



ระบบแนะนำวิดีโอเกมแบบไฮบริด

Hybrid Filtering based Video Game Recommendation System

นายจิรวัฒน์ ใจรัก

JIRAWAT JAIRUK

นายชนาภัทร ปราชญาวงศ์

CHANAPAT PRADYAWONG

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2561



ระบบแนะนำวิดีโอเกมแบบไฮบริด

Hybrid Filtering based Video Game Recommendation System

นายจิรวัฒน์ ใจรัก

JIRAWAT JAIRUK

นายชนาภัทร ปราชญาวงศ์

CHANAPAT PRADYAWONG

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2561



คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงการ

ระบบแนะนำวิดีโอเกมแบบไฮบริด

Hybrid Filtering based Video Game

Recommendation System

นิสิต

นายจิรวัฒน์ ใจรัก 58102010186

นายชนาภัทร ปราชญาวงศ์ 58102010189

ปริญญา

วิทยาศาสตร์บัณฑิต (วท.บ.)

ภาควิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการ

อ.ดร.วีรยุทธ เจริญเรืองกิจ

ลงชื่อ.....

(อ.ดร.วีรยุทธ เจริญเรืองกิจ)

อาจารย์ที่ปรึกษาโครงการ

## บทคัดย่อ

ระบบการแนะนำคือระบบที่แนะนำผลิตภัณฑ์และบริการแก่ลูกค้า การจัดอันดับผลิตภัณฑ์ของลูกค้าทั้งหมดและคุณสมบัติอื่นๆ จะถูกใช้เป็นข้อมูลเพื่อคาดการณ์สิ่งที่ลูกค้าต้องการ Collaborative Filtering (CF) และ Content-based Filtering นั้นทำงานได้ไม่ดีเมื่อใช้เทคนิคแยกต่างหาก งานวิจัยนี้จึงเสนอ ระบบแนะนำแบบไฮบริดของทั้งสองวิธี สำหรับระบบแนะนำวิดีโอเกม ในชุดข้อมูลการจัดอันดับเกมที่ได้รับจาก Amazon.com ในปี 2014 ซึ่งประกอบด้วยผู้ใช้ 168,291 คน และวิดีโอเกม 23,177 เกม รูปแบบแรกผสมผสาน CF Matrix Factorization และ Content-based Filtering รูปแบบที่สองเป็นการรวม CF Item-base และ CF User-base เข้ากับ Content-based Filtering โดยข้อมูล 25% ถูกเก็บไว้สำหรับการประเมินโมเดล เกณฑ์ Root Mean Square Error (RMSE) ใช้สำหรับการประเมิน และบรรลุความแม่นยำสูงสุดด้วย RMSE ที่ 1.7

## Abstract

Recommendation system is the system that recommends products and services to customers. The product rating of all customers and other features are used as information to predict what customers may like. The conventional Collaborative Filtering (CF) and Content-based Filtering approaches do not perform well when the individual technique is used separately. This research proposes machine learning models to create a hybrid of these two approaches for video game recommendation system on the game rating dataset obtained from Amazon.com in 2014, which consists of 168,291 users and 23,177 games. The first model combines CF Matrix Factorization and the Content-based Filtering techniques. The second combines CF Item-base and CF User-base combining with Content-based Filtering techniques. The 25% of data are holded out for evaluation of the models. The criteria Root Mean Square Error (RMSE) are used for evaluation and achieve the best accuracy with RMSE at 1.7.

## กิตติกรรมประกาศ

การดำเนินโครงการพัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด (Hybrid Filtering based Video Game Recommendation System) จะไม่ประสบความสำเร็จได้ หากปราศจากบุคคลดังต่อไปนี้

อ.ดร.วีรยุทธ เจริญเรืองกิจ อาจารย์ที่ปรึกษาโครงการที่ให้ความเมตตาปรารภนาดี และกรุณาเสียสละเวลาเพื่อให้คำปรึกษา แนะนำ ตลอดจนการตรวจแก้ไขข้อบกพร่องในการทำโครงการ รวมทั้งให้กำลังใจแก่ผู้วิจัยเสมอมา ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาและขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอกราบขอบพระคุณคณาจารย์ทุกท่านในสาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ที่ได้ประสิทธิ์ประสาทวิชาความรู้ทำให้ผู้วิจัยสามารถนำความรู้นั้นมาเป็นประโยชน์ในการทำโครงงานฉบับนี้

ขอกราบขอบพระคุณสมาชิกในครอบครัวของผู้วิจัย ที่สนับสนุนและให้ความสำคัญกับการศึกษามาโดยตลอด อีกทั้งยังให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญในการทำโครงงาน

ขอขอบคุณเพื่อนนิสิตสาขาวิทยาการคอมพิวเตอร์ที่คอยให้ความช่วยเหลือ และคอยให้กำลังใจ และรุ่นน้องสาขาวิทยาการคอมพิวเตอร์ที่คอยให้กำลังใจ

และขอขอบคุณมหาวิทยาลัยศรีนครินทรวิโรฒ ที่ผู้วิจัยได้เข้ามาศึกษาหาความรู้ สัมผัสประสบการณ์ รวมถึงได้พบกัลยาณมิตรซึ่งเป็นเสมือนอีกหนึ่งครอบครัวของผู้วิจัย

ประโยชน์ และคุณค่าอันพึงมีของโครงงานฉบับนี้ ผู้วิจัยมอบให้แก่ผู้มีพระคุณทุกท่านที่ทำให้โครงงานฉบับนี้เสร็จสมบูรณ์

จิรวัดน์ ใจรัก

ชนาภัทร ปราชญาวงศ์

## สารบัญ

|                                                                                            |   |
|--------------------------------------------------------------------------------------------|---|
| บทคัดย่อ.....                                                                              | ก |
| Abstract .....                                                                             | ข |
| กิตติกรรมประกาศ .....                                                                      | ค |
| สารบัญตาราง .....                                                                          | ช |
| สารบัญรูป.....                                                                             | ซ |
| บทที่ 1.....                                                                               | 1 |
| บทนำ.....                                                                                  | 1 |
| 1.1 ที่มาและความสำคัญของโครงการ.....                                                       | 1 |
| 1.2 วัตถุประสงค์ของโครงการ .....                                                           | 2 |
| 1.3 ขอบเขตของโครงการ.....                                                                  | 2 |
| 1.4 กรอบแนวคิดในการดำเนินโครงการ .....                                                     | 2 |
| 1.4.1 โมเดลระบบแบบ Content-based Filtering ร่วมกับ Non-negative matrix factorization ..... | 3 |
| 1.4.2 โมเดลระบบแบบ Content-based Filtering ร่วมกับ User-based CF และ Item-based CF.....    | 3 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ .....                                                        | 3 |
| บทที่ 2.....                                                                               | 4 |
| ทฤษฎี แนวคิดและงานวิจัยที่เกี่ยวข้อง.....                                                  | 4 |
| 2.1 งานวิจัยที่เกี่ยวข้อง.....                                                             | 4 |
| 2.2 องค์ความรู้เกี่ยวกับ Recommendation Systems .....                                      | 5 |
| 2.3 องค์ความรู้เกี่ยวกับ Collaborative Filtering .....                                     | 7 |
| 2.4 องค์ความรู้เกี่ยวกับ Content-based Filtering .....                                     | 9 |

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| 2.5 องค์ความรู้เกี่ยวกับ Hybrid Recommendation.....                         | 11 |
| 2.6 องค์ความรู้เกี่ยวกับ Matrix Factorization .....                         | 13 |
| 2.7 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน .....                          | 16 |
| 2.7.1 ฮาร์ดแวร์ที่ใช้ในการพัฒนา .....                                       | 16 |
| 2.7.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา .....                                       | 16 |
| 2.7.3 ภาษาโปรแกรมที่ใช้ในการพัฒนา.....                                      | 17 |
| บทที่ 3.....                                                                | 18 |
| วิธีการดำเนินโครงการ.....                                                   | 18 |
| 3.1 วิธีการดำเนินโครงการ.....                                               | 18 |
| 3.2 แผนการดำเนินโครงการ.....                                                | 19 |
| 3.3 การสำรวจข้อมูล และการคัดเลือกข้อมูล.....                                | 20 |
| 3.3.1 การเก็บรวบรวมข้อมูล และการวิเคราะห์ข้อมูล .....                       | 20 |
| 3.3.2 การคัดเลือกข้อมูลสำหรับการเทรนโมเดล.....                              | 20 |
| 3.4 การเขียนโปรแกรมพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด.....               | 21 |
| 3.4.1 การพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบ Content-based filtering .....      | 21 |
| 3.4.2 การผสานข้อมูลวิดีโอเกมสำหรับการเทรนโมเดล.....                         | 23 |
| 3.4.3 การพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบ Collaborative Filtering .....      | 23 |
| 3.5 การทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด ..... | 28 |
| บทที่ 4.....                                                                | 29 |
| ผลการดำเนินโครงการ .....                                                    | 29 |
| 4.1 แสดงค่า Root mean square error (RMSE).....                              | 29 |
| 4.2 แสดงการเปรียบเทียบค่า Root mean square error.....                       | 30 |

|                                      |    |
|--------------------------------------|----|
| บทที่ 5 .....                        | 31 |
| สรุปผล อภิปรายผล และข้อเสนอแนะ ..... | 31 |
| 5.1 สรุปผลการศึกษาค้นคว้า .....      | 31 |
| 5.2 อภิปรายผล .....                  | 32 |
| 5.3 ข้อเสนอแนะ .....                 | 32 |
| บรรณานุกรม .....                     | 33 |
| ภาคผนวก ก.....                       | 35 |

## สารบัญตาราง

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
| ตารางที่ 3-1 แผนการดำเนินโครงการตั้งแต่เดือนสิงหาคม 2561 ถึงเดือนเมษายน 2562..... | 19 |
|-----------------------------------------------------------------------------------|----|

## สารบัญรูป

|                                                                                              |    |
|----------------------------------------------------------------------------------------------|----|
| รูปที่ 3-1 ตัวอย่างข้อมูลคะแนนวิดีโอเกมจาก Amazon.com.....                                   | 20 |
| รูปที่ 3-2 ตัวอย่างข้อมูลวิดีโอเกมเฉพาะปี 2014 ที่ดึงข้อมูลด้วย API.....                     | 21 |
| รูปที่ 3-3 ตัวอย่าง Bag of words ของวิดีโอเกม 70 เกม .....                                   | 21 |
| รูปที่ 3-4 Cosine similarity matrix ของวิดีโอเกม 70 เกม.....                                 | 22 |
| รูปที่ 3-5 รูป 30 เกมแรกที่มีความคล้ายคลึงกัน.....                                           | 22 |
| รูปที่ 3-6 ตัวอย่างข้อมูลคะแนนวิดีโอเกมที่ใช้ 30 คนแรก ให้คะแนนวิดีโอเกมบ่อยที่สุด.....      | 23 |
| รูปที่ 3-7 ตัวอย่างเมทริกซ์คะแนน .....                                                       | 23 |
| รูปที่ 3-8 การแปลงค่าคะแนนในเมทริกซ์สำหรับการเทรนโมเดล.....                                  | 24 |
| รูปที่ 3-9 ตัวอย่างค่าคะแนนประมาณของเมทริกซ์ .....                                           | 24 |
| รูปที่ 3-10 Cosine similarity matrix ของผู้ใช้.....                                          | 25 |
| รูปที่ 3-11 Pearson correlation similarity matrix ของผู้ใช้ .....                            | 25 |
| รูปที่ 3-12 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ จาก Cosine similarity ของผู้ใช้.....              | 26 |
| รูปที่ 3-13 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ จาก Pearson correlation similarity ของผู้ใช้..... | 26 |
| รูปที่ 3-14 Adjust cosine similarity ของวิดีโอเกม .....                                      | 27 |
| รูปที่ 3-15 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ จาก Cosine similarity ของวิดีโอเกม.....           | 27 |
| รูปที่ 3-16 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ จาก Adjust cosine similarity ของวิดีโอเกม.....    | 27 |
| รูปที่ 3-17 ตัวอย่างการแบ่งเมทริกซ์ และสูตร Root mean square error.....                      | 28 |
| รูปที่ 4-1 รูปกราฟเส้นที่แสดงผลลัพธ์ค่า Root mean square error .....                         | 29 |
| รูปที่ 4-2 รูปกราฟแท่งที่แสดงผลลัพธ์การเปรียบเทียบค่า Root mean square error.....            | 30 |

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญของโครงงาน

ในทุกวันนี้คนเราใช้เทคโนโลยีต่างเพื่อช่วยในการหาความรู้ ทำงาน สื่อสาร และเพื่อความบันเทิง การใช้เทคโนโลยีเพื่อความบันเทิงจะมีหลายรูปแบบ เช่น การดู การฟัง การเล่น และสิ่งทีุ่ทุกวันนี้ผู้คนนิยมใช้เพื่อเป็นการคลายเครียดเป็นอย่างมาก คือ วิดีโอเกม เพราะเกมสามารถเล่นได้ ทุกวัย เล่นด้วยกันเพื่อสังสรรค์ เล่นเป็นครอบครัว เล่นกับเพื่อนๆได้ เป็นกิจกรรมอย่างหนึ่งที่สามารถใช้ผ่อนคลาย ในเวลาว่างได้เป็นอย่างดี และทุกวันนี้การเล่นเกม ไม่ใช่แค่เพื่อความสนุกสนานและการผ่อนคลายอย่างเดียว แต่สามารถทำเป็นอาชีพได้อีกด้วย

