งประกอบด้วยไฟล์ Books.csv เป็นข้อมูลรายละเอียดหนังสือ และไฟล์ Ratings.csv เป็นข้อมูลการให้คะแนนหนังสือโดยผู้ใช้งานเท่านั้น

4.3 การเตรียมข้อมูล (Data Preparation)

ขั้นตอนนี้เป็นการปรับข้อมูลดิบทั้งหมดให้เป็นข้อมูลที่พร้อมจะเข้าสู่โปรแกรม Neo4j จากนั้นใช้ภาษา Python เพื่อสำรวจข้อมูลว่ามีค่าที่หายไปหรือไม่ ทำการรวมไฟล์ Book.csv และไฟล์ Rating.csv เข้าด้วยกัน แล้วเลือกคุณลักษณะที่ใช้ และถูกกำหนดชื่อคุณลักษณะใหม่ เพื่อให้ได้ไฟล์ CSV ที่พร้อมเข้าสู่ฐานข้อมูลประเภท Neo4j ต่อไป ดังตารางที่ 2

คุณลักษณะของข้อมูล	ความหมาย
1. userID	รหัสประจำตัวผู้ใช้งาน
2. rating	คะแนนที่ผู้ใช้งานให้กับหนังสือ (1-5 คะแนน)
3. bookId	รหัสประจำตัวหนังสือ
4. bookTitle	ชื่อเรื่องหนังสือ
5. author	ผู้แต่งหนังสือ
6. publicationYear	ปีที่ตีพิมพ์หนังสือ
7. imageUrl	ลิงค์ในรูปภาพ
8. isbn	เลขมาตรฐานสากลประจำหนังสือ

ตารางที่ 2 รายละเอียดคุณลักษณะต่างๆ ของข้อมูลการให้คะแนนหนังสือ

4.4 การนำไฟล์ชุดข้อมูลจากไฟล์ CSV เข้าสู่ Neo4j (Importing CSV Data into Neo4j)

หลังจากเตรียมข้อมูลเสร็จ นำไฟล์ชุดข้อมูล CSV เข้าสู่ Neo4j และถูกจัดให้อยู่ในรูปแบบกราฟประกอบด้วยโหนดความสัมพันธ์ และคุณสมบัติที่กำหนดดังตารางที่ 3

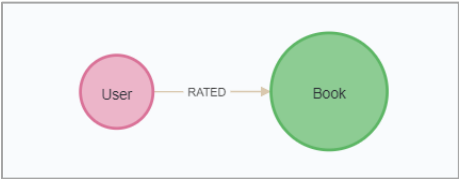
ส่วนประกอบของกราฟ	คุณสมบัติ (Property)
1. โหนด User (สีชมพู)	userId
2. โหนด Book (สีเขียว)	bookId, bookTitle, author, publicationYear, imageUrl, isbn
3. ความสัมพันธ์ RATED แบบมีทิศทางจากโหนด User ไปยังโหนด Book (สีน้ำตาล)	rating

ตารางที่ 3 ส่วนประกอบต่างๆ ของกราฟ

โดยเขียนคำสั่ง Cypher แสดงแบบจำลองกราฟแสดงความสัมพันธ์แบบมีทิศทางจากโหนดผู้ใช้งานไปยังโหนดหนังสือ ดังรูปที่ 3 และได้ผลลัพธ์คำสั่งดังรูปที่ 4

```
call db.schema.visualization
```