ในปัจจุบันการพัฒนาของเทคโนโลยีที่มีมาอย่างต่อเนื่อง ทำให้มีสินค้าและบริการใหม่เกิดขึ้น อยู่ตลอดเวลาเพื่อตอบสนองต่อความต้องการของคนทุกกลุ่ม อินเทอร์เน็ตจึงได้เข้ามามีบทบาทต่อชีวิตประจำวันของทุกคน ซึ่งได้ส่งผลให้เกิดการกระจายข้อมูลข่าวสารการแลกเปลี่ยน (Exchange) หรือซื้อขายสินค้า (Selling/Buying) ผ่านอินเทอร์เน็ต ที่เรียกว่าพาณิชย์อิเล็กทรอนิกส์ (E-commerce) และเกิดเครือข่าย สังคมระหว่างผู้ใช้งานอินเทอร์เน็ต ซึ่งนับวันยิ่งขยายตัวมากยิ่งขึ้น และด้วยการเติบโตดังกล่าวทำให้เกิด Data Overload หมายถึง การที่มีข้อมูลอยู่เป็นจำนวนมาก ส่งผลต่อการที่ผู้ใช้งานจะเลือกรับข้อมูลข่าวสาร หรือแม้กระทั่งการซื้อสินค้าหรือบริการ ก็ประสบปัญหาทั้งทางฝั่งลูกค้าที่ต้องการเลือกสินค้าหรือบริการที่มี ความเหมาะสมกับตนเองมากที่สุด และ ทางฝั่งผู้ขายหรือผู้ให้บริการก็เกิดความยากลำบากในการเสนอ ขายสินค้าที่มีอยู่เป็นจำนวนมากให้เหมาะสมกับ ต้องการของลูกค้า

จากปัญหาดังกล่าวทำให้เกิดแนวคิดในการคัดกรองข้อมูล และพัฒนาระบบแนะนำเกม โดยอาศัยเทคโนโลยีคัดเลือกรูปแบบของข้อมูล ซึ่งรวมไปถึงสินค้าหรือบริการที่มีความเหมาะสมกับ พฤติกรรมและความชอบ ของผู้ใช้งาน ส่งผลให้เกิดความพึงพอใจและตัดสินใจเลือกซื้อเกมหรือบริการอื่นๆได้ โดยพื้นฐานของระบบแนะนำ นั้นจำเป็นจะต้องอาศัยข้อมูลที่เคยมีอยู่ เช่น ข้อมูลส่วนบุคคล ข้อมูลพฤติกรรมและประวัติการซื้อสินค้าของผู้ใช้งาน

ดังนั้นผู้วิจัยจึงมีความสนใจที่จะพัฒนาระบบแนะนำเกมเพื่อแก้ปัญหาและปรับปรุงกระบวนการแนะนำให้มีความถูกต้องแม่นยำมากยิ่งขึ้น ซึ่งได้นำเทคนิคต่างๆมาผสมผสานโดยเฉพาะ เทคนิค Collaborative Filtering และ Content-based Filtering ที่โครงการนี้ได้ให้ความสนใจในการศึกษา รวมถึงการใช้ทั้งสอง เทคนิคร่วมกัน เพื่อแก้ปัญหาและพัฒนาระบบแนะนำเกม เพื่อช่วยแนะนำเกมประเภทต่างๆที่เหมาะสมกับผู้เล่น และเป็นตัวช่วยในการพิจารณาตัดสินใจในการเลือกซื้อเกมเหล่านี้

## 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาหลักการทำงานและทฤษฎีของระบบแนะนำ Recommendation Systems
- 1.2.2 เพื่อศึกษาเรียนรู้เทคนิค Collaborative Filtering และ Content-based Filtering ในการพัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด
- 1.2.3 เพื่อทดสอบประสิทธิภาพของอัลกอริทึมที่ใช้ในการพัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด
- 1.2.4 พัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด โดยใช้วิธีการกรองข้อมูลแบบผสม Hybrid Filtering

## 1.3 ขอบเขตของโครงการ

- 1.3.1 ใช้ชุดข้อมูล (Datasets) จาก Amazon ในการพัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด
- 1.3.2 ใช้ Collaborative Filtering และ Content-based Filtering พัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด
- 1.3.3 แสดงผลทดสอบประสิทธิภาพ ความแม่นยำของอัลกอริทึมต่างๆในแต่ละโมเดล

## 1.4 กรอบแนวคิดในการดำเนินโครงการ

พัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด โดยใช้วิธีการกรองข้อมูลแบบผสม (Hybrid Filtering) โดยการรวม Content-Based Filtering และ Collaborative Filtering (CF) เข้าด้วยกัน ซึ่งช่วยลดความเสี่ยงข้อจำกัดของ Content-Based Filtering และ Collaborative Filtering ได้แต่อาจทำให้มีความซับซ้อนและใช้ทรัพยากรในการแนะนำสูง โดยระบบแนะนำวิดีโอเกมแบบไฮบริด จะแบ่งเป็น 2 โมเดล ดังนี้

#### 1.4.1 โมเดลระบบแบบ Content-based Filtering รวมกับ Non-negative matrix factorization

- ทำนายหาเกมที่เกี่ยวข้องโดยใช้ Content-based Filtering
- ทำนายคะแนนของเกมที่ทำมาได้ โดยใช้ Non-negative matrix factorization Collaborative Filtering
- ทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด

#### 1.4.2 โมเดลระบบแบบ Content-based Filtering รวมกับ User-based CF และ Item-based CF

- ทำนายหาเกมที่เกี่ยวข้องโดยใช้ Content-based Filtering
- ทำนายคะแนนของเกมที่ทำมาได้ โดยใช้ User-based Collaborative Filtering ทั้งแบบ Cosine similarity และ Pearson correlation similarity
- ทำนายคะแนนของเกมที่ทำมาได้ โดยใช้ Item-based Collaborative Filtering ทั้งแบบ Cosine similarity และ Adjusted cosine similarity
- ทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 เข้าใจ หลักการทำงาน เทคนิคของระบบแนะนำและประสิทธิภาพของอัลกอริทึมต่างๆ
- 1.5.2 สามารถแนะนำเกมที่เหมาะสมให้กับผู้ใช้ ด้วยระบบแนะนำวิดีโอเกมแบบไฮบริดที่พัฒนาขึ้นได้จริง
- 1.5.3 สามารถเรียนรู้การเก็บข้อมูลที่ใช้ในการทำโครงการวิจัยได้ เช่น ทฤษฎี แนวคิด อุปกรณ์ และเครื่องมือที่ใช้ในการวิจัย
- 1.5.4 สามารถพัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริดที่มีความแม่นยำกว่าระบบแนะนำแบบ Collaborative Filtering ทั่วไป

## บทที่ 2

### ทฤษฎี แนวคิดและงานวิจัยที่เกี่ยวข้อง

#### 2.1 งานวิจัยที่เกี่ยวข้อง

##### 2.1.1 A Hybrid Approach with Collaborative Filtering for Recommender System

ผู้แต่ง : Badaro, Gilbert & Hajj, Hazem & El-Hajj, Wassim & Nachman, Lama

บทความนี้ใช้วิธีไฮบริดในการแก้ปัญหาการหา Rating ของไอเทมที่ไม่เคยถูกให้ คะแนนในเมทริกซ์ระหว่างผู้ใช้งานกับไอเทม ด้วยวิธี User-base และ Item-base (พร้อมค่าน้ำหนัก) โดยวิธีนี้ให้สามารถใช้แก้ปัญหาเรื่องความแม่นยำ และความว่างของข้อมูล ของระบบแนะนำโดยการรวม Correlation ของผู้ใช้และไอเทม ซึ่งวิธีนี้ให้ผลลัพธ์ที่ดีกว่า การใช้ User-base Collaborative Filtering หรือ Item-base Collaborative Filtering เพียง อย่างเดียว

##### 2.1.2 A Personalized Recommender System using User-based and Item based Collaborative Filtering Method

ผู้แต่ง : พันตรีหญิง รัฐวรรณ พันธุนิล, ผู้ช่วยศาสตราจารย์ ดร. นิเวศ จิระวิจิตชัย

บทความนี้พัฒนาระบบแนะนำส่วนบุคคล โดยใช้ User-based and Item-based เพื่อแก้ปัญหาขนาดของข้อมูล (Scalability Problem) และการให้ Rating ต่อชิ้นข้อมูล (Sparsity Problem) โดยจะได้ค่าพื้นที่ใต้เส้นกราฟ ค่าความแม่นยำ ค่า Normalized Discounted Cumulative Gain และค่าเฉลี่ยของความแม่นยำ ที่สูงขึ้น เป็นผลลัพธ์ที่ดีกว่า การใช้ User-based Collaborative Filtering หรือ Item-based Collaborative Filtering เพียงอย่างเดียว

##### 2.1.3 Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey

ผู้แต่ง : Bokde, Dheeraj & Girase, Sheetal & Mukhopadhyay, Debajyoti

Collaborative Filtering (CF) เป็นวิธีการที่มีความนิยมมากที่สุดในการสร้างระบบ แนะนำ โดยใช้พฤติกรรมของผู้ใช้งานในอดีต จะถูกวิเคราะห์เพื่อสร้างความเชื่อมโยงระหว่าง ผู้ใช้งานกับไอเท็ม การเติบโตอย่างรวดเร็วของอินเทอร์เน็ต ทำให้เกิดข้อมูลขึ้นเป็นจำนวน มาก และทำให้ CF พบกับปัญหา

เมื่อ Rating Matrix มีความเบาบางทางข้อมูล และการเติบโตตามธรรมชาติของข้อมูล ปัญหา เหล่านี้ แก้ไขได้ด้วย Matrix Factorization (MF) ใน บทความนี้จะพูดถึงโมเดล MF ในรูปแบบ ต่างๆ เช่น Singular Value Decomposition (SVD) Principal Component Analysis (PCA) และ Probabilistic Matrix Factorization (PMF)

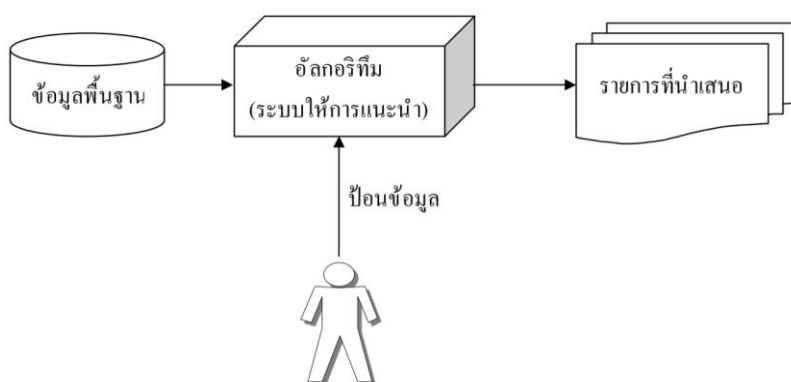
## 2.2 องค์ความรู้เกี่ยวกับ Recommendation Systems

ระบบแนะนำ (Recommendation Systems) คือ ระบบที่แนะนำข้อมูล ผลิตภัณฑ์ หรือ ผู้คน (โดยหลังจากนี้จะเรียกว่า “วัตถุ”) ให้กับผู้ใช้ระบบโดยอ้างอิงจากสมมติฐานการเรียนรู้ข้อมูล ความชอบหรือความต้องการ ณ ขณะนั้นของผู้ใช้ โดยระบบแนะนำกลายเป็นหัวข้องานวิจัยที่สำคัญ ตั้งแต่มีการปรากฏตัวของงานวิจัยเรื่อง Collaborative Filtering ในช่วงกลางทศวรรษที่ 1990 โดย ความสำเร็จในช่วงแรกๆของการทำระบบแนะนำเกิด มาจากการเจริญเติบโตของธุรกิจทางด้าน E-commerce โดยตัวอย่างของ Applications ที่ใช้ระบบแนะนำใน การเลือกซื้อหนังสือ ซีดีเพลงหรือ ผลิตภัณฑ์อื่นๆ ได้แก่ เว็บ Amazon.com และเว็บ CDNow.com เป็นต้น ทั้งนี้ เพื่อให้ลูกค้าที่ภักดีต่อ องค์กรและเพิ่มโอกาสทางธุรกิจ อย่างไรก็ตามไม่ว่าระบบแนะนำจะก้าวหน้าไปเช่นไร ระบบแนะนำก็ ยังคงต้องการการพัฒนาความสามารถมากกว่านี้ เพื่อที่จะทำให้ระบบสามารถแนะนำสิ่งต่างๆ ให้แก่ ผู้ใช้ได้อย่างมีประสิทธิภาพมากยิ่งขึ้นและใช้งานได้ง่ายขึ้นแม้แต่กับกิจกรรมต่างๆไปในชีวิตประจำวัน ตัวอย่างเช่น การแนะนำแพ็คเกจช่วงเวลาการพักร้อน การซื้อสินค้าในร้านที่มีระบบ Smart Shopping Cart

ระบบแนะนำเป็นโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่ช่วยคัดกรองวัตถุที่ตรงกับความต้องการของผู้ใช้จาก วัตถุที่มีอยู่เป็นจำนวนมาก วัตถุในความหมายของระบบแนะนำหมายถึงสิ่งที่ต้องการจะแนะนำ เช่น หนังสือ ภาพยนตร์ บริการ หรือข้อมูลข่าวสาร เป็นต้น โดยทั่วไประบบแนะนำแบบดั้งเดิม (Classical Recommendation System) จะทำงานได้ต้องอาศัยข้อมูลที่สำคัญสองส่วน คือ ข้อมูลอธิบายความชอบโดยรวมของผู้ใช้ที่มี ต่อวัตถุ และข้อมูล อธิบายลักษณะของวัตถุที่ผู้ใช้ชอบจะถูก อธิบายด้วยคะแนนที่เป็นตัวเลข (Rating Score) ที่มี ค่าอยู่ในช่วงที่ แนนอน เช่น 1 หมายถึง พึงพอใจน้อยที่สุด และ 5 หมายถึง พึงพอใจมากที่สุด เป็นต้น

ซึ่งระบบให้คำแนะนำจะประกอบไปด้วย 4 ส่วนหลักๆ

1. ส่วนข้อมูลพื้นฐานที่จำเป็นต้องใช้ในการประมวลผล เช่น โปรไฟล์ของผู้ใช้แต่ละคน
2. ส่วนการป้อนข้อมูลเป็นข้อมูลที่ได้จากการป้อนข้อมูลเข้ามาของผู้ใช้ เช่น การให้คะแนน Rating ซึ่งมีอยู่ 2 แบบ ดังนี้
  - 2.1. แบบชัดเจน (Explicit) จะแสดงอยู่ในรูปของจำนวนตัวเลขตามระดับความนิยมตั้งแต่ 1 ถึง 5 และ 1 ถึง 10 หรือระดับอื่นๆขึ้นอยู่กับการใช้งาน
  - 2.2. แบบไม่ชัดเจน (Implicit) ได้มาจากพฤติกรรมการใช้งานของผู้ใช้ต่างๆ เช่น ประวัติการซื้อสินค้า หรือประวัติการเข้ามาใช้งานของผู้ใช้ในอดีตที่ผ่านมา
3. ส่วนของอัลกอริทึมเป็นส่วนสำคัญที่สุดที่ใช้ประมวลผลข้อมูลเพื่อให้การแนะนำข้อมูลออกมาให้ตรงกับความต้องการของลูกค้ามากที่สุด ซึ่งโดยทั่วไปก็จะมีอยู่ 3 วิธีหลักคือ Content-based Filtering Collaborative Filtering และ Hybrid Filtering
4. ส่วนของการนำเสนอคำแนะนำหรือนำเสนอสินค้าหรือบริการที่ตรงกับความต้องการของผู้ใช้มากที่สุด



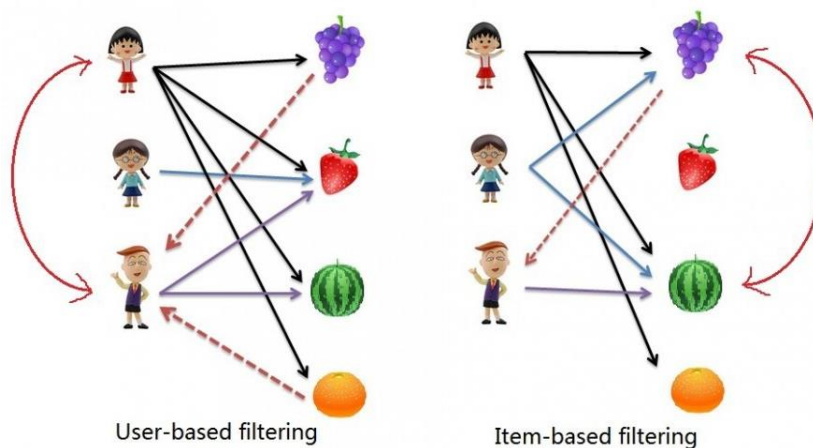
รูปที่ 2-1 สถาปัตยกรรมพื้นฐานของระบบให้คำแนะนำ

ที่มา: [ออนไลน์] [http://www.rpu.ac.th/Library\\_web/doc/RC\\_RR/2554\\_ComBus\\_Walainush.pdf](http://www.rpu.ac.th/Library_web/doc/RC_RR/2554_ComBus_Walainush.pdf)

## 2.3 องค์ความรู้เกี่ยวกับ Collaborative Filtering

Collaborative Filtering เป็นวิธีการที่เกี่ยวกับการเก็บรวบรวม และวิเคราะห์ข้อมูลพฤติกรรมของผู้ใช้งาน แล้วทำการทำนายความชอบของผู้ใช้งานนั้นๆ เมื่อเทียบกับผู้ใช้งานคนอื่นๆ โดยข้อดีของ Collaborative Filtering คือเป็นวิธีการที่ไม่จำเป็นต้องใช้ข้อมูลที่เครื่องคอมพิวเตอร์ สามารถวิเคราะห์ได้ ทำให้สามารถแนะนำวัตถุที่มีความซับซ้อน เช่น หนังภาพยนตร์ได้อย่างแม่นยำ โดยไม่จำเป็นต้องเข้าใจถึงตัววัตถุเอง

Collaborative Filtering สามารถแบ่งได้เป็น 2 กลุ่ม คือ Memory-Based หรือ Heuristic-based และ Model-based Collaborative Filtering ซึ่งในที่นี้จะกล่าวแค่ Memory-Based ที่จะทำการแนะนำวัตถุโดยใช้ข้อมูลในระบบเป็นจำนวนมากที่มีความคล้ายคลึงหรือใกล้เคียงกันกับ ผู้ใช้เช่น ความชอบ ความเคลื่อนไหว กิจกรรมของผู้ใช้คนอื่นๆในระบบ ซึ่งจะเรียกเฉพาะเจาะจงได้ว่า ระบบแนะนำแบบคัดกรองผู้ใช้ร่วม (User-based Filtering) แต่ถ้าใช้ข้อมูลลักษณะหรือคุณสมบัติของ วัตถุอื่นๆในระบบก็จะเรียกอีกแบบหนึ่งว่า ระบบแนะนำแบบคัดกรองสิ่งของร่วม (Item-based Filtering) กล่าวคือใช้ข้อมูลของสิ่งอื่นๆในระบบมาช่วยในการคำนวณด้วย

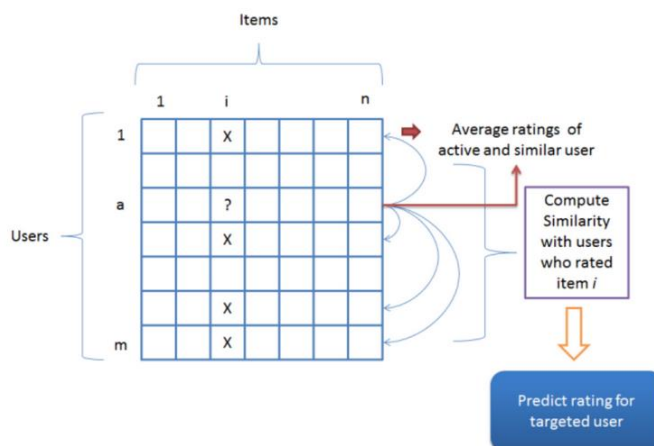


รูปที่ 2-2 Collaborative Filtering

ที่มา: [ออนไลน์] <https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f>

โดยปกติระบบแนะนำประเภทนี้จะมีขั้นตอนในการประมวลผล 3 ขั้นตอนใหญ่ๆ ดังนี้

1. สร้างประวัติผู้ใช้หรือวัตถุตามแต่ข้อมูลที่จะใช้เป็นพื้นฐานของระบบ โดยจะมีเซตของผู้ใช้งาน  $m$  คน  $U = \{u_1, u_2, \dots, u_m\}$  และ เซตของวัตถุจำนวน  $n$  วัตถุ  $I = \{i_1, i_2, \dots, i_n\}$  ผู้ใช้งาน  $u_i$  แต่ละคน ได้เคยให้คะแนนวัตถุต่างๆไว้รายการหนึ่ง ให้ เป็น  $I_{ui}$  โดยจุดมุ่งหมายของเทคนิคนี้ คือ การทำนายคะแนนของวัตถุที่ไม่เคยถูกให้คะแนน



รูปที่ 2-3 Collaborative Filtering Process using User-based Approach

ที่มา: A Hybrid Approach with Collaborative Filtering for Recommender Systems

2. คัดเลือกผู้ใช้ที่เคยเลือกวัตถุนั้น (Co-Rated Item) หรือวัตถุที่เคยถูกเลือกจากผู้ใช้งานดังกล่าว (Couser Rate) ที่มีความใกล้เคียงหรือคล้ายคลึงกันขึ้นมาตามจำนวนที่กำหนด ไว้โดยการเปรียบเทียบประวัติผู้ใช้หรือวัตถุตามแต่ข้อมูลที่ใช้เป็นพื้นฐานของระบบ ซึ่งอาจใช้ค่าความเหมือนโคไซน์ (Cosine Similarity) สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson Correlation Coefficient) หรือ ระยะทางแบบยุคลิด (Euclidean Distance)

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

รูปที่ 2-4 สมการ Cosine Similarity

ที่มา: [ออนไลน์] [https://en.wikipedia.org/wiki/Cosine\\_similarity](https://en.wikipedia.org/wiki/Cosine_similarity)

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}}$$

รูปที่ 2-5 สมการ Pearson Correlation Coefficient

ที่มา: A Hybrid Approach with Collaborative Filtering for Recommender Systems

3. คำนวณว่าผู้ใช้และวัตถุนั้นมีความเหมาะสมกันมากน้อยเพียงใด โดยอาศัยข้อมูลที่หาได้ใน ขั้นตอนที่ 2

## 2.4 องค์ความรู้เกี่ยวกับ Content-based Filtering

Content-based Filtering เป็นวิธีการกรองข้อมูลที่จะนำมาใช้ในระบบแนะนำข้อมูล โดยนิยมใช้ในระบบสารสนเทศที่ให้ความสนใจกับเนื้อหาตามคุณสมบัติของข้อมูลเป็นหลัก วิธีนี้จะตรวจสอบว่าข้อมูลหรือเนื้อหาส่วนใดที่ตรงกันหรือมีความคล้ายคลึงกันกับข้อมูลส่วนบุคคลของผู้ใช้ (User profile) ถ้าตรงกันก็จะนำมาแสดงผลรายการแนะนำให้แก่ผู้ใช้ แต่ถ้าไม่ตรงกันก็จะไม่มีการแสดงผลการแนะนำ ถึงแม้ว่าข้อมูลหรือเนื้อหาอาจจะมีลักษณะใกล้เคียงกับที่ผู้ใช้ต้องการก็ตาม ดังนั้น วิธีการนี้จะเป็นการคำนวณหาความคล้ายคลึงระหว่างข้อมูลหรือเนื้อหา กับข้อมูลส่วนบุคคลของผู้ใช้ โดยการจับคู่ข้อมูลหรือเนื้อหา กับข้อมูลส่วนบุคคลของผู้ใช้ เพื่อใช้ในระบบสารสนเทศตามความสนใจ ของผู้ใช้ เทคนิคที่ใช้ในการกรองสารสนเทศโดยดูที่เนื้อหา มีการนำหลายเทคนิคมาใช้ เช่น การจัดกลุ่มข้อมูล (Clustering) การแบ่งกลุ่มข้อมูลแบบเบย์เซียน (Bayesian Classifiers) การจำแนกข้อมูลโดยใช้ต้นไม้ตัดสินใจ (Decision Trees) โครงข่ายประสาทเทียมที่มีการเลียนแบบ การทำงานของสมองของมนุษย์ (Artificial Neural Networks) เป็นต้น

Content-based Filtering เป็นวิธีการที่มาจากคำอธิบายของวัตถุและโปรไฟล์ความชอบของ ผู้ใช้โดยในระบบจะมี Keywords ที่ใช้ในการอธิบายวัตถุและโปรไฟล์ของผู้ใช้ที่ถูกสร้างขึ้นมาเพื่อระบุ ชนิดของวัตถุที่ผู้ใช้ชอบหรือก็คือ อัลกอริทึมนี้จะพยายามที่จะแนะนำวัตถุที่ผู้ใช้ชอบมาก่อน (หรือกำลังสนใจอยู่ในปัจจุบัน) โดยจะมีวัตถุที่เป็น Candidate ต่างๆที่จะถูกนำมาเปรียบเทียบกับวัตถุ ที่ผู้ใช้เคยให้คะแนนมาก่อน แล้วจึงแนะนำวัตถุที่เหมาะสมที่สุดให้กับผู้ใช้ ซึ่งวิธีนี้มีรากฐานมาจากการ รวบรวมข้อมูลและการกรองข้อมูล