รูปที่ 3 คำสั่ง Cypher การสร้างกราฟแสดงโหนดและความสัมพันธ์

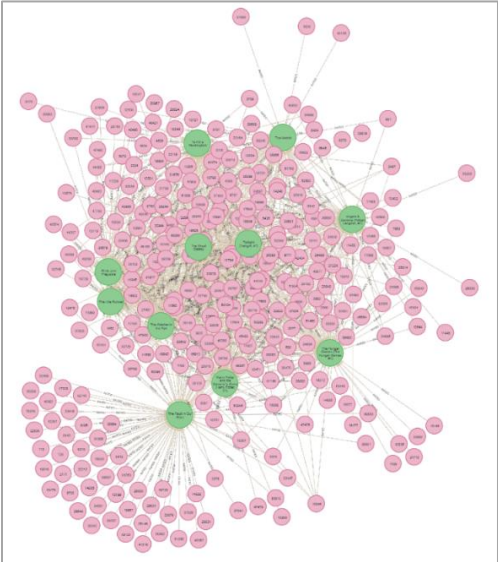


รูปที่ 4 ผลลัพธ์คำสั่ง Cypher การสร้างกราฟแสดงโหนดและความสัมพันธ์

ต่อไปจะแสดงคำสั่ง Cypher การนำเข้าชุดข้อมูล CSV เข้าสู่ Neo4j เป็นจำนวน 2,000 แถว ดังรูปที่ 5 และได้ผลลัพธ์คำสั่งดังรูปที่ 6

```
LOAD CSV WITH HEADERS FROM "file:///goodreadDataset.csv" as row
MERGE (u:User {userId:toInteger(row.userId)})
MERGE (b:Book {bookTitle:toInteger(row.bookId),
bookTitle:row.bookTitle, author:row.author,
publicationYear:row.publicationYear, imageUrl:row.imageUrl,
isbn:row.isbn})
MERGE (u)-[r:RATED{rating:toInteger(row.rating)}]->(b)
RETURN u, b
LIMIT 2000
```

รูปที่ 5 คำสั่ง Cypher ในการสร้างกราฟแสดง  
เครือข่ายการให้คะแนนหนังสือของผู้ใช้งาน



รูปที่ 6 ผลลัพธ์คำสั่ง Cypher การสร้างกราฟแสดง  
เครือข่ายการให้คะแนนหนังสือของผู้ใช้งาน

พบว่ากราฟแสดงเครือข่ายการให้คะแนนหนังสือของผู้ใช้งาน มีโหนดผู้ใช้งาน 289 โหนด, โหนดหนังสือ 11 โหนด และจำนวนความสัมพันธ์การให้คะแนนหนังสือ 1,099 เส้น

#### 4.5 การคำนวณคะแนนความชอบหนังสือคล้ายกันระหว่าง ผู้ใช้เป้าหมายและผู้ใช้ในระบบ (Calculating the similarity between users)

ต่อไปจะหาคะแนนความชอบหนังสือคล้ายกันระหว่างผู้ใช้งานด้วยวิธี Cosine similarity เช่น ผู้ใช้งาน user21713 และผู้ใช้งาน user41074 มีความชอบหนังสือคล้ายกันหรือไม่ สามารถเขียนคำสั่ง Cypher ดังรูปที่ 7 และผลลัพธ์คำสั่งดังรูปที่ 8

```
MATCH (u1:User {userId: 21713})-[r1:RATED]->(b1:Book)->[r2:RATED]-(u2:User {userId: 41074})
RETURN u1.userId AS user21713,
u2.userId AS user41074,
collect(b1.bookTitle) as ratedBook,
collect(r1.rating) AS user21713Rating,
collect(r2.rating) AS user41074Rating,
round(gds.alpha.similarity.cosine(collect(r1.rating), collect(r2.rating)),2)
AS CosineSimilarity
```

รูปที่ 7 คำสั่ง Cypher การคำนวณคะแนนความคล้ายกันระหว่างผู้ใช้งาน  
user21713 และผู้ใช้งาน user41074

user21713	user41074	ratedBook	user21713Rating	user41074Rating	CosineSimilarity
21713	41074	["Mockingjay (The Hunger Games, #3)", "Catching Fire (The Hunger Games, #2)", "The Hunger Games (The Hunger Games, #1)", "Harry Potter and the Prisoner of Azkaban (Harry Potter, #3)"]	[4, 4, 5, 4]	[3, 3, 3, 4]	0.98

รูปที่ 8 ผลลัพธ์คำสั่ง Cypher การคำนวณคะแนนความคล้ายกันระหว่าง  
ผู้ใช้งาน user21713 และผู้ใช้งาน user41074

พบว่าผู้ใช้งาน user21713 และผู้ใช้งาน user41074 มีความชอบหนังสือคล้ายกันด้วยคะแนนความคล้ายเท่ากับ 0.98

ต่อไปจะค้นหาผู้ใช้งานทั้งหมดที่ชอบหนังสือคล้ายกับผู้ใช้งาน user21713 โดยมีคะแนนความคล้ายมากที่สุด 5 ค่า ดังรูปที่ 9 และผลลัพธ์คำสั่งดังรูปที่ 10

```

MATCH (u2:User)
WITH u2
ORDER BY u2.userId

MATCH (u1:User {userId: 21713})-[r1:RATED]->(b1:Book)-[r2:RATED]-(u2:User)
WITH u1,
      u2,
      collect(r1.rating) AS u1Rating,
      collect(r2.rating) AS u2Rating,
      round(gds.alpha.similarity.cosine(collect(r1.rating),collect(r2.rating)),2)
AS CosineSimilarity
WHERE CosineSimilarity > 0.5

RETURN collect(u2.userId) AS similarUser,
       CosineSimilarity, COUNT(*) AS userNum
ORDER BY CosineSimilarity DESC
LIMIT 5