วิธีการของระบบนี้จะแตกต่างจากระบบแนะนำคัดกรองร่วม คือ แทนที่จะแนะนำวัตถุโดยใช้ข้อมูลของผู้ใช้หรือวัตถุที่มีความใกล้เคียงกัน ระบบจะทำการเลือกวัตถุโดยอิงจากความพึงพอใจของผู้ใช้ว่าชอบวัตถุในลักษณะใด คุณสมบัติอย่างไร ซึ่งยังแสดงให้เห็นอีกด้วยว่าผู้ใช้แต่ละคนมีเกณฑ์การเลือกวัตถุอย่างไรสนใจหรือยึดคุณสมบัติใดของวัตถุเป็นหลัก โดยทั่วไปแล้วระบบแนะนำประเภทนี้จะ มีขั้นตอนการทำงานดังนี้

1. สร้างประวัติผู้ใช้ระบบ โดยอ้างอิงจากลักษณะหรือคุณสมบัติของวัตถุซึ่งคำนวณจากกิจกรรมความเคลื่อนไหวที่ผ่านๆมาของผู้ใช้
2. คำนวณความสัมพันธ์ระหว่างประวัติผู้ใช้และวัตถุ
3. นำผลลัพธ์ที่ได้จากข้อ 2 มาปรับค่าน้ำหนักแต่ละคุณสมบัติของวัตถุ
4. เลือกวัตถุที่มีความสัมพันธ์กับผู้ใช้เป้าหมายมากที่สุด โดยดูจากความสำคัญที่ผู้ใช้เป้าหมายให้ไว้ในแต่ละคุณสมบัติ และค่าน้ำหนักของแต่ละคุณสมบัติของวัตถุที่มีอยู่ในระบบ



รูปที่ 2-6 Content-based Filtering

ที่มา: [ออนไลน์] <https://valiancesolutions.com/recommender-systems-101/>

## 2.5 องค์ความรู้เกี่ยวกับ Hybrid Recommendation

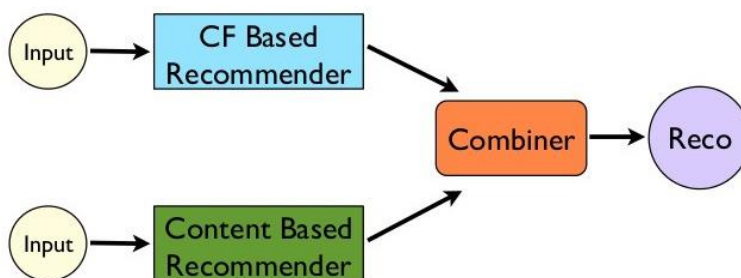
ระบบแนะนำมากมายใช้วิธีการกรองข้อมูลแบบผสม (Hybrid Filtering) โดยการรวม Content-based และ Collaborative Filtering เข้าด้วยกัน ซึ่งช่วยหลีกเลี่ยงข้อจำกัดของ Content-based และ Collaborative Filtering ได้แต่อาจทำให้มีความซับซ้อนและใช้ทรัพยากรในการแนะนำสูง โดยหลายผลงานแสดงให้เห็นว่ามีความแม่นยำมากกว่า Content-based Filtering และ Collaborative Filtering

Hybrid Filtering จะรวมข้อมูลคุณลักษณะของวัตถุกับผู้ใช้และข้อมูล Rating เพื่อให้ได้ผลประโยชน์จากข้อมูลทั้งสองแบบ ระบบแนะนำแบบไฮบริดทำการรวมอัลกอริทึม A เข้ากับ B โดยพยายามที่จะใช้ข้อดีของ A ในการแก้ปัญหาของ B เช่น Collaborative Filtering จะมีปัญหาเรื่องการแนะนำวัตถุใหม่ เพราะวัตถุใหม่ไม่มี ข้อมูล Rating แต่เรื่องนี้ไม่ใช่ปัญหาสำหรับ Content-based Filtering เพราะว่าการทำนายวัตถุใหม่มาจาก ตัว วัตถุเอง (คุณลักษณะ) ซึ่งมีให้อยู่แล้วเมื่อวัตถุใหม่เข้าสู่ระบบ ซึ่งการสร้างระบบแนะนำแบบไฮบริด โดย การรวม Collaborative Filtering เข้ากับ Content-based Filtering เราสามารถก้าวข้ามข้อจำกัดต่างๆของ แต่ละอัลกอริทึมได้ เช่น ปัญหา Cold-Start และ Popularity Bias

วิธีการที่จะนำ Content-Based Filtering และ Collaborative Filtering มารวมกันเป็นระบบแนะนำแบบ Hybrid Filtering สามารถแบ่งได้ ดังนี้

1. Combining Separate Recommenders เป็นทางเลือกหนึ่งในการสร้างระบบแนะนำแบบ Hybrid ซึ่งก็คือ การใช้ Content-based และ Collaborative Filtering แยกกันคนละส่วน จากนั้นจะมีผลลัพธ์ออกมาสองแบบ ซึ่งแบบแรกสามารถรวมผลลัพธ์ที่ได้จากระบบแนะนำแต่ละแบบแล้วให้การแนะนำเดียวกัน โดยจะใช้วิธีการรวมแบบ Linear หรือการ Vote หรืออีกทางหนึ่งคือใช้ผลลัพธ์จากระบบแนะนำใดระบบแนะนำหนึ่งเท่านั้น ซึ่งจะเลือกจากผลลัพธ์ที่ดีที่สุดโดยอ้างอิงจากเมตริกซ์คุณภาพในการแนะนำ
2. Adding Content-based Characteristics to Collaborative Models เป็นการนำคุณสมบัติเฉพาะบางอย่างของวิธี Content-based มาใส่ไว้ในวิธี Collaborative Filtering ซึ่งมีหลายระบบที่ทำระบบแนะนำแบบ Hybrid ในรูปแบบนี้เช่น Fab หรือ Collaborative via Content ซึ่งใช้ Collaborative Filtering แบบดั้งเดิมข้อมูลความชอบของผู้ใช้จะเป็นแบบ Content-Based ซึ่งเป็นการแก้ปัญหาข้อมูลของผู้ใช้ที่มีการ Rating วัตถุน้อยเกินไปซึ่งทำให้เกิดข้อจำกัดแบบ Sparsity

3. Adding Collaborative Characteristics to Content-based Models เป็นการนำคุณสมบัติเฉพาะบางอย่างของวิธี Collaborative Filtering มาใส่ไว้ในวิธี Content-based วิธีที่นิยมใช้กัน คือ การใช้เทคนิคการลดมิติของกลุ่มข้อมูลผู้ใช้ที่อ้างอิงตาม Content ซึ่งตัวอย่างของวิธีการรูปแบบนี้ ได้แก่ Latent Semantic Indexing ซึ่งเป็น Algorithm ที่ใช้ในการสร้างมุมมองแบบ Collaborative ให้กับ User Profile ที่แสดงในรูปเทอมของเวกเตอร์ ซึ่งจากผลลัพธ์จากวิธี Hybrid ในรูปแบบนี้ทำให้ระบบแนะนำ มีประสิทธิภาพเพิ่มขึ้น
4. Developing a Single Unifying Recommendation Model เป็นการสร้างโมเดลทางคณิตศาสตร์จากลักษณะเฉพาะของวิธี Content-based และ Collaborative Filtering เช่น Bayesian Mixed-Effects Regression Models ซึ่งใช้วิธีการของ Markov Chain Monte Carlo ในการประมาณค่าพารามิเตอร์แล้วคาดคะเนหาค่า Rating



รูปที่ 2-7 Hybrid recommendations

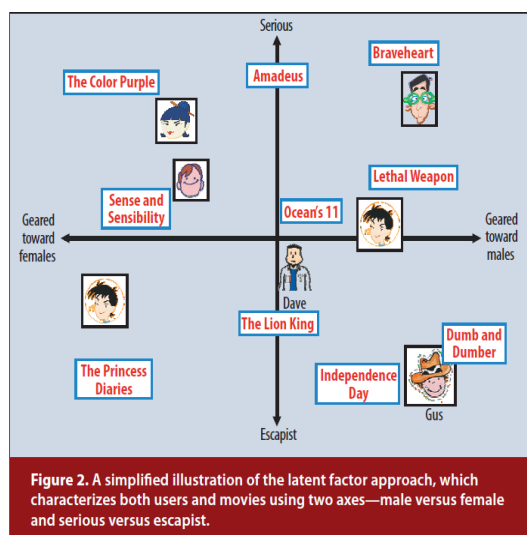
ที่มา: [ออนไลน์] <https://www.marutitech.com/recommendation-engine-benefits/>

## 2.6 องค์ความรู้เกี่ยวกับ Matrix Factorization

เมื่อผู้ใช้ให้คะแนนกับภาพยนตร์ที่พวกเขาดู (พวกเขาสามารถให้คะแนนจากหนึ่งถึงห้า) คะแนนเหล่านี้สามารถแสดงในรูปแบบของเมทริกซ์ ที่แต่ละแถวแสดงถึงผู้ใช้แต่ละคน ในขณะที่แต่ละคอลัมน์แสดงภาพยนตร์ที่แตกต่างกัน เห็นได้ชัดว่าเมทริกซ์จะกระจายเพราะทุกคนไม่ได้ดูหนังทุกเรื่อง (เราทุกคนมีรสนิยมแตกต่างกันเมื่อพูดถึงภาพยนตร์)

จุดแข็งประการหนึ่งของ Matrix factorization คือมันสามารถรวม Implicit Feedback ได้ ซึ่งเป็นข้อมูลที่ไม่ได้รับโดยตรง แต่สามารถได้มาจากการวิเคราะห์พฤติกรรมของผู้ใช้ การใช้จุดแข็งนี้เราสามารถประเมินได้ว่าผู้ใช้จะชอบหนังที่ไม่เคยเห็นมาก่อนหรือไม่ และหากการประมาณคะแนนสูงเราสามารถแนะนำภาพยนตร์นั้นให้กับผู้ใช้

Matrix Factorization สร้าง latent feature เพื่อเปรียบเทียบผู้ใช้รายหนึ่งกับผู้ใช้รายอื่น (feature base) อธิบายง่ายๆด้วยภาพประกอบที่เรียบง่ายของ Latent Factor ซึ่งกำหนดลักษณะทั้งผู้ใช้และภาพยนตร์โดยใช้สองแกน - Male เทียบกับ Female และ Serious เทียบกับ Escapist ดังรูปที่ 2-8

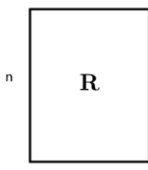


รูปที่ 2-8 Latent Factor

ที่มา: [ออนไลน์] <https://towardsdatascience.com/paper-summary-matrix-factorization-techniques-for-recommender-systems-82d1a7ace74>

สมมติว่าเรามีเซตของผู้ใช้งาน  $U$  (users) โดยมีขนาด  $n$  คน และมีเซตของเกม  $D$  (games) ซึ่งมีขนาด  $m$  เกม นอกจากนั้น สมมติว่าเรามีประวัติการให้คะแนนของแต่ละผู้ใช้งาน ว่าเขาให้คะแนนเกมแต่ละเกมไปเท่าไร

|    | D1 | D2 | D3 | D4 |
|----|----|----|----|----|
| U1 | 5  | 3  | -  | 1  |
| U2 | 4  | -  | -  | 1  |
| U3 | 1  | 1  | -  | 5  |
| U4 | 1  | -  | -  | 4  |
| U5 | -  | 1  | 5  | 4  |



$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^k p_{ik} q_{kj}$$

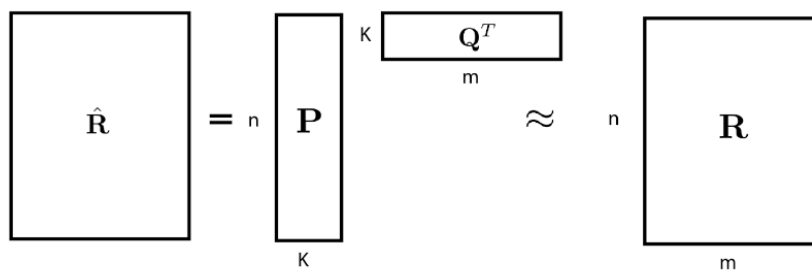
รูปที่ 2-9 เมทริกซ์ตัวอย่างและสมการคำนวณคะแนน

เราสามารถเขียนคะแนน การให้คะแนนของผู้ใช้งาน ในรูปของเมทริกซ์ได้ โดยเราจะใช้ชื่อว่า  $R$  ซึ่งมีขนาดเท่ากับขนาดของผู้ใช้งานทั้งหมดและจำนวนเกมทั้งหมด หรือว่า  $n \times m$  ในแต่ละตำแหน่งของเมทริกซ์ เราจะใช้สัญลักษณ์ว่า  $r_{ij}$  แทนจำนวนคะแนนที่ผู้ใช้งาน  $i$  ให้คะแนนเกม  $j$  ส่วนเกมที่ผู้ใช้งานไม่เคยให้คะแนน เราก็ไม่ต้องใส่ข้อมูลใดๆไปบนเมทริกซ์นั้น หรือทิ้งค่าว่างๆไว้เป็น 0 ก็ได้