```

รูปที่ 9 คำสั่ง Cypher การสร้างรายการผู้ใช้งานทั้งหมดที่มีความคล้ายกับ  
ผู้ใช้งาน user21713

"similarUser"	"CosineSimilarity"	"userNum"
[314, 725, 1088, 1136, 2487, 2900, 3022, 5379, 5436, 5461, 6063, 62]	1.0	138
[52, 6634, 8167, 8440, 9011, 9246, 9731, 9771, 10111, 10146, 10249, 10509, 10751, 11239, 11285, 11408, 11868, 11945, 11999, 12874, 12946, 13000, 13360, 13826, 14222, 14285, 14546, 14603, 15054, 15318, 15604, 15889, 16587, 16913, 17242, 17434, 17566, 17643, 18100, 18199, 18316, 18957, 19942, 19984, 20467, 20782, 20991, 21217, 21487, 21676, 22534, 23576, 24389, 24834, 25182, 25214, 25278, 26053, 26398, 26661, 26750, 27612, 28767, 28824, 29644, 29689, 29703, 30184, 30313, 30601, 30681, 30833, 31305, 31760, 32058, 32419, 32918, 32923, 33207, 33716, 33872, 33890, 34531, 34688, 35259, 35982, 36031, 36099, 36695, 37041, 37153, 37746, 38080, 38082, 39423, 40126, 40251, 40490, 40566, 41240, 42404, 44243, 45493, 45554, 46421, 46977, 47730, 47746, 48291, 48440, 48559, 49288, 49289, 49295, 49297, 49830, 50101, 50102, 50104, 50133, 50610, 51166, 51480, 52007, 52469, 53145, 53292]	0.99	54
[1169, 1185, 2171, 3662, 3922, 5885, 6630, 7563, 8669, 9722, 10140, 10246, 10335, 10610, 10944, 11854, 12471, 13282, 13544, 14936, 16569, 17984, 18313, 18361, 19171, 19729, 20076, 20848, 21228, 22602, 23612, 24499, 26145, 26629, 26718, 30283, 30944, 32055, 32305, 32592, 33065, 33697, 33864, 37284, 38475, 41318, 43602, 43985, 44397, 45269, 47476, 47800, 48297, 53245]	0.98	20
[598, 3114, 11691, 13274, 16377, 17663, 18031, 19724, 24845, 25164, 28831, 29123, 37834, 41074, 48482, 49298, 51460, 51762, 51838, 52583]	0.97	9
[2077, 11927, 15494, 24326, 24582, 28158, 32635, 32748, 51065]	0.96	5
[439, 11599, 18798, 31001, 52036]		

รูปที่ 10 ผลลัพธ์คำสั่ง Cypher การสร้างรายการผู้ใช้งานทั้งหมดที่มีความคล้ายกับผู้ใช้งาน user 21713

พบว่าผู้ใช้งานทั้งหมดที่ชอบหนังสือคล้ายกับผู้ใช้งาน user21713 ด้วยคะแนนความคล้ายมากที่สุดเท่ากับ 1 จำนวน 138 คน รองลงมาคะแนนความคล้ายเท่ากับ 0.99 (จำนวน 54 คน), 0.98 (จำนวน 20 คน), 0.97 (จำนวน 9 คน) และ 0.96 (จำนวน 5 คน)

#### 4.6 การสร้างแบบจำลองกราฟ (Graph Modeling)

หลังจากสร้างความสัมพันธ์ความคล้ายระหว่างผู้ใช้งานเมื่อค่าคะแนนความคล้ายมากกว่า 0.5 และสร้างความสัมพันธ์การแนะนำ จากรูปที่ 10 ผู้สนใจจึงสุ่มเลือกผู้ใช้งาน user1185 และผู้ใช้งาน user21713 มีความชอบหนังสือคล้ายกันด้วยคะแนนความคล้ายเท่ากับ 0.99 เพื่อต้องการทราบว่าผู้ใช้งานสอง

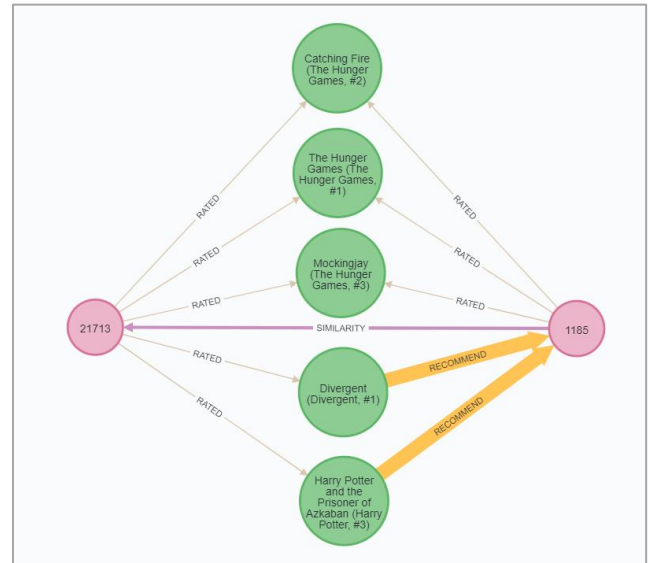
คนนี้ ให้คะแนนหนังสือเรื่องเดียวกันเรื่องอะไรบ้าง โดยสามารถเขียนคำสั่ง Cypher ดังรูปที่ 11 และผลลัพธ์คำสั่งดังรูปที่ 12

```

MATCH (u1:User{userId:21713})-[r:RATED]->(b:Book)
MATCH (u1:User)-[s:SIMILARITY]-(u2:User{userId:1185})
WHERE s.similarity > 0.5
RETURN u1, u2, b
LIMIT 5

```

รูปที่ 11 คำสั่ง Cypher การสร้างกราฟการแนะนำหนังสือให้กับผู้ใช้งาน user1185 ซึ่งมีความชอบคล้ายกับผู้ใช้งาน user21713



รูปที่ 12 ผลลัพธ์คำสั่ง Cypher การสร้างกราฟการแนะนำหนังสือให้กับผู้ใช้งาน user1185 ซึ่งมีความชอบคล้ายกับผู้ใช้งาน user21713

พบว่าผู้ใช้งาน user1185 และผู้ใช้งาน user21713 ให้คะแนนหนังสือเล่มเดียวกัน 3 เรื่อง ได้แก่ Catching Fire, The Hunger Games และ Mockingjay ดังนั้น แนะนำหนังสือเรื่อง Divergent และ Harry Potter and the Prisoner of Azkaban ให้กับผู้ใช้งาน user1185

#### 4.7 การสร้างรายการแนะนำหนังสือ (A List of Recommended Book)

ในขั้นตอนสุดท้าย จะค้นหารายการแนะนำหนังสือจำนวน 10 เล่ม ให้กับผู้ใช้งาน user1185 ซึ่งมีความคล้ายกับผู้ใช้งานทั้งหมดด้วยคะแนนความคล้ายมากกว่า 0.5 โดยเขียนคำสั่ง Cypher ดังรูปที่ 13 และผลลัพธ์คำสั่งดังรูปที่ 14



```
MATCH (u2:User{userId:1185})-[rec:RECOMMEND]-(b:Book)
RETURN b.bookTitle AS recommendBook, b.imageUrl AS imageUrl
ORDER BY recommendBook
LIMIT 10
```

รูปที่ 13 คำสั่ง Cypher การสร้างรายการแนะนำหนังสือให้กับ  
ผู้ใช้งาน user1185

	recommendBook	imageUrl
1	"1984"	"https://images.gr-assets.com/books/1348990566m/5470.jpg"
2	"Angels & Demons (Robert Langdon, #1)"	"https://images.gr-assets.com/books/1303390735m/960.jpg"
3	"Animal Farm"	"https://images.gr-assets.com/books/1424037542m/7613.jpg"
4	"Divergent (Divergent, #1)"	"https://images.gr-assets.com/books/1328559506m/13335037.jpg"
5	"Harry Potter and the Prisoner of Azkaban (Harry Potter, #3)"	"https://images.gr-assets.com/books/1499277281m/5.jpg"
6	"Harry Potter and the Sorcerer's Stone (Harry Potter, #1)"	"https://images.gr-assets.com/books/1474154022m/3.jpg"
7	"Pride and Prejudice"	"https://images.gr-assets.com/books/1320399351m/1885.jpg"
8	"The Catcher in the Rye"	"https://images.gr-assets.com/books/1398034300m/5107.jpg"
9	"The Diary of a Young Girl"	"https://images.gr-assets.com/books/1358276407m/48855.jpg"
10	"The Fault in Our Stars"	"https://images.gr-assets.com/books/1360206420m/11870085.jpg"