สิ่งที่เราต้องการจะหาคือเมทริกซ์อันดับต่ำ (Low-Rank Matrix) ที่จะบรรยายทั้งผู้ใช้และเกมที่ผู้ใช้ ชื่นชอบ และจะเห็นว่าเมทริกซ์  $P$  มีจำนวนแถวเท่ากับจำนวนผู้ใช้งานเลย ถ้าเรามีเมทริกซ์นี้แล้วในแต่ละแถว เราสามารถใช้บอกได้ว่า ผู้ใช้งานคนใดชอบเกมคล้ายๆกัน นอกจากนั้น ถ้าเราเอาแถวหนึ่งจากเมทริกซ์  $P$  มาคูณกับเมทริกซ์  $Q$  เราจะสามารถบอกได้ว่า ผู้ใช้งานคนนั้นควรจะชอบเกมอะไรเพิ่มบ้างนอกเหนือจากที่เค้าชอบอยู่

$$R \approx P \times Q^T = \hat{R}$$

โดยที่  $P$  มีขนาด  $n \times k$  และ  $Q^T$  มีขนาด  $k \times m$ . สิ่งที่เราต้องการคือเราสามารถประมาณเมทริกซ์  $R$  ด้วย  $\hat{R} = P \times Q^T$



Non-Negative Matrix Factorization สำหรับ Collaborative Filtering

รูปที่ 2-10 เมทริกซ์  $R$  และสมการคำนวณเมทริกซ์  $R$

สิ่งที่เราต้องการจะหาคือทำให้ความผิดพลาดของการประมาณ ( $e_{ij}$ ) ในทุกๆตำแหน่งของ  $\hat{R}$  เมื่อเทียบกับ  $R$  มีค่าน้อยที่สุด นั่นคือในแต่ละตำแหน่งจะมีค่าความผิดพลาด คำนวณได้จากสมการ Squared Error ซึ่งก็คือ ค่าความผิดพลาดระหว่างคะแนนโดยประมาณและคะแนนจริง

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^K p_{ik}q_{kj})^2$$

รูปที่ 2-11 สมการ Squared Error

เพื่อลดค่าความผิดพลาด เราต้องรู้ทิศทาง เพื่อที่จะปรับค่าพารามิเตอร์ หรือเราต้องใช้ Gradient Descent เพื่อที่จะสามารถหาพารามิเตอร์ที่ดีที่สุดได้ด้วยการดิฟค่าความผิดพลาดนี้ เทียบกับพารามิเตอร์ที่เราต้องการจะหา

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij}q_{kj}$$

$$\frac{\partial}{\partial q_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij}p_{ik}$$

รูปที่ 2-12 สมการ Squared Error ที่ใช้ Gradient Descent

เราใช้ Regularization เพื่อหลีกเลี่ยงการเกิด Overfitting ค่าของการประมาณเมทริกซ์  $P$  และ  $Q$  อาจจะมีมากเกินไป หรืออาจจะติดลบไปมากๆ ซึ่งเราไม่อยากจะให้เกิดขึ้น โดยส่วนมากแล้ว เราจะใส่สิ่งที่เรียกว่า Regularization Term เพิ่มพารามิเตอร์  $\beta$  เข้าไปด้วยเพื่อทำให้ค่าที่เราประมาณมีขนาดไม่ใหญ่จนเกินไป และ ปรับสมการ Squared Error ใหม่ ได้ดังต่อไปนี้

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^K p_{ik}q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^K (\|P\|^2 + \|Q\|^2)$$

รูปที่ 2-13 สมการ Squared Error ที่ใช้ Regularization

หลังจากนั้นเราค่อยๆทำวนไปแต่ทุกๆแถวและหลักของเมทริกซ์ และอัปเดต  $p_{ij}, q_{ij}$  ใหม่ ได้ดังต่อไปนี้

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2e_{ij}q_{kj} - \beta p_{ik})$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha(2e_{ij}p_{ik} - \beta q_{kj})$$

รูปที่ 2-14 สมการอัปเดตค่า  $p_{ij}, q_{ij}$

โดย  $\alpha$  สามารถเรียกว่า Learning Rate นั่นเอง และ  $\beta$  เป็นค่าที่บอกว่าเราควรจะให้น้ำหนักของ Regularization กับพารามิเตอร์มากขนาดไหน

## 2.7 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

ในการทำโครงการได้ดำเนินการโดยใช้เครื่องมือที่เป็นทั้งฮาร์ดแวร์และซอฟต์แวร์ดังต่อไปนี้

### 2.7.1 ฮาร์ดแวร์ที่ใช้ในการพัฒนา

2.7.1.1. คอมพิวเตอร์ Intel(R) Core (TM) i5-3470HQ CPU 3.20 GHz

### 2.7.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา

2.7.2.1 Jupyter notebook



รูปที่ 2-15 รูปสัญลักษณ์โปรแกรม Jupyter notebook

ที่มา : [ออนไลน์] [https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Jupyter\\_logo.svg/250px-](https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Jupyter_logo.svg/250px-Jupyter_logo.svg.png)

[Jupyter\\_logo.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Jupyter_logo.svg/250px-Jupyter_logo.svg.png)

**Jupyter notebook** คือ เป็น Interactive Computing Environment ที่ให้ผู้ใช้สามารถป้อนคำสั่ง แล้วให้ประมวลผลคำสั่งนั้นๆ พร้อมแสดงผลลัพธ์ต่อท้ายชุดคำสั่งตัว Notebook สามารถเก็บ Code และตัว Markdown ที่ช่วยให้แสดงข้อมูลในรูปแบบต่างๆได้ เช่น ต้องการแสดงเป็น Heading แบบต่างๆ

## 2.7.3 ภาษาโปรแกรมที่ใช้ในการพัฒนา

### 2.7.3.1 Python



รูปที่ 2-16 สัญลักษณ์ภาษา Python Programming Language

ที่มา : [ออนไลน์] <https://pluralsight.imgix.net/paths/python-7be70baaac.png>

**Python** คือ ภาษาโปรแกรมขั้นสูงชนิดหนึ่ง ถูกออกแบบให้ code สามารถ อ่านได้ง่าย โดยใช้ “ช่องไฟ” หรือ whitespace เป็นตัวแบ่งว่า code แต่ละบรรทัดอยู่ภายใต้ block ไหน แทนที่จะใช้ปีกกา ( {, } ) แบบภาษาอื่น โครงสร้างภาษามีพื้นฐานการใช้จำนวนบรรทัดให้น้อย ถ้าเทียบกับ C++ และ Java ลักษณะ feature มีด้วยกันหลายแบบ เช่น สามารถบริหารจัดการหน่วยความจำได้เอง, รองรับการทำงานหลาย paradigm, object-oriented, functiona

## บทที่ 3

### วิธีการดำเนินโครงการ

การดำเนินโครงการในครั้งนี้ ผู้วิจัยได้ดำเนินการศึกษาค้นคว้าตามขั้นตอนดังนี้

- 3.1 วิธีการดำเนินโครงการ
- 3.2 แผนการดำเนินโครงการ
- 3.3 การสำรวจข้อมูล และการคัดเลือกข้อมูล
- 3.4 การเขียนโปรแกรมพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด
- 3.5 การทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด

#### 3.1 วิธีการดำเนินโครงการ

- 3.1.1 ศึกษาค้นคว้าแนวคิด และทฤษฎีที่เกี่ยวข้องกับระบบแนะนำ
- 3.1.2 รวบรวมข้อมูล และวิเคราะห์ข้อมูล เพื่อค้นหารูปแบบในข้อมูล และกำหนดขอบเขตของโครงการวิจัย
- 3.1.3 วิเคราะห์ และออกแบบการทำงานของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด
- 3.1.4 เขียนโปรแกรมพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด และทำการเทรนโมเดลระบบแนะนำที่พัฒนาขึ้น
- 3.1.5 ทำการทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด
- 3.1.6 ปรับปรุง และแก้ไขข้อผิดพลาดหลังการทดสอบประสิทธิภาพ
- 3.1.7 ประเมินผล สรุปผลการทดลองและจัดทำรูปเล่ม



### 3.3 การสำรวจข้อมูล และการคัดเลือกข้อมูล

#### 3.3.1 การเก็บรวบรวมข้อมูล และการวิเคราะห์ข้อมูล

ทำการเก็บรวบรวมข้อมูล โดยใช้ชุดข้อมูลคะแนนวิดีโอเกม จากเว็บไซต์ Amazon.com ซึ่งในชุดข้อมูลมีข้อมูลทั้งหมด 1,324,753 แถว ประกอบไปด้วย 826,767 ผู้ใช้ และ 50,210 เกม ดังรูปที่ 3-1

|   | User           | Game       | Rating | Datetime   |
|---|----------------|------------|--------|------------|
| 0 | AB9S9279OZ3QO  | 0078764343 | 5.0    | 2013-07-07 |
| 1 | A24SSUT5CSW8BH | 0078764343 | 5.0    | 2013-08-24 |
| 2 | AK3V0HEBJMQ7J  | 0078764343 | 4.0    | 2013-07-04 |
| 3 | A10BECPH7W8HM7 | 043933702X | 5.0    | 2014-07-10 |
| 4 | A2PRV9OULX1TWP | 043933702X | 5.0    | 2013-12-04 |
| 5 | AE7GUHCDQ4UI   | 043933702X | 1.0    | 2013-04-17 |
| 6 | A48ABFDDRMKI8  | 043933702X | 5.0    | 2013-07-19 |
| 7 | A26B0P6K95SIKW | 0439339960 | 3.0    | 2010-11-01 |
| 8 | AZ3UWOC8QSO6C  | 0439339987 | 5.0    | 2013-04-25 |
| 9 | A182S3ANC0W7DL | 0439342260 | 4.0    | 2012-12-19 |

รูปที่ 3-1 ตัวอย่างข้อมูลคะแนนวิดีโอเกมจาก Amazon.com

#### 3.3.2 การคัดเลือกข้อมูลสำหรับการเทรนโมเดล

ทำการคัดเลือกข้อมูลจากปี 2014 ซึ่งเป็นปีล่าสุดในชุดข้อมูลคะแนนวิดีโอเกม มาใช้ในการเทรนโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด โดยในปี 2014 มีข้อมูลทั้งหมด 222,009 แถว ประกอบไปด้วย 168,291 ผู้ใช้ และ 23,177 เกม

ทำการดึงข้อมูลของวิดีโอเกมเฉพาะปี 2014 โดยใช้ Application Programming Interface (API) ซึ่งข้อมูลของวิดีโอเกม ประกอบไปด้วย ชื่อวิดีโอเกม แพลตฟอร์มวิดีโอเกม ประเภทวิดีโอเกม และผู้จัดจำหน่ายวิดีโอเกม

ข้อมูลวิดีโอเกมเฉพาะปี 2014 ที่ทำการดึงข้อมูลด้วย API มีข้อมูลที่ใช้ได้ทั้งหมด 70 เกม โดยมีค่ารหัสวิดีโอเกมที่เข้ากันกับข้อมูลทั้งหมดในปี 2014 มีค่าไม่ซ้ำกัน และไม่มีค่าว่างในตารางข้อมูล ดังรูปที่ 3-2

|   | Name                               | Platform                 | Genre   | Publisher            |
|---|------------------------------------|--------------------------|---------|----------------------|
| 0 | 2014 FIFA World Cup Brazil         | X360                     | Sports  | Electronic Arts      |
| 1 | Assassins Creed: Rogue             | PS3, X360                | Action  | Ubisoft              |
| 2 | Borderlands: The Pre-Sequel        | X360, PS3, PC            | Shooter | Take-Two Interactive |
| 3 | Cabelas Big Game Hunter: Pro Hunts | PS3, X360, WiiU          | Shooter | Activision           |
| 4 | Call of Duty: Advanced Warfare     | PS4, Xone, PS3, X360, PC | Shooter | Activision           |

รูปที่ 3-2 ตัวอย่างข้อมูลวิดีโอเกมเฉพาะปี 2014 ที่ดึงข้อมูลด้วย API

### 3.4 การเขียนโปรแกรมพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด

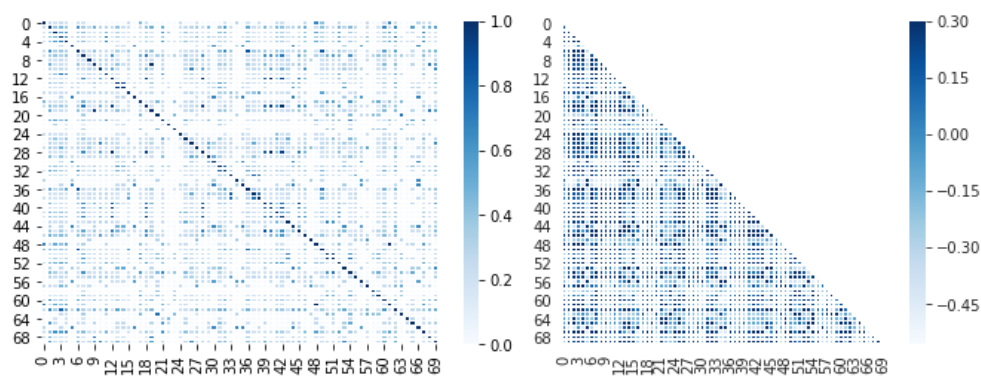
#### 3.4.1 การพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบ Content-based filtering