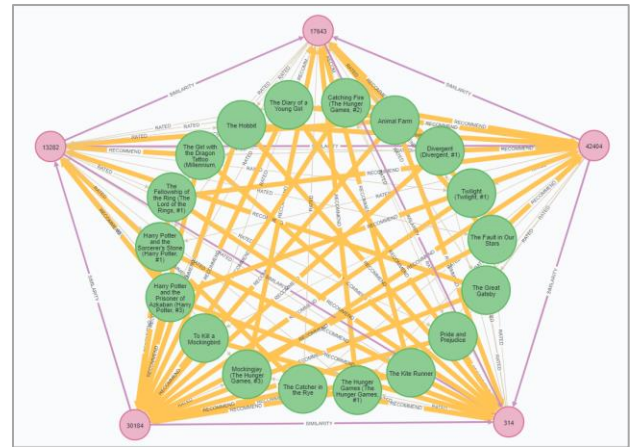
รูปที่ 14 ผลลัพธ์คำสั่ง Cypher การสร้างรายการแนะนำหนังสือให้กับ  
ผู้ใช้งาน user1185

## 5. ผลการวิจัย

จากการสร้างระบบแนะนำหนังสือโดยใช้ Neo4j ในการหาคะแนนความคล้ายกันระหว่างผู้ใช้งานด้วยวิธี Cosine Similarity พบว่าสามารถแนะนำหนังสือให้ตรงตามความชอบของผู้ใช้งานได้อย่างอิงจากคะแนนหนังสือ จึงแสดงกราฟสมบูรณ์ (Complete Graph) แสดงเครือข่ายการแนะนำหนังสือ โดยสามารถเขียนคำสั่ง Cypher ดังรูปที่ 15 และผลลัพธ์คำสั่งดังรูปที่ 16

```
MATCH (u1:User)-[r:RATED]->(b:Book)
MATCH (u1:User)-[s:SIMILARITY]-(u2:User)
WHERE s.similarity > 0.5
RETURN u1, u2, b
LIMIT 30
```

รูปที่ 15 คำสั่ง Cypher จากการสร้างกราฟแสดง  
เครือข่ายการแนะนำหนังสือ



รูปที่ 16 ผลลัพธ์คำสั่ง Cypher จากการสร้างกราฟแสดง  
เครือข่ายการแนะนำหนังสือ

## 6. สรุปผลการดำเนินงานวิจัย

บทความนี้นำเสนอการสร้างระบบแนะนำหนังสือโดยใช้วิธีการกรองข้อมูลร่วม และคำนวณค่าความคล้ายด้วยวิธี Cosine Similarity ผู้สนใจได้ทดลองชุดข้อมูลการให้คะแนนหนังสือจาก goodreads.com ซึ่งมีหนังสือจำนวน 10,000 เล่ม ที่ผู้ใช้งานจำนวน 53,434 คน ให้คะแนนจำนวน 981,756 ครั้ง (คะแนน 1-5) โดยมีขั้นตอนการวิจัยเพื่อแนะนำหนังสือที่ตรงตามความชอบให้กับผู้ใช้งาน ได้แก่ การศึกษาและทบทวนวรรณกรรมที่เกี่ยวข้อง การทำความเข้าใจข้อมูล การเตรียมข้อมูลโดยใช้ภาษา Python การนำไฟล์ชุดข้อมูลจากไฟล์ CSV เข้าสู่ Neo4j การสร้างรายการแนะนำหนังสือก็ต่อเมื่อคะแนนความคล้ายมีค่ามากกว่า 0.5 ตลอดจนถึงการสร้างกราฟสมบูรณ์แสดงเครือข่ายการแนะนำหนังสือ ทั้งนี้เป็นการสร้างระบบแนะนำหนังสือเบื้องต้นจากการใช้ Neo4j เป็นหลัก บทความนี้ได้นำเสนอเพียงเทคนิคการวัดความคล้ายระหว่างผู้ใช้งานด้วยวิธี Cosine Similarity เทคนิคนี้ยังคงมีข้อจำกัดคือ หากผู้ใช้งานไม่มีการให้คะแนนหนังสือในระบบ เมื่อนำข้อมูลนี้ไปสร้างรายการแนะนำหนังสือให้กับผู้ใช้งาน อาจจะส่งผลให้รายการแนะนำที่ได้ไม่ตรงตามความชอบของผู้ใช้งาน นอกจากนี้อาจจะมีเทคนิคอื่นๆ ที่ค่อนข้างเหมาะสมกับชุดข้อมูลนี้มากกว่า จึงควรมีการเปรียบเทียบกับเทคนิคการวัดคะแนนความคล้ายกันที่หลากหลาย เช่น Pearson Similarity, Euclidean Similarity จากการพิจารณาผลประโยชน์ประสิทธิภาพแต่ละเทคนิค เพื่อเลือกใช้เทคนิคที่เหมาะสมกับชุดข้อมูล