ใช้ข้อมูลวิดีโอเกมทั้งหมด 70 เกม ในการเทรนโมเดลระบบแนะนำวิดีโอเกมแบบ Content-based filtering โดยใช้วิธี Bag of words รวมข้อมูลต่างๆของวิดีโอเกม ดังรูปที่ 3-3

| Name                               | bag_of_words                             |
|------------------------------------|------------------------------------------|
| 2014 FIFA World Cup Brazil         | X360 Sports Electronic Arts              |
| Assassins Creed: Rogue             | PS3 X360 Action Ubisoft                  |
| Borderlands: The Pre-Sequel        | X360 PS3 PC Shooter Take-Two Interactive |
| Cabelas Big Game Hunter: Pro Hunts | PS3 X360 WiiU Shooter Activision         |
| Call of Duty: Advanced Warfare     | PS4 Xone PS3 X360 PC Shooter Activision  |

รูปที่ 3-3 ตัวอย่าง Bag of words ของวิดีโอเกม 70 เกม

ทำการแปลงข้อมูลวิดีโอเกมเป็นเวกเตอร์ แล้วใช้ Cosine similarity คำนวณความคล้ายคลึงกัน  
ของวิดีโอเกมทั้งหมด 70 เกม ดังรูปที่ 3-4



รูปที่ 3-4 Cosine similarity matrix ของวิดีโอเกม 70 เกม

ทำการทำนายวิดีโอเกมที่มีความคล้ายคลึงกับเกมที่เลือก (Call of Duty: Advanced Warfare)  
มากที่สุด 30 เกมแรก ดังรูปที่ 3-5

```
Game = 'Call of Duty: Advanced Warfare'
recommendations(Game)

['Cabelas Big Game Hunter: Pro Hunts',
'Destiny',
'Wolfenstein: The New Order',
'Borderlands: The Pre-Sequel',
'Far Cry 4',
'Metal Gear Solid V: Ground Zeroes',
'Sniper Elite 3',
'Transformer: Rise of the Dark Spark',
'Titanfall',
'Plants vs. Zombies: Garden Warfare',
'Dark Souls II',
'Teenage Mutant Ninja Turtles: Danger of the Ooze',
'GRID Autosport',
'Persona 4: Arena Ultimax',
'Assassins Creed: Rogue',
'South Park: The Stick of Truth',
'Thief (2014)',
'MX vs. ATV Supercross',
'Naruto Shippuden: Ultimate Ninja Storm Revolution',
'Sacred 3',
'Castlevania: Lords of Shadow 2',
'Pro Evolution Soccer 2015',
'Dragon Ball Z: Battle of Z',
'LEGO The Hobbit',
'How to Train Your Dragon 2',
'Lego Batman 3: Beyond Gotham',
'Dengeki Bunko Fighting Climax',
'Hatsune Miku: Project Diva F 2nd',
'Strider (2014)',
'Watch Dogs']
```

รูปที่ 3-5 รูป 30 เกมแรกที่มีความคล้ายคลึงกัน

### 3.4.2 การผสานข้อมูลวิดีโอเกมสำหรับการเทรนโมเดล

ทำการนำข้อมูลวิดีโอเกมที่มีความคล้ายคลึงกับเกมที่เลือก (Call of Duty: Advanced Warfare) มากที่สุด 30 เกมแรก ผสานกับข้อมูลคะแนนวิดีโอเกมในปี 2014 และคัดเลือกข้อมูล โดยเลือกข้อมูลที่ใช้ 30 คนแรก ให้คะแนนวิดีโอเกม 30 เกมที่น้อยที่สุด ในการเทรนโมเดลระบบแนะนำวิดีโอเกมแบบ Collaborative Filtering ดังรูปที่ 3-6

|   | User           | Game       | Rating |
|---|----------------|------------|--------|
| 0 | A3DSVJ3RYX7SNF | B00G6MW4SM | 3      |
| 1 | A3DSVJ3RYX7SNF | B00IGH1LJ8 | 1      |
| 2 | A3DSVJ3RYX7SNF | B00IGHP28Y | 1      |
| 3 | A3IOB1VUWZJIL  | B00DC9SWWE | 3      |
| 4 | A3IOB1VUWZJIL  | B00IGHY7LC | 2      |
| 5 | A1M19BGTJ5PI1I | B00DC9T2J6 | 4      |
| 6 | A1M19BGTJ5PI1I | B00DC9SWWE | 4      |
| 7 | A1GD1G8XXPRYQ8 | B006JSY3MI | 4      |
| 8 | A1GD1G8XXPRYQ8 | B00DC9T2J6 | 1      |
| 9 | A1N4PW6FTMBUSQ | B00DC9SWWE | 5      |

รูปที่ 3-6 ตัวอย่างข้อมูลคะแนนวิดีโอเกมที่ใช้ 30 คนแรก ให้คะแนนวิดีโอเกมบ้อยที่สุด

### 3.4.3 การพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบ Collaborative Filtering

ทำการสร้างเมทริกซ์คะแนน จากข้อมูลคะแนนวิดีโอเกมที่ใช้ 30 คนแรก ให้คะแนนวิดีโอเกม 30 เกมที่น้อยที่สุด โดยภายในเมทริกซ์คะแนน บรรจุด้วยค่าคะแนนที่ผู้ใช้ให้กับวิดีโอเกมนั้นๆ และมีค่าคะแนนว่างเยอะ เป็นจำนวนร้อยละ 92 จากทั้งหมด ดังรูปที่ 3-7

|   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | ... |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | ... |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 3 | 5.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |

รูปที่ 3-7 ตัวอย่างเมทริกซ์คะแนน

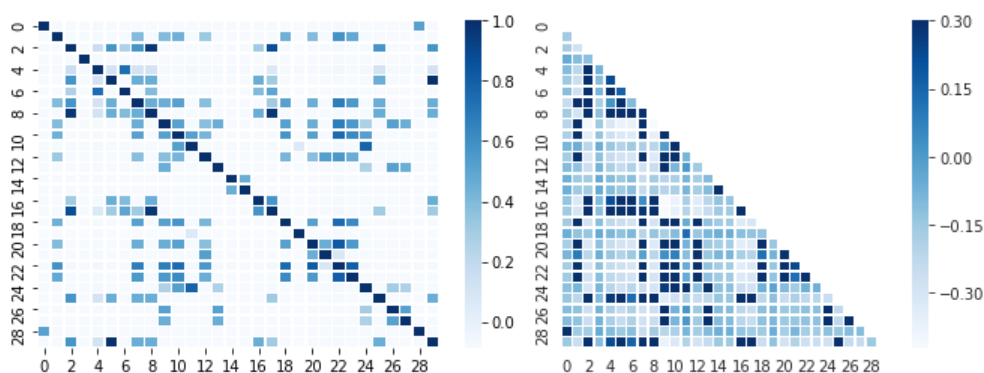


### 3.4.3.2 การพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบ User-based

#### Collaborative Filtering

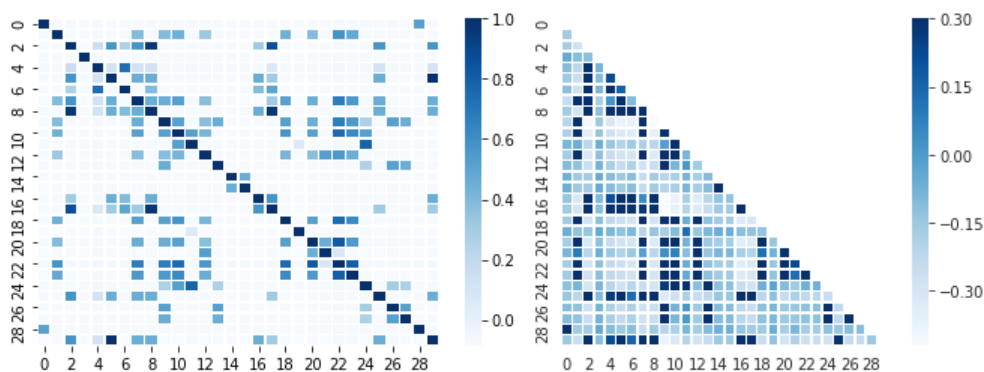
ใช้เมทริกซ์คะแนน ในการเทรนโมเดลระบบแนะนำวิดีโอเกมแบบ User-based Collaborative Filtering สำหรับการทำนายหาค่าคะแนนประมาณ ในเมทริกซ์คะแนนเดิม โดยใช้ Cosine similarity และ Pearson correlation similarity คำนวณความคล้ายคลึงกันของผู้ใช้

ทำการแปลงข้อมูลผู้ใช้ในเมทริกซ์คะแนนเป็นเวกเตอร์ แล้วใช้ Cosine similarity หาคความคล้ายคลึงกันระหว่างผู้ใช้ ดังรูปที่ 3-10



รูปที่ 3-10 Cosine similarity matrix ของผู้ใช้

ทำการแปลงข้อมูลผู้ใช้ในเมทริกซ์คะแนนเป็นเวกเตอร์ แล้วใช้ Pearson correlation similarity หาคความคล้ายคลึงกันระหว่างผู้ใช้ ดังรูปที่ 3-11



รูปที่ 3-11 Pearson correlation similarity matrix ของผู้ใช้

ทำการทำนายหาค่าคะแนนประมาณ ในเมทริกซ์คะแนนเดิม โดยใช้ Cosine similarity  
 ดังรูปที่ 3-12 และโดยใช้ Pearson correlation similarity ดังรูปที่ 3-13

|   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | ... |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | ... |
| 1 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 9 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |

รูปที่ 3-12 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ  
 จาก Cosine similarity ของผู้ใช้

|   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | ... |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 | ... |
| 1 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 9 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |

รูปที่ 3-13 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ  
 จาก Pearson correlation similarity ของผู้ใช้

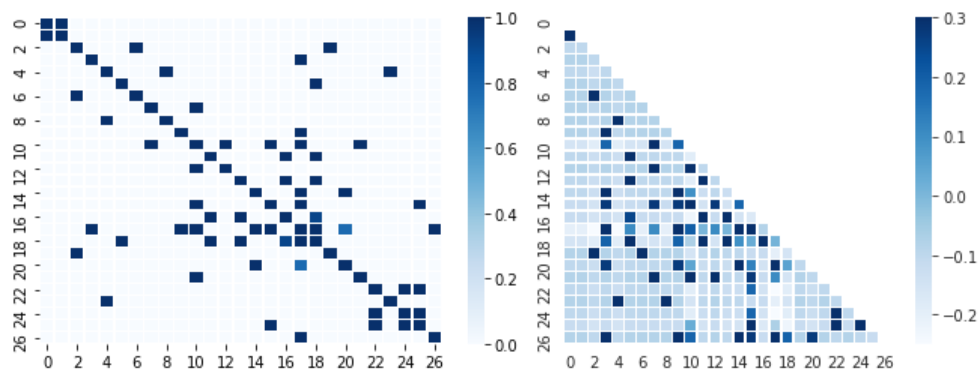
### 3.4.3.3 การพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบ Item-based

#### Collaborative Filtering

ใช้เมทริกซ์คะแนน ในการเทรนโมเดลระบบแนะนำวิดีโอเกมแบบ Item-based Collaborative Filtering สำหรับการทำนายหาค่าคะแนนประมาณ ในเมทริกซ์คะแนนเดิม โดยใช้ Cosine similarity และ Adjust cosine similarity คำนวณความคล้ายคลึงกันของวิดีโอเกม

ทำการแปลงข้อมูลวิดีโอเกมในเมทริกซ์คะแนนเป็นเวกเตอร์ แล้วใช้ Cosine similarity หาคความคล้ายคลึงกันระหว่างวิดีโอเกม

ทำการแปลงข้อมูลวิดีโอเกมในเมทริกซ์คะแนนเป็นเวกเตอร์ แล้วใช้ Adjust cosine similarity หาคความคล้ายคลึงกันระหว่างวิดีโอเกม ดังรูปที่ 3-14



รูปที่ 3-14 Adjust cosine similarity ของวิดีโอเกม

ทำการทำนายหาค่าคะแนนประมาณ ในเมทริกซ์คะแนนเดิม โดยใช้ Cosine similarity  
 ดังรูปที่ 3-15 และโดยใช้ Adjust cosine similarity ดังรูปที่ 3-16

|   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | ... |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | ... |
| 1 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | ... |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 3 | 5.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 7 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 4.0 | ... |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 9 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | ... |

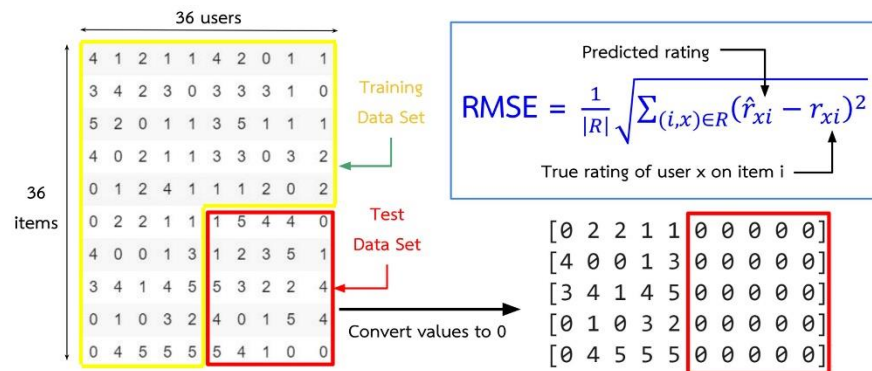
รูปที่ 3-15 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ  
 จาก Cosine similarity ของวิดีโอเกม

|   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | ... |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | ... |
| 1 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | ... |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 3 | 5.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 7 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 4.0 | ... |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... |
| 9 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | ... |

รูปที่ 3-16 ตัวอย่างเมทริกซ์ค่าคะแนนประมาณ  
 จาก Adjust cosine similarity ของวิดีโอเกม

### 3.5 การทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด

ทำการทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด โดยวัดด้วยค่า Root mean square error ในแต่ละโมเดล โดยเปรียบเทียบค่าคะแนนดั้งเดิม กับค่าคะแนนประมาณที่ทำนายได้ เปรียบเทียบเฉพาะค่าคะแนนในส่วน Test data set ที่ทำการแบ่งเมทริกซ์คะแนนไว้ 1 ส่วน จากทั้งหมด 4 ส่วน ดังรูปที่ 3-17



รูปที่ 3-17 ตัวอย่างการแบ่งเมทริกซ์ และสูตร Root mean square error

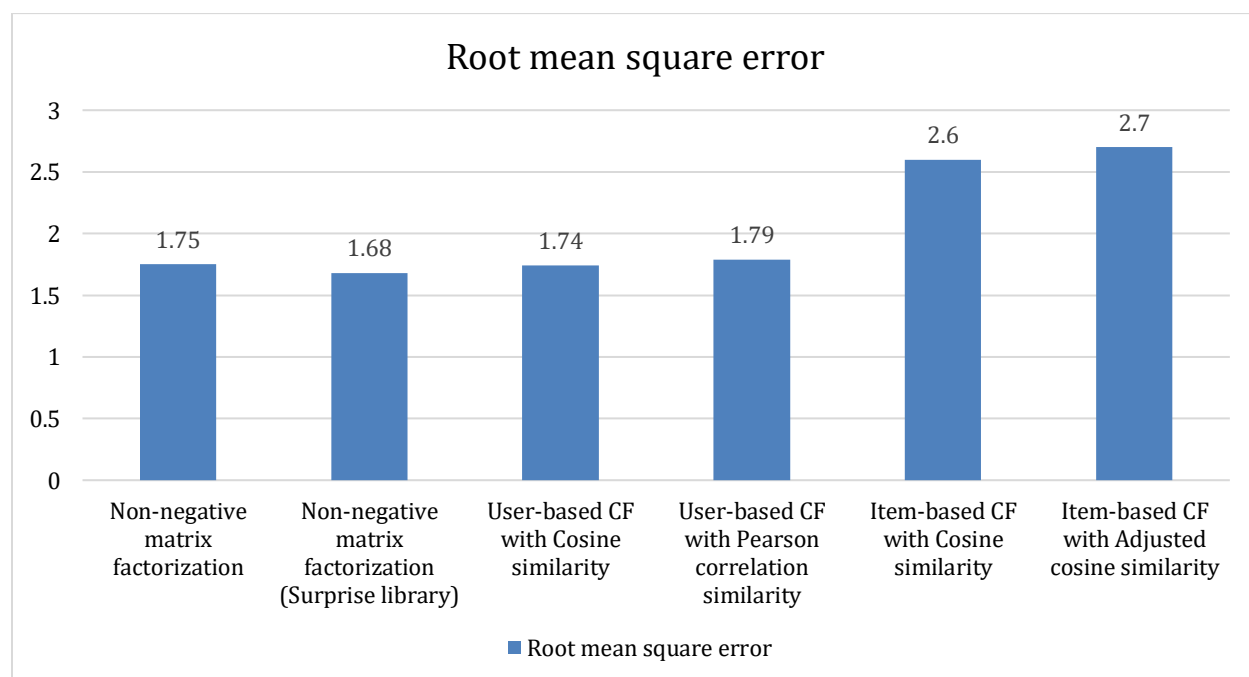
## บทที่ 4

### ผลการดำเนินโครงการ

การดำเนินโครงการในครั้งนี้ผู้วิจัยได้เสนอผลการดำเนินโครงการ โดยแสดงผลลัพธ์ดังต่อไปนี้

#### 4.1 แสดงค่า Root mean square error (RMSE) จากชุดข้อมูลวิดีโอเกม

1. โมเดลระบบแนะนำวิดีโอเกมแบบ Collaborative Filtering (CF) โดยใช้
  - 1.1. Non-negative matrix factorization (NMF) ได้ค่า RMSE = 1.75
  - 1.2. Non-negative matrix factorization Surprise library ได้ค่า RMSE = 1.68
2. โมเดลระบบแนะนำวิดีโอเกมแบบ User-based Collaborative Filtering โดยใช้
  - 2.1. Cosine similarity ได้ค่า RMSE = 1.74
  - 2.2. Pearson correlation similarity ได้ค่า RMSE = 1.79
3. โมเดลระบบแนะนำวิดีโอเกมแบบ Item-based Collaborative Filtering โดยใช้
  - 3.1. Cosine similarity ได้ค่า RMSE = 2.6
  - 3.2. Adjusted cosine similarity ได้ค่า RMSE = 2.7

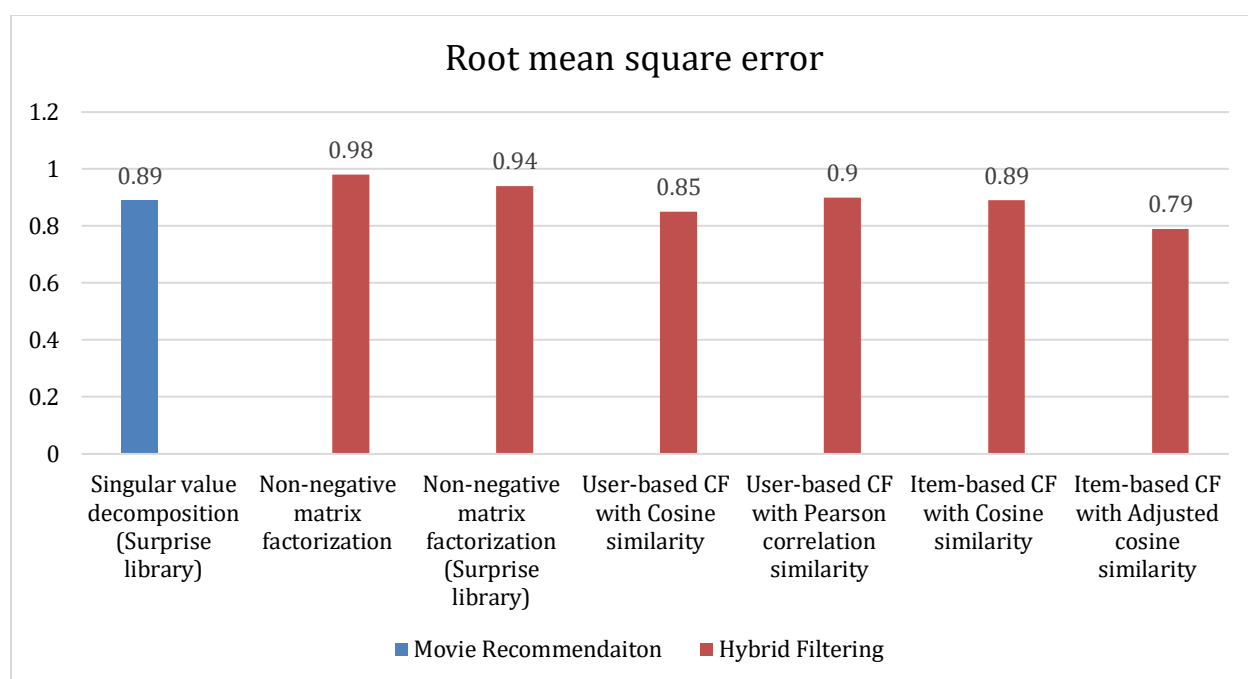


รูปที่ 4-1 รูปกราฟเส้นที่แสดงผลค่า Root mean square error

## 4.2 แสดงค่า Root mean square error จากชุดข้อมูลภาพยนตร์

ทำการนำชุดข้อมูล The Movies Dataset จากเว็บไซต์ Kaggle.com ที่มีคนทำระบบแนะนำภาพยนตร์ อยู่แล้ว และคัดเลือกข้อมูล โดยเลือกข้อมูลที่มีผู้ใช้ 30 คนแรก ที่ให้คะแนนภาพยนตร์บ่อยที่สุด และภาพยนตร์ 30 เรื่องแรกที่ถูกให้คะแนนบ่อยที่สุด แล้วทำการเทรนด้วยโมเดลระบบแนะนำแบบ Collaborative Filtering และ ทดสอบประสิทธิภาพ ความแม่นยำของโมเดล โดยวัดด้วยค่า Root mean square error

ทำการเปรียบเทียบค่า Root mean square error ที่ได้กับค่า Root mean square error ดั้งเดิม ของระบบแนะนำภาพยนตร์ ดังรูปที่ 4-2



รูปที่ 4-2 รูปกราฟแท่งที่แสดงผลการเปรียบเทียบค่า Root mean square error

## บทที่ 5

### สรุปผล อภิปรายผล และข้อเสนอแนะ

#### 5.1 สรุปผลการศึกษาค้นคว้า

เมื่อได้ศึกษา และพัฒนาระบบแนะนำวิดีโอเกมแบบไฮบริด โดยใช้วิธีการกรองข้อมูลแบบผสม (Hybrid Filtering) รวมเทคนิค Content-based Filtering และ Collaborative Filtering เข้าด้วยกัน สามารถสรุปผลลัพธ์ของการพัฒนาได้ดังต่อไปนี้

ผู้วิจัยได้ทำการพัฒนาโมเดลระบบแนะนำวิดีโอเกมแบบ Content-based Filtering รวมกับ Non-negative matrix factorization และ โมเดลระบบแนะนำวิดีโอเกมแบบ Content-based Filtering รวมกับ User-based Collaborative Filtering และ Item-based Collaborative Filtering โดยทำนายหาค่าคะแนนประมาณ จากเมทริกซ์คะแนนเดิม ที่แปลงมาจากข้อมูลคะแนนวิดีโอเกมในปี 2014 เพื่อเปรียบเทียบประสิทธิภาพ และวัดความแม่นยำของโมเดลด้วยค่า Root mean square error

จากการทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด พบว่าในแต่ละโมเดล ผู้วิจัยสามารถวัดความแม่นยำด้วยค่า Root mean square error ได้ค่อนข้างดี แต่ยังได้ค่า Root mean square error ที่สูงอยู่ เนื่องจากความเบาบางของเมทริกซ์ ที่มีค่าคะแนนว่างเยอะ ถึงร้อยละ 92 จากทั้งหมด และจากการเปรียบเทียบค่า Root mean square error กับระบบแนะนำภาพยนตร์ ด้วยระบบแนะนำแบบ Collaborative Filtering ได้ค่า Root mean square error ที่ดีที่สุดเท่ากับ 0.79 ซึ่งมีประสิทธิภาพดีมาก และมีค่าใกล้เคียงกับเทคนิค Singular value decomposition ที่ได้ค่า Root mean square error เท่ากับ 0.89 ของระบบแนะนำภาพยนตร์

## 5.2 อภิปรายผล

จากการดำเนินโครงการสามารถนำระบบแนะนำวิดีโอเกมแบบไฮบริด มาใช้ในการแนะนำวิดีโอเกมให้กับผู้ใช้ได้ โดยใช้เทคนิค Content-based Filtering ทำนายหาวิดีโอเกมที่มีความคล้ายคลึงกัน จากข้อมูลต่างๆของเกม แล้วใช้เทคนิค Collaborative Filtering ทำนายหาค่าคะแนนประมาณ จากข้อมูลคะแนนวิดีโอเกมที่คล้ายคลึงกัน ที่ได้มาจากการเทรนด้วยเทคนิค Content-based Filtering และจากการทดสอบประสิทธิภาพ ความแม่นยำของโมเดลระบบแนะนำวิดีโอเกมแบบไฮบริด ด้วยค่า Root mean square error พบว่าในแต่ละโมเดลมีประสิทธิภาพดี ได้ค่า Root mean square error ที่ดีที่สุดเท่ากับ 1.74 สรุปได้ว่าระบบแนะนำวิดีโอเกมแบบไฮบริด สามารถนำมาใช้ในการแนะนำวิดีโอเกมให้กับผู้ใช้ได้แม่นยำ ถ้าข้อมูลมีความเบาบางน้อย (Low Sparsity)