## 7. เอกสารอ้างอิง

- [1] D. Naga Jyothi, " Book Recommendation System using Neo4j Graph Database," *The International journal of analytical and experimental modal analysis*, vol. 12, no. 6, pp. 498-504, 2020.
- [2] Phonexay Vilakone, Doo-Soon Park, Khamphaphone Xinchang and Fei Hao, " An Efficient movie recommendation algorithm based on improved k-clique," *Vilakone et al. Hum. Cent. Comput. Inf. Sci.*, vol. 8, no. 38, pp. 1-15, 2018.
- [3] Victor Martinez, Fernando Berzal and Juan-Carlos Cubero, "A Survey of Link Prediction in Complex Networks," *ACM Computing Surveys*, vol. 49, no. 4, pp. 69:1-69:33, 2016.
- [4] Li, Y.M., Hsiao, H.W. and Lee, Y.L., "Recommending social network application via social filtering mechanisms," *Inf. Sci.*, Vols. 18-30, p. 239, 2013.
- [5] Sridhar Dilip Sondur, mit P Chigadani and Shantharam Nayak, " Similarity Measures for Recommender Systems: A Comparative Study," *Journal for Research*, vol. 2, no. 3, pp. 76-80, 2016.
- [6] SahilK, "https://www.kaggle.com/sahilkirpekar/goodreads10k-dataset-cleaned," 2020. [Online].
- [7] Rofeca Giri Rymmai and Saleema JS, " Book Recommendation using Cosine Similarity," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 276-281, 2017.
- [8] RamniHarbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula and Gaurav Srivastav, " Movie Recommendation System using Cosine Similarity and KNN," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 5, pp. 556-559, 2020.
- [9] Neda Rajabpour, Amirmahdi Mohammadighavam, Ali NaserasadiAli and NaserasadiMajid Estilayee, "TFR: A Tourist Food Recommender System based on Collaborative Filtering," *International Journal of Computer Applications*, vol. 181, no. 11, pp. 30-39, 2018.
- [10] ทศนวรรณ แก้วใส และสุพจน์ นิตย์สุวัฒน์, "ระบบแนะนำภาพยนตร์ด้วยเทคนิคตัวกรองเชิงร่วมมือร่วมกับวิธีเคมีน," ใน *การประชุมทางวิชาการระดับชาติ ด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 5*, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2552.
- [11] ธนพล พุกเส็ง, สุชา สมานชาติ และสุนันทา สดสี, "ระบบผู้แนะนำโดยอาศัยความไว้วางใจร่วมกับผู้เชี่ยวชาญ," *วารสารวิทยาศาสตร์และเทคโนโลยี*, ปีที่ 23, ฉบับที่ 13, หน้า 517-537, 2558.
- [12] วรณช ศรีพลัง และวงกต ศรีอุไร, "การพัฒนาระบบแนะนำข้อมูลสำหรับห้องสมุดออนไลน์โดยใช้วิธีการกรองข้อมูลแบบพึ่งพาผู้ใช้ร่วมและข้อมูลส่วนบุคคล," *วารสารมหาวิทยาลัยศรีนครินทรวิโรฒ (สาขาวิทยาศาสตร์และเทคโนโลยี)*, ปีที่ 9, ฉบับที่ 118, หน้า 150-164, 2560.
- [13] นลินี โสพัศสถิตย์, "การใช้ระบบแนะนำสนับสนุนการตัดสินใจ," มหาวิทยาลัยราชภัฏสวนสุนันทา, กรุงเทพมหานคร, 2555.
- [14] วุฒิชัย รมสายหยุด, นีโอโพรเจ : ฐานข้อมูลกราฟสำหรับการวิเคราะห์, กรุงเทพฯ: ซีเอ็ดดูเคชั่น, 2563.