## 5.3 ข้อเสนอแนะ

เนื่องจากโครงการนี้ยังมีจุดบกพร่องอยู่ในชุดข้อมูล เช่น ความเบาบางของข้อมูลคะแนนวิดีโอเกม หลังจากที่แปลงข้อมูลคะแนนวิดีโอเกมเป็นเมทริกซ์คะแนนแล้ว ในเมทริกซ์คะแนนมีค่าคะแนนว่างเยอะถึงร้อยละ 92 จากทั้งหมด จึงทำให้การทำนายหาค่าคะแนนประมาณ ได้ประสิทธิภาพไม่ดีเท่าที่ควรจะเป็น เนื่องจากความเบาบางของข้อมูลคะแนนวิดีโอเกม สมควรใช้ชุดข้อมูลที่มีความเบาบางต่ำกว่านี้

## บรรณานุกรม

- Phukseng, Thanaphon & Smanchat, Sucha & Sodsee, Sunantha. (2558). **Recommender System Using Trust and Expert**. Faculty of Information Technology, King Mongkut's University of Technology North Bangkok.
- ชัยวัฒน์ ตีรวิรัชจร. (2550). **การทำนายข้อมูลโดยการรวม Content-Based Filtering With Item-Based Colaborative Filtering ด้วยกฎความสัมพันธ์**. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- John S. Breese; David Heckerman & Carl Kadie (2541). **Empirical analysis of predictive algorithms for collaborative filtering**. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98).
- Badaro, Gilbert & Hajj, Hazem & El-Hajj, Wassim & Nachman, Lama. (2556). **A Hybrid Approach with Collaborative Filtering for Recommender Systems**. 2556 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2556.
- พันธุ์หญิง รัฐวรรณ พันธุนิล, ผู้ช่วยศาสตราจารย์ ดร. นิเวศ จิระวิจิตชัย. (2560). **A PERSONALIZED RECOMMENDER SYSTEM USING USER-BASED AND ITEM-BASED COLLABORATIVE FILTERING METHOD**. ปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยศรีปทุม.
- MAYA.HRISTAKEVA. (2558). **A PRACTICAL GUIDE TO BUILDING RECOMMENDER SYSTEMS**. สืบค้นเมื่อ 22 พฤศจิกายน 2561, จาก <https://buildingrecommenders.wordpress.com/2015/11/20/overview-of-recommender-algorithms-part-4/>
- Koren, Yehuda & Bell, Robert & Volinsky, Chris. (2009). **Matrix factorization techniques for recommender systems**. IEEE, Computer Journal, 42(8), 30-37. Computer. 42. 30 - 37. 10.1109/MC.2009.263.

Bokde, Dheeraj & Girase, Sheetal & Mukhopadhyay, Debajyoti. (2558). **Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey**. Procedia Computer Science. 49. 10.1016/j.procs.2015.04.237.

Gábor Takács et al (2551). **Matrix factorization and neighbor-based algorithms for the Netflix prize problem**. In Proceedings of the 2008 ACM Conference on Recommender Systems, Lausanne, Switzerland, October 23 - 25, 267-274.

Patrick Ott (2551). **Incremental Matrix Factorization for Collaborative Filtering**. Science, Technology and Design 01/2008, Anhalt University of Applied Sciences.

Daniel D. Lee and H. Sebastian Seung (2544). **Algorithms for Non-negative Matrix Factorization**. Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference. MIT Press. pp. 556–562.

Daniel D. Lee and H. Sebastian Seung (2542). **Learning the parts of objects by non-negative matrix factorization**. Nature, Vol. 401, No. 6755. (21 October 1999), pp. 788-791.

M.Deshpande and G. Karypis. **Item-based top-n recommendation algorithms**. ACM Trans. Inf. Syst., 22(1):143-177, 2004.

B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Reidl. **Item-based collaborative filtering recommendation algorithms**. In Proceedings of the 10th International World Wide Web Conference, pages 285-295, 2001.

## ภาคผนวก ก

### การทำงานในส่วนของ Python

ไฟล์ Content based & Non-negative matrix factorization.ipynb

- ทำการติดตั้งผ่าน Google Colaboratory เพื่อใช้ Surprise library

```
!pip install surprise
```

- Modules ที่ใช้ในไฟล์ Content based & Non-negative matrix factorization.ipynb

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Module   | Pandas                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| การทำงาน | <p>ใช้ Pandas ในการโหลดไฟล์ข้อมูล excel จะได้ข้อมูลในรูปแบบตาราง (DataFrame) ที่แบ่งข้อมูลตามแถวและคอลัมน์</p> <p>ใช้ Pandas ในการสร้าง Series สำหรับเก็บข้อมูลวิดีโอเกม ในรูปแบบดัชนี (Index)</p> <ul style="list-style-type: none"> <li>• Series()</li> </ul> <p>ใช้ Pandas ในการ Merge ข้อมูลวิดีโอเกม</p> <ul style="list-style-type: none"> <li>• <code>merge(df_rating, df_content, left_on='Game', right_on='Game')</code><br/><code>.sort_values('Name')</code></li> </ul> <p>ใช้ Pandas ในการสร้างเมทริกซ์</p> <ul style="list-style-type: none"> <li>• <code>pivot(index = 'User', columns = 'Game', values = 'Rating').fillna(0)</code></li> </ul> |
| Module   | Numpy                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| การทำงาน | <p>ใช้ Numpy ในการคำนวณตัวเลขต่างๆ และแบ่งข้อมูลวิดีโอเกมใส่ใน Array</p> <ul style="list-style-type: none"> <li>• <code>map(lambda x: x.split(','))</code></li> </ul> <p>ใช้ Numpy ในการรวมข้อมูลวิดีโอเกมไว้ใน Bag of words</p> <ul style="list-style-type: none"> <li>• <code>drop(columns = [col for col in df_title.columns if col!= 'bag_of_words'], inplace = True)</code></li> </ul> <p>ใช้ Numpy ในการคำนวณความเบาบางของเมทริกซ์</p> <ul style="list-style-type: none"> <li>• <code>count_nonzero(R_df)</code></li> </ul> <p>ใช้ Numpy ในการแบ่งเมทริกซ์สำหรับการเทรนโมเดล และทดสอบประสิทธิภาพ</p>                                                    |

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <p>ใช้ Numpy ในการเขียนฟังก์ชันของเทคนิค Non-negative matrix factorization</p> <ul style="list-style-type: none"> <li>MF(R, K=4, alpha=0.1, beta=0.01, iterations=40)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Module   | Sklearn                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| การทำงาน | <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการแปลงข้อมูลของวิดีโอเกม Vector</p> <ul style="list-style-type: none"> <li>countVectorizer()</li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการสร้าง Cosine similarity matrix</p> <ul style="list-style-type: none"> <li>cosine_similarity(count_matrix, count_matrix)</li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการคำนวณหาค่า Root mean square error</p> <ul style="list-style-type: none"> <li>MSE = round(mean_squared_error(predicted_rating, actual_rating),3)</li> <li>RMSE = round(sqrt(MSE),3)</li> </ul> |
| Module   | Seaborn                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| การทำงาน | <p>ใช้ Seaborn ในการสร้าง Heatmap สำหรับพลอตกราฟสองมิติเพื่อแสดงผล</p> <ul style="list-style-type: none"> <li>sns.heatmap(cosine_sim, linewidth=0.5, cmap="Blues")</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Module   | Matplotlib                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| การทำงาน | <p>ใช้ Matplotlib ในการคำนวณด้านคณิตศาสตร์ และพลอตกราฟสองมิติจาก array รวมทั้งบันทึกผลที่ได้ออกมาเป็นรูปภาพ</p> <ul style="list-style-type: none"> <li>plt.show(), plt.figure(figsize=((16,4))), plt.plot(x, y), plt.xticks(x, x), plt.xlabel("Iterations"), plt.ylabel("Mean Square Error"), plt.grid(axis="y")</li> </ul>                                                                                                                                                                                                                                                                        |
| Module   | Surprise                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| การทำงาน | <p>ใช้ Surprise ในการในการคำนวณหาค่า Root mean square error จากเทคนิคแบบ Non-negative matrix factorization</p> <ul style="list-style-type: none"> <li>evaluate(nmf, data, measures=['RMSE', 'MAE'])</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                     |

## ไฟล์ Content based & User-based and Item-based.ipynb

- Modules ที่ใช้ในไฟล์ Content based & Non-negative matrix factorization.ipynb

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Module   | Pandas                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| การทำงาน | <p>ใช้ Pandas ในการโหลดไฟล์ข้อมูล excel จะได้ข้อมูลในรูปแบบตาราง (DataFrame) ที่แบ่งข้อมูลตามแถวและคอลัมน์</p> <p>ใช้ Pandas ในการสร้างเมทริกซ์</p> <ul style="list-style-type: none"> <li><code>pivot(index = 'User', columns = 'Game', values = 'Rating').fillna(0)</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Module   | Numpy                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| การทำงาน | <p>ใช้ Numpy ในการคำนวณตัวเลขต่างๆ และการคำนวณด้านคณิตศาสตร์</p> <p>ใช้ Numpy ในการแบ่งเมทริกซ์สำหรับการเทรนโมเดล และทดสอบประสิทธิภาพ</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Module   | Sklearn                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| การทำงาน | <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการหา Cosine similarity เพื่อสร้าง Cosine similarity matrix</p> <ul style="list-style-type: none"> <li><code>cosine_sim = 1-pairwise_distances(M, metric="cosine")</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการหา Pearson correlation similarity เพื่อสร้าง Pearson correlation similarity matrix</p> <ul style="list-style-type: none"> <li><code>pearson_sim = 1-pairwise_distances(M, metric="correlation")</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการเขียนฟังก์ชันสำหรับหาผู้ใช้ที่คล้ายคลึงกัน</p> <ul style="list-style-type: none"> <li><code>findksimilarusers(user_id, ratings, metric = metric, k=k)</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการเขียนฟังก์ชันสำหรับหาค่าคะแนน ของผู้ใช้ที่คล้ายคลึงกัน</p> <ul style="list-style-type: none"> <li><code>predict_userbased(user_id, item_id, ratings, metric = metric, k=k)</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการเขียนฟังก์ชันสำหรับหาวิดีโอเกมที่คล้ายคลึงกัน</p> <ul style="list-style-type: none"> <li><code>findksimilaritems(item_id, ratings, metric=metric, k=k)</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการเขียนฟังก์ชันสำหรับหาค่าคะแนน ของวิดีโอเกมที่คล้ายคลึงกัน</p> |

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | <ul style="list-style-type: none"> <li>• <code>predict_itembased(user_id, item_id, ratings, metric = metric, k=k)</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการเขียนฟังก์ชันการคำนวณ Adjusted cosine similarity matrix สำหรับวิดีโอเกม</p> <ul style="list-style-type: none"> <li>• <code>computeAdjCosSim(M)</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการเขียนฟังก์ชันแบบเทคนิค Adjusted cosine similarity สำหรับหาค่าคะแนน ของวิดีโอเกมที่คล้ายคลึงกัน</p> <ul style="list-style-type: none"> <li>• <code>predict_userbased(user_id, item_id, ratings, metric = metric, k=k)</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการทำนายค่าคะแนนประมาณในเมทริกซ์</p> <ul style="list-style-type: none"> <li>• <code>prediction = predict_userbased(user_id, item_id, ratings, metric)</code></li> <li>• <code>prediction = predict_itembased(user_id, item_id, ratings)</code></li> <li>• <code>prediction = predict_itembased_adjcos(user_id,item_id,ratings)</code></li> </ul> <p>ใช้ Scikit-learn ซึ่งเป็น library ใน Python ในการคำนวณหาค่า Root mean square error</p> <ul style="list-style-type: none"> <li>• <code>MSE = round(mean_squared_error(predicted_rating, actual_rating),3)</code></li> <li>• <code>RMSE = round(sqrt(MSE),3)</code></li> </ul> |
| Module   | Seaborn                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| การทำงาน | <p>ใช้ Seaborn ในการสร้าง Heatmap สำหรับพลอตกราฟสองมิติเพื่อแสดงผล</p> <ul style="list-style-type: none"> <li>• <code>sns.heatmap(cosine_sim, linewidth=0.5, cmap="Blues")</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Module   | Matplotlib                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| การทำงาน | <p>ใช้ Matplotlib ในการคำนวณด้านคณิตศาสตร์ และพลอตกราฟสองมิติจาก array รวมทั้งบันทึกผลที่ได้ออกมาเป็นรูปภาพ</p> <ul style="list-style-type: none"> <li>• <code>plt.show()</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |